Civil Liberties in Personalising Autocracies

How personalism degrades civil liberties by disempowering repressors

Author: Siebren Kooistra

Student Number: 3959627

Programme: BSc Sociology, Rijksuniversiteit Groningen

Supervisor: dr. Jacob Dijkstra

Second evaluator: dr. Rita Smaniotto

June 2022

Abstract:

This thesis investigates the way personalism (e.g., Geddes et al., 2018) affects four civil liberties. I theorise greater personalism to decrease freedom of expression through limiting perceived coup threat, decrease freedom of assembly because the dictator has less need for the general population to monitor the elite, and lower protection of life and physical integrity and freedom of movement because the competence of coercive institutions degrades. I test these hypotheses using multi-level (random intercept) models and find that more personalist countries have slightly lower civil liberties. The competence of coercive institutions operationalised as the rigour and impartiality of the public administration might mediate the relation between personalism both protection of life and physical integrity and freedom of movement, and possibly the relations to other civil liberties. My findings seem in line with Frantz et al. (2019), but substantively weak results and problems in accounting for dependence structures restrict my conclusions.

Table of Contents

Introduction:

The rule of one takes the effort of many and the compliance of more. This means that explaining such a rule of one, a personalised autocracy, requires a look at the interactions between a dictator, their elite, and the general populace. Previous work has already done much to explain how dictators gain personal power and consolidate it (e.g., Acemoglu et al., 2009; Geddes et al., 2018; Svolik, 2012). I will look into the consequences of personalism for four civil liberties, rights allotting freedom from various kinds of coercion or freedom to make certain choices (Davenport, 2007a).

Existing research into the civil liberties and personalism has looked into the effect of personalisation on the protection of life and physical integrity (Frantz et al., 2019; Greitens, 2016) and freedom of expression (Boleslavsky et al., 2021; Hollyer et al., 2019). I try to build on this knowledge, and will explicitly distinguish between freedom of expression, freedom of assembly, freedom of movement and protection of life and physical integrity. The processes creating personalised autocracy might affect different civil liberties in distinctive ways, making it theoretically meaningful to separate out the effects of personalism on various civil liberties (Davenport, 2007b; Møller & Skaaning, 2013a). To deepen knowledge on these effects, I will also test to what extent the relation between personalism and civil liberties is mediated by variables linked to the mechanisms I identify in my theory. My research question is:

"To what extent do the level of personalism and changes in the level of personalism within an autocratically ruled country affect respect for various civil liberties, and what are the mechanisms underlying these relations?"

To explain how personalised autocracy affects civil liberties, it is important to first establish what personalism is, and what perspective I will take on it. *Personalism* is the extent to which power in a regime is concentrated in the hands of a single individual (Geddes et al., 2018; Sinkkonen, 2021; Svolik, 2012). To explain how personalism evolves over time, I will use the coalition-formation model established by Acemoglu et al. (2008, 2009, 2012). This model starts from the assumption that autocratic regimes lack effective formal rules and procedures to regulate political actions (Gehlbach et al., 2016). This makes credible commitments hard, costly, or outright impossible (Acemoglu et al., 2008, 2009; Gehlbach et al., 2016; Svolik, 2012). When there is no trust in an independent law enforcement system (judiciary, police) for example, members of the elite with influence over parts of it will be reluctant to give up this influence for fear of later being persecuted by that same system. Even if the person who receives power from others is benevolent, the fact that the power of this individual makes them impervious to external control makes others mistrustful. As such, those with access to power in an autocratic regime (the elite) feel like they must use it to have their interests met, and strive to increase their power for further security.

Because members of an autocratic elite strive to increase their power, they want a ruling coalition including themselves to be as small possible (Acemoglu et al., 2008). This means that members of a ruling coalition have an incentive to exclude others when they can safely do so. The incentive of members of the ruling coalition to shrink the ruling coalition creates the risk of factionalist infighting, and dictators tend to exploit factionalism in autocratic regimes by acting as an arbiter between factions (Geddes et al., 2018). By strategically choosing sides in internal conflicts, successful dictators eliminate powerful factions within the regime until the remaining factions are no longer able to limit this dictator (Geddes et al., 2018; Svolik, 2012). A dictator that has accrued some personal power can further reinforce factionalism by encouraging competition among government institutions and limiting the elite's opportunities to communicate and establish mutual trust (Geddes et al., 2018; Greitens, 2016). Still, leveraging internal conflict to personalise the regime fails more often than not, since (the rest of) the elite will eliminate a dictator if they comprehend the dictator's strategy on time (Svolik, 2012), while cohesive ruling coalitions lack the necessary factionalism (Geddes et al., 2018; see also Levitsky & Way, 2016). But people do not always have the foresight to avoid being outplayed (Jandoc & Juarez, 2019), and exogenous events or failed resistance can tip the balance of power (Acemoglu et al., 2008; Chin, 2020).

To summarise, previous research suggests that personalism results from the disruptions to the balance of power within an autocratic elite, and reduces protection of life and physical integrity and freedom of expression. I will broaden this research by considering not only protection of life and physical integrity and freedom of expression, but also freedom of assembly and freedom of movement. I will also try to deepen the research by looking at the mechanisms underlying relations between personalism and civil liberties.

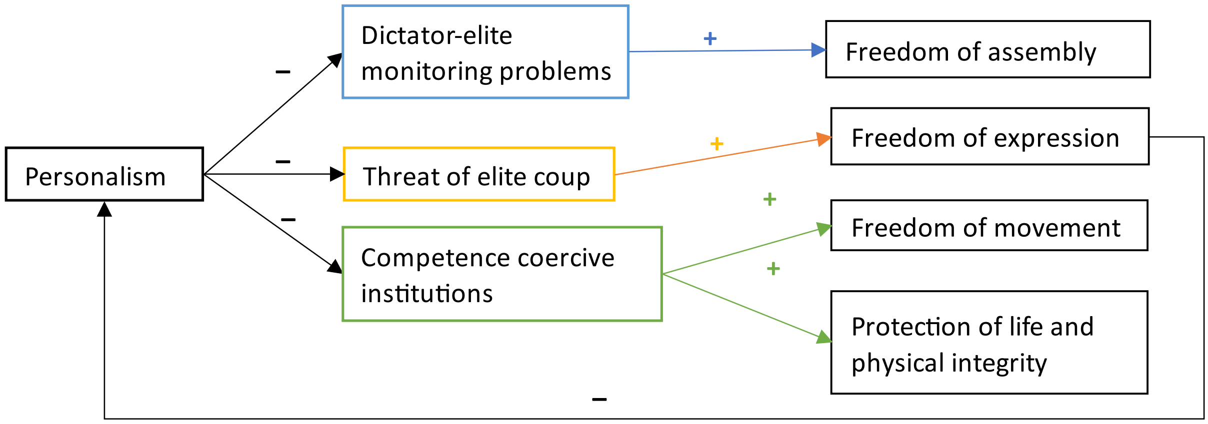
In the next section, I will detail the mechanisms through which personalism affects civil liberties. After this theoretical argument, I use a third section to introduce the dataset and my empirical approach. The fourth section gives descriptive statistics of the dataset and model results. I draw my conclusions and reflect on them in a fifth section, which rounds off the article.

Theory:

In this section, I will outline my hypotheses and the reasoning behind them. In doing this, I build on the general perspective presented in the introduction. My hypotheses are summarised in Figure 1.

Figure 1

Conceptual model of the relations between personalism and freedom of assembly, freedom of expression, freedom of movement and protection of life and physical integrity.



Note: Although omitted in the figure, it can be assumed that there are also direct arrows from personalism to civil liberties to account for other mechanisms linking personalism to civil liberties, even if these are not of interest to this study.

Freedom of expression: Transparency and shifting risks

Freedom of expression is the freedom to express, seek and receive thoughts and opinions in private or in public (Coppedge et al., 2021a, p. 46, p. 307; Møller & Skaaning, 2013b, p. 1071; Universal Declaration of Human Rights, 1948, Art. 19).

For the mechanism linking of personalism to freedom of expression and personalism, I propose that weak dictators maintain freedom of expression to keep their rivals in check, but sufficiently powerful dictators no longer need to do this and want to remove these checks when acting against their rivals. By and large, dictators mostly have an opportunity to establish themselves when the elite is factionalised and prone to infighting (Geddes et al., 2018). For the situation where a dictator is still relatively weak, this creates an initial environment where members of the elite have to be more wary of being ousted in infighting than having the regime come down due to popular protest. This is the case because members of the elite need to overcome less of a coordination problem than the general populace to rise up against a dictator (Svolik, 2012; Geddes et al., 2018). To keep the threat of an internal power grab in check, members of the elite use the fact that cooperation problems for the general populace decrease when they can coordinate on shared knowledge of infighting in and thus vulnerability of the ruling coalition (Hollyer et al., 2019). To enable the creation of such shared knowledge, the dictator and other members of the elite can facilitate the general populace (journalists in particular) in observing and reporting infighting in the elite. While this poses a risk to the elite (including the dictator), this is outweighed by the fact that it discourages infighting that might lead to their own exclusion (Boleslavsky et al., 2021; Hollyer et al., 2019).

The effort required to restrict freedom of expression makes it unlikely that the dictator, or any subgroup of the elite for that matter, will unilaterally restrict freedom of expression as long as they do not completely dominate the coalition. Comprehensively limiting freedom of expression requires a great deal of coordination. And when it happens, it is likely to signal both to the rest of the elite and to the general populace that that subgroup is an urgent threat to their interests of retaining, respectively, power and freedom of expression. By unilaterally trying to limit freedom of expression, members of the ruling coalition incur the risk of facing coordinated resistance from both the rest of the ruling coalition and the general populace. Unless at least one of these threats can be removed, limiting freedom of expression makes the perpetrator's position untenable.

A dictator in a factionalised ruling coalition that is still vulnerable to coordinated resistance from the rest of the elite has a use for freedom of expression and incurs a lot of risk by limiting it. However, once their power becomes great enough to make resistance from within the elite ineffectual (see Svolik, 2012), they have an opportunity to limit freedom of expression. Since their domination of the elite means that they no longer need external control to keep their rivals in check, their priority lies with limiting freedom of expression to disrupt coordination in the general populace (Boleslavsky et al., 2021; Hollyer et al., 2019). When moving to limit freedom of expression the dictator will still face popular resistance, but since the dictator can use considerable government resources it is likely that they can overcome this counterforce.

A point of note here is that the mechanism I sketched above not only implies that freedom of expression is limited after personalism is at sufficiently high levels, but also that limiting freedom of expression stimulates greater personalism. Decreasing freedom of expression takes away the constraints on infighting that drove the elite to accept some freedom of expression in the first place, and as such lower freedom of expression can amplify personalisation by permitting the dictator more aggressive use of their power dominance.

Empirically, I expect that from a factionalised ruling coalition that allows some freedom of expression, increasing personalism leaves freedom of expression unchanged at first but leads to a decrease in freedom of expression once the dictator has consolidated their position. I also expect there to be a reverse relation of less freedom of expression leading to increasing personalism, but I do not posit a specific hypothesis about this. This makes my hypothesis on freedom of expression:

H_1: *Freedom of expression in a country has a curvilinear association with personalism, remaining fairly constant at low levels of personalism and decreasing at high levels of personalism.*

Freedom of assembly: Who monitors the elite?

Freedom of assembly is the freedom to peacefully gather to signal opinions or concerns in public (Coppedge et al., 2021a, p. 225; Møller & Skaaning, 2013b, p. 1071).

My theory on the relation between personalism and freedom of assembly is based on the ability of a dictator to monitor their elite. Once in power, dictators need to monitor their elites both to make sure that they carry out their policies and to prevent any members of the elite from illicitly building an independent power base. For monitoring and disciplining their officials, dictators can use freedom of assembly by justifying punishment of threatening officials with public discontent. In addition, giving the general populace some opportunities to voice their discontents grants the dictator a source of information about unusual or dysfunctional behaviour among the executive elite (Geddes, 2018; Zu, 2020). However, the downside of freedom of assembly is that protests can be targeted against the dictator and their government as a whole as well as against specific policies. The dictator will want to prevent such protests, and repress them when they do occur (Davenport, 2007a; Zu, 2020). A dictator thus faces a trade-off between leaving some freedom of assembly to monitor their elite, and further limiting it to minimise the risk of protest against the regime as a whole.

This trade-off makes the creation of parallel coercive institutions and overlap in roles very important (Greitens, 2016). It is common for dictators to create several *coercive institutions*, government agencies tasked with repressing dissent, with similar tasks and unclear authority hierarchies between them (Greitens, 2016). When dictators do this, they can use these parallel organisations to monitor each other. At a more general level, a dictator can make the military, a support party and secret police monitor each other (Geddes et al., 2018). This means that dictators can keep watch over the majority of their government apparatus without the need for popular protest. Restrictions to freedom of assembly then become more advantageous, since the risks of popular protest remain while the benefits have decreased.

Succinctly, freedom of assembly offers a way to control the executive elite, but can be supplanted by using the elite to monitor itself once a dictator has sufficient power. My hypothesis on freedom of assembly is then:

H_2: *Higher levels of personalism are associated with lower levels of freedom of assembly in a country.*

Protection of life and physical integrity and freedom of movement: The competence of repressors

Protection of life and physical integrity is protection from forms of purposeful physical violence and violent coercion (e.g., torture), as well as protection against state-sanctioned killing (Coppedge et al., 2021a, p. 173; Davenport, 2007a, p. 2). *Freedom of movement* is the freedom of inhabitants of a country to go to and be at any location they wish to be for any amount of time (Coppedge et al., 2021a, pp. 182-184; Møller & Skaaning, 2013b, p. 1071; Universal Declaration of Human Rights, 1948, Art. 16).

Protection of life and physical integrity, and to a lesser extent freedom of movement, could deteriorate as personalism undermines the competence of *coercive institutions*, government agencies tasked with repressing dissent (Egorov & Sonin, 2020; Greitens, 2016). Less competent coercive institutions resort to means of repressing dissent that limit the civil liberties of large groups more readily (Greitens, 2016).

The reduction in the competence of coercive institutions follows from the clientelist divide-and-rule strategy that a personalising dictator uses, which can be illustrated with four tactics fitting such a strategy. First, rewards and job security based on loyalty might lead to neglect of competence as a selection criterion, greater tolerance of corruption, and reduced willingness to critique ineffectual policies among agents in coercive institutions (Svolik, 2012). Second, dictators might purposefully seek out less competent agents to prevent being outsmarted and removed (Egorov & Sonin, 2020; Libman, 2020). Third, dictators can use position rotation systems to control their elite, and officials that are constantly rotated can neither gain experience with local conditions nor nurture loyalty from their subordinates (Greitens, 2016). Fourth, when dictators create competing agencies fulfilling the same tasks to stimulate competition among them, the information gathered by these agencies remains fragmented and the agencies hamper each other's actions (Greitens, 2016). Overall, a dictator's strategy of divide and rule means the people selected into coercive institutions are less capable that they would be under less personalised government, and have limited opportunities to accrue competence over time.

When the competence of coercive institutions decreases, they are more likely to resort to violent and indiscriminate measures that limit civil liberties on a large scale (Greitens, 2016). This happens because disorganisation within agencies, a lack of trust among agencies and little investment in informant networks among the general population limits insight into nascent threats (Egorov & Sonin, 2020; Greitens, 2016). And for the times when coercive institutions are able to detect a threat to the government, their persuasive power is limited because confused organisation means that they have little foothold among the general population (Greitens, 2016). With little insight into threats before they grow large and reduced means to control them in a peaceful manner, coercive institutions are more likely to have to resort to violence when a threat to the government is detected (Greitens, 2016). And as coercive institutions become less competent at assessing threats, their lack of insight also reduces their ability to deploy targeted measures against these threats (Greitens, 2016) Instead of violence against clear targets, coercive institutions then have to be more indiscriminate (Greitens, 2016). Inability to gather information on threats and identify targets of repressive action also means that coercive institutions are more likely to resort to large-scale monitoring that restricts freedom of movement. To illustrate, instead of singling out dissenters through intelligence-gathering, coercive institutions with less access to information might have to establish extensive access regulations and checkpoint systems to protect important locations or disrupt open resistance. Because of this, protection of life and physical integrity is probably violated more intensely and more indiscriminately under more personalised governments, and freedom of movement is reduced.

Because I expect personalism to gradually reduce both the protection of life and physical integrity and freedom of movement through the declining competence of coercive institutions, my hypotheses for these two civil liberties are:

H_3: *Higher levels of personalism are associated to lower levels of protection of life and physical integrity in a country.*

H_4: *Higher levels of personalism are associated to lower levels of freedom of movement in a country.*

Methods:

This section will start with a description of the dataset and peculiarities in the data collection. I will then discuss the operationalisations of concepts introduced in the theory section. This is supplemented with a description of my control variables and their operationalisation. The section ends with an analysis plan.

Dataset

In order to measure both personalisation and civil liberties, I combine two datasets. For personalisation, I use data on autocratic regimes from Geddes et al. (2018) extended by a latent variable model for personalism by Wright (2021). I retrieve civil liberties indicators from version 11.1 of the V-Dem dataset (Coppedge et al., 2021b), which uses latent variable models to provide continuous measures of various kinds of civil liberties. To obtain a regime type variable, I use another dataset by Geddes et al. (2014) which contains the same years and countries as the 2018 dataset. The Geddes et al. (2014, 2018) datasets run from 1946 to 2010 and cover autocratic regimes. V-Dem extends from 1789 to 2020 and also includes democratic regimes, which means that I only use a subset of the V-Dem dataset. All three datasets are time-series cross-sectional (or panel) data sets, containing yearly observations per country (country-years). Since years within a country will be related, the dataset has a multilevel structure with country-years at level 1 nested in countries over time in level 2. The full dataset includes 4591 country-years from 119 countries observed for 1 to 65 years, but the effective size when accounting for missingness is 3406 country-years from 109 countries observed for 1 to 55 years. I will report results for this subset.

Data collection procedure

Both the Geddes et al. (2014, 2018) datasets and the V-Dem dataset (Coppedge et al., 2021c) are coded by either the scholars constructing the dataset, expert surveys or research assistants. My subset of the V-Dem data also includes a number of variables extracted from other datasets (Coppedge et al., 2021c). This means that for many variables, the final values will depend on the availability of experts and their judgements on the state of a variable in a specific country-year. V-Dem uses multiple coders per country-year, and when feasible provides estimates of the uncertainty of reported values (Coppedge et al., 2021c). Geddes et al. (2014, 2018) do not provide extensive information about the number of coders or the reliability of their estimates.

Variable modelling

The values of personalism and the scale items for civil liberties have been created via Item-Response Theory (IRT) models (Pemstein et al., 2021; Wright, 2021). This means that the responses to basic indicators (as given by coders) were taken as imperfect measurement of a latent variable, and a probable value for this latent variable was inferred from aggregating the basic indicators (Pemstein et al., 2021). For the personalism score, this aggregation was done across a number of items (Wright, 2021), while the civil liberties measures aggregate across multiple coders' responses (Pemstein et al., 2021). For personalism, I use Wright's (2021) measure directly, while for the civil liberties I use a combination of direct scores, results from Bayesian factor analyses included in the V-Dem dataset (Pemstein et al., 2021, pp. 25-26) and simple sums and/or means (uniform weighing of items) calculated either by myself or by the V-Dem team.

Lags and coding dates

The Geddes et al. (2014, 2018) datasets use January 1st as the date of coding, while V-Dem measures averages over a year. If personalism increases in the middle of one year, these changes will only be reflected in the data in the next year. This means that V-Dem scores often reflect the effect of a change in the level of personalism in the previous year. An important advantage of this is that changes in personalism in a year then always precede changes in respect for civil liberties, satisfying one condition for causal inference. As such, the personalism scores are matched to their nominal years. Previous research on the relation between repression and personalism also used this approach (Frantz et al., 2019).

Operationalisations of core variables

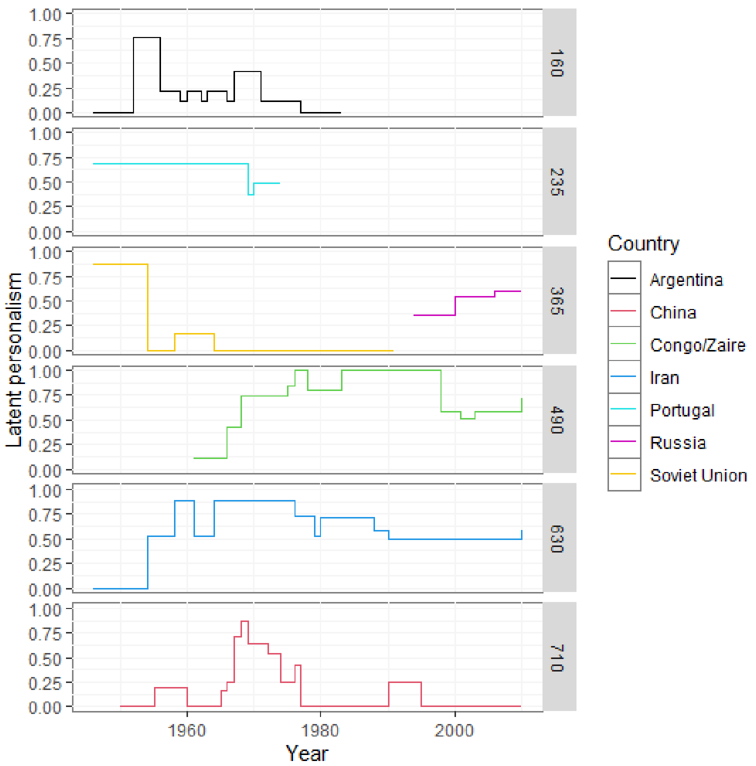
My main variables of interest are personalisation and four indicators for various kinds of civil liberties. The civil liberties measures are measures of actual respect for or violation of civil liberties.

Personalism

The basic indicators with which personalism was estimated are eight binary variables recording whether access to high office depends on loyalty to the regime leader, a leader creates a new political support party after gaining power, a regime leader controls appointments to a party executive committee, the party executive committee is either absent or powerless (a ‘rubber stamp’), the leader personally controls the security apparatus, the regime leader promotes officers loyal to himself or his group and forces officers from other groups out of the military, the leader creates paramilitary forces and whether the leader imprisons or kills officers from other groups without a fair trial (Geddes et al., 2018; Wright, 2021). Wright (2021) aggregates these indicators to a variable on the 0 to 1 range via an IRT model, with values closer to 1 indicating a greater degree of personalism. The indicators had Cronbach’s Alpha values between 0,66 and 0,83 in individual years. This variable was observed for all country-years in the full dataset (N = 4591, n = 119, T = 1-65).

Figure 2

Time series of personalism in the People’s Republic of China, the Democratic Republic of Congo (and former Zaire), Russia (and the former Soviet Union), Portugal, Iran and Argentina



Note. The time series only depict the period in which a country was classified as autocratic by Geddes et al. (2018). The 1992-1993 break in the time series for the Soviet Union and Russia signifies the transition from one country to the other and a short period where Russia was deemed democratic in the Geddes et al. (2018) coding scheme.

A practical problem with a constructed variable to measure personalism is the lack of natural reference values for interpreting the variable. To aid interpretation of the personalism scores, I have added example time-series of personalism levels for six countries in Figure 1.

To discuss a few examples, personalism rose to near unity for Mobutu Sese Seko, who is broadly recognised as a dictator with an exceptionally complete grasp on power in the Democratic Republic of the Congo/Zaire (e.g., Turner & Young, 1985). Meanwhile, communist autocracies such as the USSR and the People's Republic of China tend to have very low levels of personalism. Where this pattern is disrupted that is theoretically sensible, with episodes such as the dominant position of Joseph Stalin (McDermott, 2014) and the attempt to achieve greater power by Mao Zedong during the cultural revolution (Wright, 2021, pp. 5-6).

Freedom of expression

Freedom of expression is a variable combining IRT model values based on ordinal expert-coded assessments of the direct or indirect government censorship of print and/or broadcast media, harassment of journalists, media self-censorship, the freedom of ordinary people to discuss political matters at home and in the public sphere, and the freedom of academic and cultural expression (Coppedge et al., 2021a, p. 307). The indicators are aggregated by (Bayesian) factor analysis (Coppedge et al., 2021a, p. 307; Pemstein et al., 2020, pp. 25-26). Cronbach's Alpha varied between 0,80 and 0,93 for individual years. The variable runs from 0 to 1 with higher values indicating higher levels of freedom of expression. To avoid problems with floating point arithmetic when estimating models of freedom of expression, I multiplied the 0 to 1 scale a hundredfold (to a 0 to 100 scale).[1] The variable was observed for all country-years in the full dataset (*N* = 4591, *n* = 119, *T* = 1-65).

Freedom of assembly

Freedom of association and assembly is also measured by IRT model values based on ordinal expert coding ranging from 0 if state authorities do not allow peaceful assemblies and are willing to use lethal force to prevent them to 4 if state authorities almost always allow and actively protect peaceful assemblies except in rare cases of lawful, necessary and proportionate limitations (Coppedge et al., 2021a, p. 225). The resulting variable has a mean of zero in the complete V-Dem dataset (which includes democracies excluded from my dataset), with higher values indicating greater freedom of assembly. This variable had 4,40% of values missing (*N* = 4389, *n* = 114, *T* = 1-65).

[1] The personalism variable did not require this treatment because it is not a dependent variable. The models presented later were also estimated with personalism multiplied by a hundred, but this did not change conclusions.

Protection of life and physical integrity

Protection of life and physical integrity is measured as the mean of expert-coded ordinal indicators of freedom from torture and freedom from political killings rescaled to a 0 to 1 range (Coppedge et al., 2021a, p. 293) Higher values indicate greater protection of life and physical integrity. As with freedom of expression, I multiplied the scale by a hundred to avoid problems with floating point arithmetic. The items have correlations between 0,74 and 0,88 within individual years. The variable was observed for all country-years in the full dataset (*N* = 4591, *n* = 119, *T* = 1-65).

Freedom of movement

For freedom of movement, I averaged IRT model values from expert-coded ordinal indicators of freedom of domestic movement for men, freedom of domestic movement for women and freedom of foreign movement (Coppedge et al., 2021a, pp. 182-184). All three items have a mean of zero in the complete V-Dem dataset, with higher scores indicating greater freedom of movement. Cronbach's Alpha varies from 0,76 to 0,91 for individual years. The scale was computed as the mean of the three indicators. This variable was observed for all country-years in the full dataset (*N* = 4591, *n* = 119, *T* = 1-65).

Rigour and impartiality public administration

The rigour and impartiality of the public administration is measured by IRT model values based on expert coding ranging from 0 if the law is not respected by public officials and arbitrary or biased administration of the law is rampant to 4 if the law is generally fully respected by public officials and arbitrary or biased administration of the law is very limited (Coppedge et al., 2021a, pp. 175-176). The original (ordinal) variable was also coded 0 if no functioning public administration existed. Again, the resulting variable has a mean of zero for the complete V-Dem dataset with higher values indicating greater rigour and impartiality of the public administration. This variable was observed for all country-years in the full dataset (*N* = 4591, *n* = 119, *T* = 1-65).

Control variables and operationalisations

To avoid bias in effect estimates from my observational data, I include a number of control variables that might change over time and possibly between countries and over time. I believe that these variables can affect both respect for civil liberties and personalism, but I will only sketch a possible mechanism for each variable.

Regime type

Regime type is a nominal variable with four categories: single-party regimes, military regimes, personalist regimes and monarchies. These categories identify whether control over policy, leadership selection and the security apparatus in an autocratic regime are primarily in the hands of a ruling party (single-party), the military (military), a narrower group centred around an individual dictator (personalist) of a royal family (monarchy, Geddes et al., 2014, p. 318). Wright (2021, pp. 7-9) shows that these regime type variables are distinct from the personalism variable, which will be reflected in my bivariate results presented later. The variable is constructed from a set of four mutually exclusive (linearly dependent) binary variables, of which three are entered into regressions (Geddes et al., 2014).

International or internal conflict

The most important control variables are separate binary variables indicating whether the country was in international conflict or internal conflict (Coppedge et al., 2021a, pp. 362-363), which previous work has shown to be an important predictor of respect for civil liberties (Hill & Jones, 2014). Conflict might also affect personalism, for example because a government under external pressure is more likely to concentrate formal responsibilities in a single individual (e.g., Geddes et al., 2018, pp. 160-162), or conflict forces military spending that creates a powerbase independent from the dictator (e.g., Geddes et al. 2018, pp. 162-163; Svolik, 2012, pp. 127-133). These variables were always missing simultaneously, which was the case for 16,99% of all country-years (N = 3811, n = 115, T = 1-55).

Political violence

Since political violence is a well-established predictor of government repression (e.g., Davenport et al., 2021) and political violence might accompany or encourage personalisation, I include a measure of political violence by non-state actors (Coppedge et al., 2021a, pp. 224-225). This variable had 4,47% of its values missing (N = 4386, n = 114, T = 1-65).

Population size

I control for logged ($\log_{10}$) population size (Coppedge et al., 2021a, p. 360), as population size might affect the functioning of government both with regard to its tendency to respect civil liberties and the degree of personalism. Logged population size had 18,62% missing values (N = 3736, n = 113, T = 1-55).

GDP per capita and GDP growth

I include both GDP growth and the logged (ln) GDP per capita (Coppedge et al., 2021a, p. 358-359), as both the level of economic prosperity and changes in it may affect the political climate such that they might make personalism more (or less) likely, but also make restrictions on civil liberties more (or less) likely. Logged GDP per capita had 5,97% of values missing (N = 4317, n = 114, T = 1-61). For GDP growth, the percentage of missing values was 6,32% (N = 4301, n = 114, T = 1-61).

Analysis plan

Since I will be estimating models for four kinds of civil liberties, and thus running a large number of independent statistical tests, I will apply a Bonferroni correction to my significance level of α = 0,05 and use a significance level of α = 0,0125 at the level of a single test. I will carry out my analyses in R (R Core Team, 2021) using the "lme4" package to estimate multi-level models (Bates et al., 2015) extended by the "clubSandwich" package for clustered standard errors (Pustejovsky & Tipton, 2016). I estimate all models with REML, but re-estimate using normal ML when comparing models.

The models I will be estimating are linear random intercept models of the form:

$$CL_{i,t} = \alpha_i + \lambda_t + \beta_0 + \beta_P P_{i,t} + \beta_{RI} RI_{i,t} + \sum_{k=1}^{K} \beta_k X_{k,i,t-1} + \sum_{j=1}^{3} \beta_{K+j} RT_{j,i,t} + \varepsilon_{i,t}$$

where I assume that countries have persistent differences in their mean level of respect for civil liberty $CL_{i,t}$ over time, modelled by deviations from a between-country intercept β_0 using the term α_i. These country-specific deviations have a normal distribution with a mean of 0 and a standard deviation σ_α. Note that these country deviations can be regarded as an error term, as they use those parts of deviations of observed values from predicted values that are stable over time.

In addition to the random country effects, I model global changes in the level of respect for a civil liberty over time by year-fixed effects λ_t. These are dummies for all years except 1946 (which is represented by the intercept). I assume that the effect β_P of personalism ($P_{i,t}$) is the same in every country i. Rigour and impartiality of the public administration ($RI_{i,t}$) and the control variables $X_{k,i,t\text{-}1}$ are also assumed to have the same effect (β_{RI}, β_k) in every country. To account for the fact that personalism is coded to the value on the first of January of year t, I lag the control variables by one year (to t - 1). The regime type variables with personalist regimes as the reference category ($RT_{j,i,t}$) are an exception, as these also have the first of January of year *t* as coding date.

While the random country intercepts can be regarded as a level 2 error term, the model also has a level 1 error term ($\varepsilon_{i,t}$) for the part of deviations from predicted values that changes over time. I assume these errors to follow a normal distribution with a mean of 0 and a different standard deviation $\sigma_{\varepsilon,i}$ in each country using Pustejovsky and Tipton's (2017) extension of MacKinnon and White's (1985) heteroscedasticity-robust standard errors. This means that the variances used to calculate the standard errors differ from the within-country variance reported for my models, since this within-country variance is estimated assuming homoscedasticity. The random country intercepts and standard error clustering correct for the dependence of prediction errors in the same country that result from persistent differences between countries, but not for dependence in prediction errors caused by the tendency for values in the same country to be more similar as they are closer in time.

For the civil liberties where I do not assume a mediation through the rigour and impartiality of the public administration, I will first estimate the random intercept model without year-fixed effects or predictors. In a second model I add the year-fixed effects. A third model adds the personalism variable. Then, I estimate a fourth model where I add the control variables (international conflict, internal conflict, political violence, population size, GDP per capita, GDP growth, and three regime type dummies). For the civil liberties where I do not assume a mediation, this fourth model is used to test my hypothesis of personalism being related to the civil liberty in question. The other models clarify this model by showing the stability of the relation to control variables and showing the distribution of variance between countries and within countries. In the case of freedom of expression, I also estimate a model with personalism and its square as predictors in addition to year-fixed effects to test my hypothesis of a curvilinear association. If this model suggests a curvilinear association between personalism and freedom of expression, I will incorporate that into the final model.

For the civil liberties where I can test mediation, I further estimate a fifth model where I not only include the control variables, but also the rigour and impartiality of the public administration. In these cases, this model tests my mediation hypothesis. In addition to the models with civil liberties as the dependent variable, I will estimate random intercept models with the rigour and impartiality of the public administration as dependent variable. Here, I first estimate a model with random country effects, add year-fixed effects in a second model, estimate a third model with personalism as predictor and add the control variables also used in the other regressions as a fourth step (international conflict, internal conflict, political violence, population size, GDP per capita, GDP growth, and three regime type dummies). Similar to the models for civil liberties without mediation by the rigour and impartiality of the public administration, I use the last model to test my hypothesis that personalism influences the rigour and impartiality of the public administration while the other models contextualise and clarify this model.

Results:

In this section, I will present the results of my analyses on the dataset introduced in the last section. I will first give univariate and bivariate descriptives of the dataset to aid interpretation of further results, and then move on to a discussion of the estimated models. R script for all analyses can be found in the Appendix.

Univariate statistics

To get an impression of univariate distributions of my variables, I will first discuss the distribution of over-time country means for the various variables. These give a first indication of the chances that the assumption of normally distributed country effects will be met and provide a succinct summary of the data, given that within-country observations have over-time dependence and are thus probably quite similar to each other and their mean. When describing the distribution of over-time country means I provide intra-country correlations, but to give more insight into the over-time developments in variables I will separately discuss some important over-time changes in the univariate distributions.

To start on the first point, the over-time country means are summarised by their mean, standard deviation, skew and kurtosis in Table 1. To get a first impression of missing data effects, these descriptive statistics are reported for both the full dataset and complete cases (where any country-year with a missing value on any of the variables is removed). On average, personalism is at a medium level of 0,40 among complete cases. There is a fair amount of variation around this mean, with a standard deviation of 0,21. The distribution of personalism is nearly symmetric (*Skew = 0,02*), and has fewer extreme values than a normally distributed variable (*Kurtosis* = -0,67). There is correlation in personalism within countries, but this intra-country correlation is actually not that large (*ICC* = 0,41). This suggests that the level of personalism in a given year tells something about the level of personalism in other years in the same country, but not so much as to have the level of personalism be very stable over long periods.

The mean of freedom of movement is not especially low or high (*M* > -0,01), that of freedom of assembly (*M* = -0,90) is somewhat low. Since these variables are constructed in such a way that their mean in the entire V-Dem dataset containing both autocracies and non-autocracies is zero, the mean level of freedom of assembly in autocracies is lower than the overall mean in both autocracies and non-autocracies. Given a theoretical 0-100 range, protection of life and physical integrity (*M* = 35,16) and freedom of expression (*M* = 28,93) are low. Between-country standard deviations are quite high for all civil liberties, as the range of the variables can be covered by less than six standard deviations while a normally distributed variable would have 99,7% (instead of 100%) of its observations within three standard deviations from the mean. This ratio would suggest lower kurtosis than a normal distribution if the variables were symmetrically distributed. However, freedom of movement and freedom of expression both have above-normal kurtosis (1,25 and 0,53), which is possible because both variables are skewed (-0,86 and 0,96 respectively). Protection of life and physical integrity and freedom of assembly have below-normal kurtosis (-0,30 and -0,20 respectively), but while freedom of assembly is closer to symmetry (*Skew* = 0,27) protection of life and physical assembly has clear right-skew (*Skew* = 0,72). The intra-country correlation of the civil liberties varies somewhat, from freedom of movement having rather strong over-time dependence (*ICC* = 0,76) to freedom of expression varying considerably more within countries over time (*ICC* = 0,46).

For the control variable, it is noteworthy that the mean population level in the countries in the complete dataset over time is approximately 23.000.000 people ($\overline{10^{\log_{10}(Population)}} \approx$ 23.294.375) and the over-time mean GDP per capita across all countries is approximately \$4.500 ($\overline{e^{\ln(GDP\ per\ capita)}} \approx$ \$4.423,96) in 2011 US dollars (Bolt et al., 2018), while the average over-time mean GDP growth per year is 2% (*M* = 0,02). The average level of rigour and impartiality of the public administration in the autocratic countries in the complete dataset is somewhat below the average level in all (autocratic and non-autocratic) countries in the V-Dem dataset (*M* = -0,60). Population levels are very strongly related within one and the same country (*ICC* = 0,90), while the intra-country correlation of GDP growth is very low (*ICC* = 0,02). GDP growth also has a rather large number of average GDP growth levels far from the country mean (*Kurtosis* = 1,82). Figure 3 shows the by-year distribution of personalism using a heatmap that bins observations by value-year combinations (like a bivariate histogram), and this suggests that very large outliers (of up to 272%) that distort the mean of the countries with those outliers might explain the high kurtosis. Lastly, the distribution of over-time country

Figure 3

Heatmap of GDP growth over time for complete cases

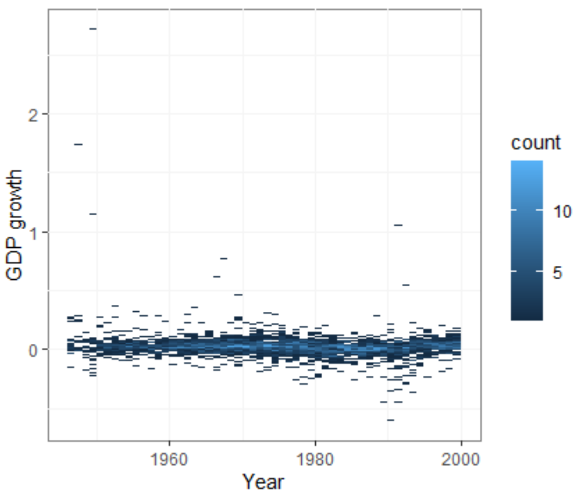


Table 1

Comparison of descriptive statistics between complete cases and the full dataset (M = mean, SD = standard deviation, S = skew, K = kurtosis, ICC = intra-country correlation)

Variable	Data	*M*	*SD*	*S*	*K*	*ICC*
Personalism	Full	0,41	0,21	-0,01	-0,55	0,52
	Complete	0,40	0,21	0,02	-0,67	0,55
Freedom of movement	Full	-0,01	1,08	-0,73	1,16	0,82
	Complete	>-0,01	1,13	-0,86	1,25	0,86
Freedom of expression	Full	29,28	18,26	0,87	0,48	0,60
	Complete	28,93	18,85	0,96	0,53	0,65
Freedom of assembly	Full	-0,91	1,01	0,19	-0,24	0,71
	Complete	-0,90	1,00	0,27	-0,20	0,71
Protection of life and physical integrity	Full	35,11	21,03	0,75	-0,15	0,71
	Complete	35,16	21,77	0,72	-0,30	0,75
Political violence	Full	-0,03	1,28	-0,05	-0,71	0,78
	Complete	-0,07	1,32	0,02	-0,70	0,81
$\log_{10}$ (Population)	Full	6,88	0,55	0,48	0,71	0,94
	Complete	6,89	0,55	0,44	0,68	0,94
ln(GDP per capita)	Full	8,06	0,82	0,40	-0,24	0,79
	Complete	7,99	0,81	0,29	-0,31	0,83
GDP growth	Full	0,02	0,02	0,49	0,20 [a]	0,04
	Complete	0,02	0,03	-0,24	1,82	0,04
Rigour and impartiality public administration	Full	-0,58	0,96	0,54	0,33	0,71
	Complete	-0,60	0,97	0,43	0,08	0,73

Note: Kurtosis is excess kurtosis, such that the normal distribution has a kurtosis of 0.
[a] The comparatively low kurtosis of GDP growth for the full dataset results from a number of countries with outliers being dropped from the calculation due to ill-defined means.

means of political violence is notable because it has far lower kurtosis than a normally distributed variable (*Kurtosis* = -0,70).

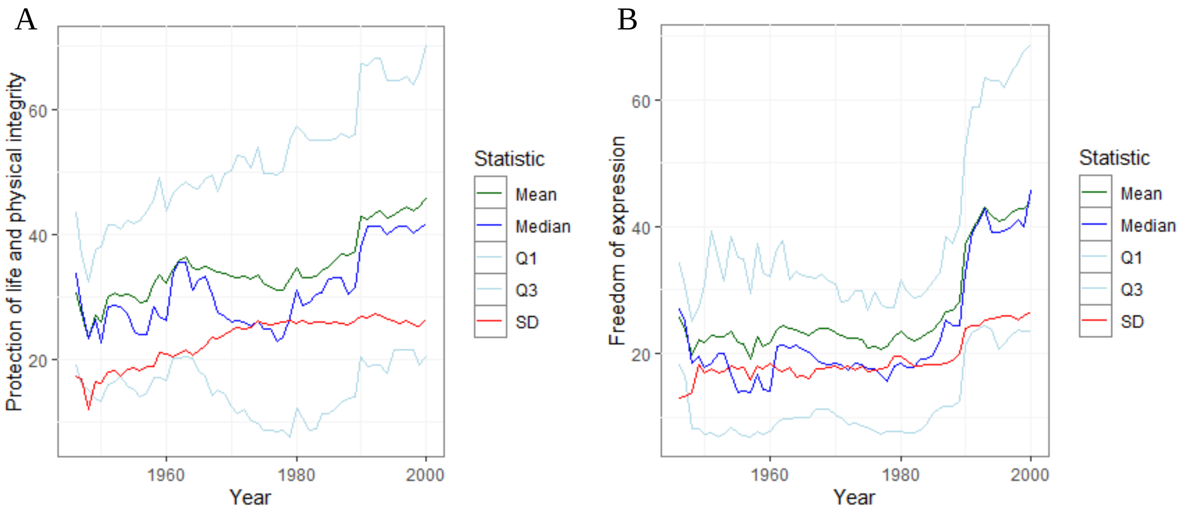
To expand on the intra-country correlations reported in Table 1, I will also describe two important over-time changes in the variables. The most important change in this regard is shown in Figure 4, which depicts how the yearly summary statistics of protection of life and physical integrity and freedom of expression evolve over time. Around the end of the Cold War in 1990, there is a clear upward shift in freedom of expression from approximately 0,2 to approimately 0,4. Protection of life and physical integrity also increases, but comparatively less. Generally, the civil liberties increase to varying extents around the end of the Cold War.

Interestingly, there is no clear shift in the proportion of single-party regimes around the end of the Cold War (See the Appendix, lines 2560 to 2606). A second point of interest is that the distribution of personalism moves steadily upward throughout the time period, starting at a mean of approximately 0,3 and ending up close to 0,5 by 2000.

As Table 1 suggests, the differences between the full dataset and complete cases are minor, with an important exception. Missingness of internal conflict and international conflict means that the complete cases only cover the period from 1946 to 2000 instead of running until 2010. This means that the data available cannot be used for conclusions about the early twenty-first century and developments therein.

Figure 4

Summary statistics of protection of life and physical integrity and freedom of expression over time for complete cases



Note: Panel A shows protection of life and physical integrity, Panel B shows freedom of expression. The mean is shown in dark green, the median in dark blue, the first and third quartile in light blue and the standard deviation in red.

Bivariate statistics

Although modelling will focus on variation within countries, accessing this variation without having intra-country correlation distort results is rather difficult without using a formal modelling approach. This section will only present the bivariate correlation between over-time country means (between-country correlations) in Table 2. I will focus on theoretically relevant correlations that are significant at the five percent level, which is any correlation of at least 0,19 for 109 independent observations.

Countries that have a higher average level of any one civil liberty also tend to have high values for other civil liberties ($0,44 \leq r \leq 0,78$). This means that the effect of personalism on any one civil liberty might be distorted when not controlling for the effect of other civil liberties. Beyond the correlations among the civil liberties, higher mean levels of personalism tend to be accompanied by lower average protection of life and physical integrity (r = -0,24), while the associations between personalism and the other civil liberties are statistically indistinghuishable from zero. Greater rigour and impartiality of the public administration over time has a strong association with greater protection of life and physical integrity over time (r = 0,61). Countries with higher over-time means for the rigour and impartiality of the public administration also have more freedom of movement (r = 0,36), greater freedom of expression (r = 0,44), and greater freedom of assembly (r = 0,38). Although the association is in the expected direction, mean rigour and impartiality of the public administration is not significantly associated with mean levels of personalism (r = -0,14).

As expected, countries with more years under personalist regimes have higher average personalism (r = 0,44), although personalistic regimes and regimes with high levels of personalism are still statistically distinghuishable. Countries with more years under military or single-party regimes are less personalistic (r = -0,27 and r = -0,23 respectively). The average levels of personalism in countries under a monarchy for differing lengths of time are statistically indistinghuishable (r = 0,06). Beyond the regime type variables, other control variables do occasionally have sizeable correlations with civil liberties, but seldomly have a strong correlation to personalism. For example, political violence is strongly related to protection of life and physical integrity (r = -0,52), but too weakly related to personalism to say anything sensible about the relation (r = -0,01). This suggests that these control variables will probably not be particularly important in controlling the relation between personalism and civil liberties. At the same time, the between-country correlations in Table 2 might not accurately represent the within-country associations.

Table 2

Bivariate associations between over-time country means for complete cases

Variable	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.	16.
1. Personalism	-															
2. Freedom of movement	-0,13	-														
3. Freedom of expression	-0,18	0,72	-													
4. Freedom of assembly	-0,17	0,58	0,78	-												
5. Protection of life and physical integrity	-0,24	0,44	0,57	0,51	-											
6. Political violence	-0,01	-0,11	-0,15	-0,14	-0,52	-										
7. $\log_{10}$ (Population)	-0,15	-0,14	-0,13	-0,20	-0,22	0,21	-									
8. ln (GDP per capita)	-0,23	0,06	-0,02	-0,10	0,32	-0,27	-0,10	-								
9. GDP growth	-0,19	-0,05	-0,13	-0,12	0,14	-0,14	0,01	0,33	-							
10. Rigour and impartiality public administration	-0,14	0,36	0,44	0,38	0,61	-0,47	-0,20	0,18	0,25	-						
11. International conflict	-0,03	-0,04	-0,02	-0,02	-0,04	0,17	0,07	0,05	0,09	0,13	-					
12. Internal conflict	-0,01	-0,12	-0,13	-0,22	-0,30	0,42	0,24	-0,14	-0,04	-0,34	-0,02	-				
13. Personalist regime	0,44	0,26	0,16	0,08	-0,16	0,14	-0,06	-0,18	-0,35	-0,11	0,08	0,09	-			
14. Military regime	-0,27	0,02	-0,12	-0,15	-0,24	0,27	0,11	0,11	0,15	-0,26	-0,13	0,11	-0,22	-		
15. Single-party regime	-0,23	-0,20	<0,01	0,06	0,18	-0,17	0,09	-0,08	0,04	0,21	-0,02	-0,13	-0,60	-0,40	-	
16. Monarchy	0,06	-0,07	-0,10	-0,05	0,20	-0,23	-0,20	0,25	0,26	0,11	0,08	-0,04	-0,20	-0,17	-0,32	-

Note: n is 109, correlations larger than 0,19 are significant at $\alpha = 0,05$

Regression modelling

Freedom of expression

The first civil liberty for which I will discuss the regression models is freedom of expression. I will report the results for the first two models with random country intercepts and random country intercepts plus year-fixed effects in running text, while the models adding theoretically important predictors are also reported in Table 3.

Model 1 containing only the random country intercepts can be used to see which variation in freedom of expression is attributable to persistent between-country differences, and which variation lies in over-time changes within countries. The estimates suggests that most variation is between-country variance at 330,36, while pooled within-country variance is lower at 177,95. The variance decomposition implies an intra-country correlation of 0,65, while the between-country intercept is 28,81 (*SE* = 1,77, 98,75% CI [24,33; 33,30]). Since Model 1 does not contain any regressors, these statistics should and do approximately match the statistics reported in Table 1 (*M* = 28,93, *ICC* = 0,65).

Adding the year-fixed effects in Model 2 significantly improves the model fit ($\chi^2(54)$ = 789,67, *p* < 0,001). This is also reflected in the decrease in both between-country variance and within-country variance from Model 1 to Model 2. There is a proportional decrease by 12,24 percent in variance, calculated by taking subtracting the total variance in Model 2 compared to that in Model 1 from one ($R_1^2 = 1 - (\hat{\sigma}_{\varepsilon,2} + \hat{\sigma}_{\alpha,2})/(\hat{\sigma}_{\varepsilon,1} + \hat{\sigma}_{\alpha,1})$; Snijders & Bosker, 2012, p. 112). The intra-country correlation is a little higher than in Model 1 at 0,68. The mean level of freedom of expression rises significantly after 1990, with the 1946 mean level being 27,61 (*SE* = 3,64, 98,75% CI [17,59; 37,62]) while the 2000 mean level is 27,61 + 15,26 ≈ 42,86 (*b* = 15,26, *SE* = 4,51, 98,75% CI [2,82; 27,69]).

Model 3 is the first model reported in Table 3, and adds the personalism variable. Countries with the highest possible level of personalism are predicted to have a level of freedom of expression that is 17,03 points lower (*SE* = 3,46, 98,75% CI [-26,28; -9,94]) than countries with the lowest possible level of personalism, corrected for the mean level of freedom of expression in that country over time and the mean level of freedom of expression across countries in a year. I deem this a small-to-medium sized effect, since a country that has the lowest possible level of personalism instead of the highest possible level would in expectation have only about a within-country standard deviation’s lower level of freedom of assembly. Model 3 significantly improves model fit compared to Model 2 ($\chi^2(1)$ = 239,68, *p* < 0,001). The year-fixed effects still improve the model fit, given that personalism is in the model ($\chi^2(54) = 898{,}47$, $p < 0{,}001$). The decrease in total compared to Model 1 is 17,85 percent, and since the intra-country correlation is still approximately 0,68 between-country and within-country variance decreased roughly equally.

Model 4 adds the square of personalism to the model with just a linear personalism term. This squared term only minimally decreases total variance compared to Model 3 ($R_1^2 = 0{,}19$), while neither the linear personalism term ($b = -7{,}29$, $SE = 8{,}22$, 98,75% CI [-28,62; 14,04]) nor the quadratic term ($b = -11{,}20$, $SE = 9{,}62$, 98,75% CI [-36,36; 13,96]) is significant. Since the differences with Model 3 are minimal and the quadratic term is insignificant, my hypothesis of a curvilinear association between personalism and freedom of expression seems implausible. The residual analysis of Model 5 will provide yet more evidence against the hypothesis of curvilinearity.

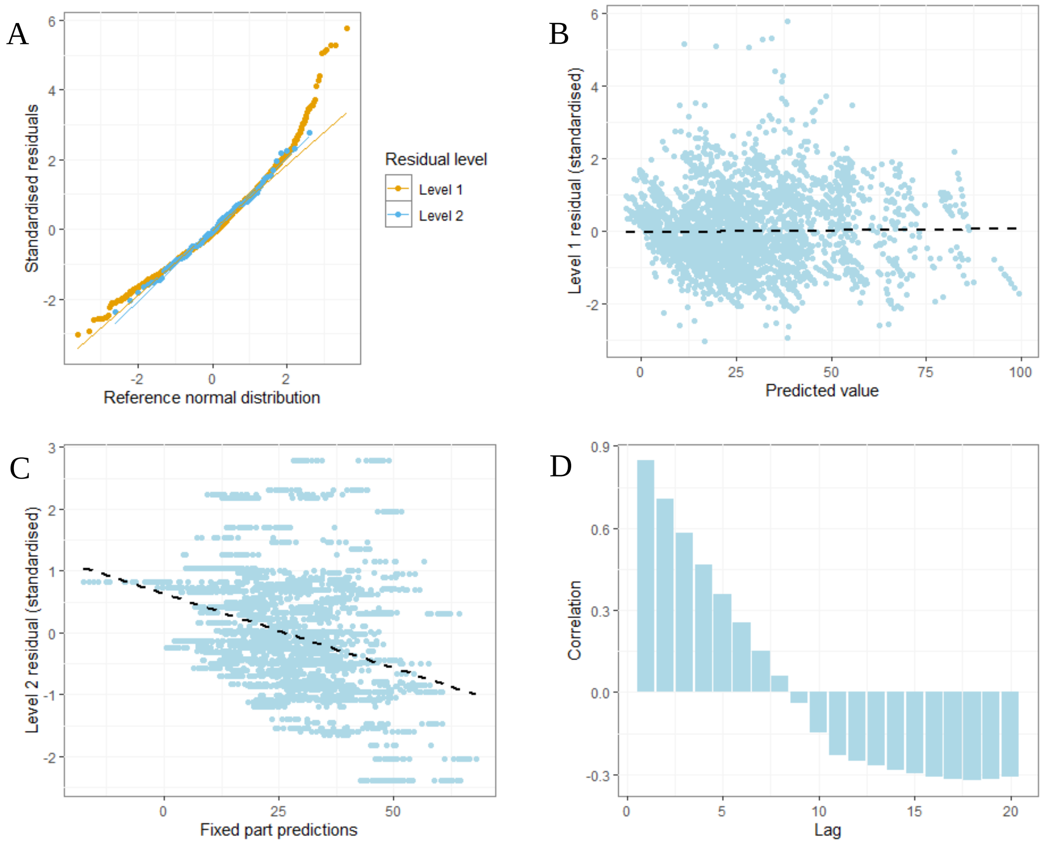
Model 5 adds political violence, logged population, logged GDP per capita, GDP growth, internal and international conflict and regime type as control variables. The effect of personalism changes little with the addition of the control variables ($b = -18{,}11$, $SE = 3{,}14$, 98,75% CI [-25,38; -8,71]), while none of the control variables is significant at $\alpha = 0{,}0125$. The control variables do seem to improve model fit together compared to Model 3 ($\chi^2(9) = 177{,}73$, $p < 0{,}001$), but between-country variance increases raising the intra-country correlation to 0,72. The total decrease in variance compared to Model 1 is 10,71 percent, less than in Model 3 to such an extent that the model might be misspecified (Snijders & Bosker, 2012, p. 156). The year-fixed effects still contribute to model fit ($\chi^2(54) = 718{,}45$, $p < 0{,}001$).

I estimated a model adding the rigour and impartiality of the public administration (lines 3585 to 3646 in the Appendix) to test my assumption that this would not be a meaningful mediating variable for freedom of expression. My assumption seems questionable, as the effect of personalism in this model is decreased ($b = -11{,}38$, $SE = 3{,}02$, 98,75% CI [-19,23; -3,52]) while the rigour and impartiality of the public administration has a strong association with freedom of expression ($b = 10{,}00$, $SE = 1{,}33$, 98,75% CI [6,52; 13,47]).

To assess the validity of Model 4, I analysed the residuals of the model. The residuals can be separated into two ‘levels’. The level 2 residuals are differences between general intercept and the country-specific intercepts, showing the distribution of the random intercept term. The level 1 residuals stem from differences between predicted values (including random effects) and observed values. The level 1 residuals are intuitively more similar to the residuals of a ‘normal’ single-level model. The diagnostic plot for the level 1 residuals shown in Panel B of Figure 5 suggest that a straight-line approximation of the relation between freedom of expression and the predictors (including personalism) does not lead to systematic errors in my predictions. This conflicts with the hypothesis of a curvilinear association. However, the residuals are right-skewed as shown in Panel A, which means that my assumption of normally distributed errors used to calculate confidence intervals is somewhat questionable. Panel C shows that standardised level 2 residuals are related to predicted values, which suggests that persistent differences in freedom of expression are related to one or mode predictors. Panel D shows strong residual autocorrelation for the level 1 residual, which means that prediction errors for the same country in one year and subsequent years are very similar, violating the assumption of independent errors.

Figure 5

QQ-plot, predicted value-standardised residual plots and autocorrelation plot for freedom of expression.



Note: The standardised residuals are standardised with country-level standard

Table 3

Linear random intercept model with freedom of expression as dependent variable and year-fixed effects (N = 3406, n = 109, T = 1-55).

	Model 3		Model 4		Model 5	
Variable	b (SE) [a]	p	b (SE) [a]	p	b (SE) [a]	p
Intercept	33,36 (3,30)	<0,01	26,39 (2,91)	<0,01	127,16 (49,86)	0,01
Personalism	-17,03 (3,46)	<0,01	-7,29 (8,22)	0,38	-18,11 (3,14)	<0,01
Personalism^2			-11,20 (9,62)	0,25		
Political Violence					0,70 (1,29)	0,59
$\log_{10}$(Population)					-13,46 (7,15)	0,07
ln(GDP per cap)					-0,42 (2,39)	0,86
GDP Growth					-2,39 (2,10)	0,28
Internal conflict (1 = conflict)					-2,13 (1,60)	0,19
International conflict (1 = conflict)					-1,72 (1,44)	0,25
Regime type						
Personalist (ref)						
Military					-2,64 (3,27)	0,43
Single-party					-5,86 (4,74)	0,22
Monarchy					9,98 (6,43)	0,16
Variance						
Between-country	284,46		280,89		327,97	
Within-country	133,11		132,69		125,90	
R_1^2 [b]	0,18		0,19		0,11	
ICC	0,68		0,68		0,72	
$LR\ \chi^2$ Year-fixed [c]	898,47	<0,01	898,49	<0,01	697,50	<0,01
$LR\ \chi^2$ [d]	239,68	<0,01	12,92	<0,01	177,73	<0,01

Note: [a] The standard errors are country-clustered. [b] Proportional decrease in level 1 variance compared to Model 1. [c] Compares the model with year effects to an equivalent model without them. [d] Compares Model 3 to Model 2, Model 4 to Model 3 and Model 5 to Model 3.

Freedom of assembly

As with freedom of expression, the first models are only reported in running text. Model 1 includes only the random country intercepts. Between-country variance is 0,96 and within-country variance is 0,39. Most of the variance in freedom of assembly is thus located between countries, making freedom of assembly fairly stable over time within countries (*ICC* = 0,71). The general mean of freedom of assembly is -0,90 (*SE* = 0,09, 98,75% CI [-1,14; -0,66]), as is to be expected from Table 1. Model 2 adds year-fixed effects to the random country intercepts, which significantly improves model performance ($\chi^2(54) = 519{,}49$, $p < 0{,}001$) although the individual comparisons between 1946 ($b = -0{,}76$, $SE = 0{,}19$, 98,75% CI [-1,29; -0,24]) and later years rarely point to individually significant differences. Unexplained variance is lowered by taking changes over time into account ($R_1^2 = 0{,}07$), but the ratio of between-country to within-country variation is more or less the same ($ICC = 0{,}73$).

Model 3 adds personalism as a continuous predictor, and this significantly improves the model fit ($\chi^2(1) = 156{,}10$, $p < 0{,}001$) with a total variance decrease of 11,14 percent compared to Model 1. Countries at the highest level of personalism tend to have a value for freedom of assembly that is 0,67 points lower ($SE = 0{,}17$, 98,75% CI [-1,12; -0,22]) than that of countries at the lowest level of personalism, given the long-term mean of the country and the mean level of freedom of assembly in a year. Again, this seems like a small-to-medium-sized effect. Model 4 controls the effect of personalism for a selection of control variables in addition to a country's persistent deviation from the cross-country mean level of freedom of assembly and year-fixed effects. These control variables seem to significantly improve model fit together ($\chi^2(9) = 248{,}51$, $p < 0{,}001$). However, none of the individual coefficients differs significantly from zero and the variance is substantially greater than that of Models 2 or 3 ($R_1^2 = 0{,}02$). The relation between freedom of assembly and personalism is somewhat stronger in Model 4 than in Model 3 ($b = -0{,}71$, $SE = 0{,}16$, 98,75% CI [-1,12; -0,30]), but the difference is not statistically or substantively meaningful. Models 3 and 4 fit the hypothesised decrease in freedom of assembly with greater personalism.

As with freedom of expression, I also estimated a model including the rigour and impartiality of the public administration for freedom of assembly (See lines 3951 to 4010 in the Appendix). Again, the rigour and impartiality of the public administration has a statistically significant association with freedom of assembly ($b = 0{,}42$, $SE = 0{,}06$, 98,75% CI [0,27; 0,57]) while adding that variable makes the association of personalism to freedom of assembly substantially weaker ($b = -0{,}43$, $SE = 0{,}15$, 98,75% CI [-0,81; -0,05]).

The most important parts of the residual analysis for Model 4 of freedom of assembly are shown in Figure 6. Panel A shows that the level 2 residuals are fairly normally distributed. The level 1 residuals have a rather stretched right tail, although the distribution is fairly close to a normal distribution otherwise. Panel B suggests that residual values still have fairly strong persistence over time. Further predicted value-residual plots of the level 1 and level 2 residuals showed the level 1 residuals to conform to the linear model fairly well while level 2 residuals have a clear downward slope.

Table 4

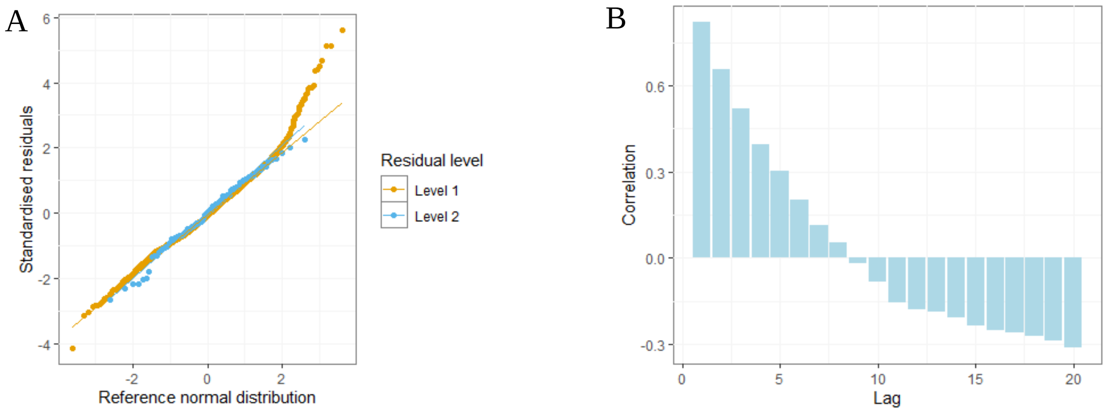
Linear random intercept model with freedom of assembly as dependent variable and year-fixed effects (N = 3406, n = 109, T = 1-55).

	Model 3		Model 4	
Variable	b (SE) [a]	p	b (SE) [a]	p
Intercept	-0,54 (0,20)	0,01	4,69 (2,92)	0,11
Personalism	-0,67 (0,17)	<0,01	-0,71 (0,16)	<0,01
Political Violence			-0,12 (0,07)	0,08
$\log_{10}$(Population)			-0,80 (0,41)	0,06
ln(GDP per cap)			0,01 (0,12)	0,94
GDP Growth			-0,08 (0,12)	0,54
Internal conflict (1 = conflict)			0,06 (0,09)	0,52
International conflict (1= conflict)			0,04 (0,10)	0,73
Regime type				
Personalist (ref)				
Military			-0,21 (0,15)	0,17
Single-party			-0,20 (0,19)	0,31
Monarchy			0,78 (0,31)	0,04
Variance				
Between-country	0,87		1,01	
Within-country	0,32		0,30	
R_1^2 [c]	0,11		0,03	
ICC	0,73		0,77	
$LR\ \chi^2$ Year-fixed [b]	562,54	<0,01	570,00	<0,01
$LR\ \chi^2$ [d]	156,10	<0,01	248,51	<0,01

Note: [a] Country-clustered standard errors. [b] Compares model with year effects to an equivalent model without them. [c] Proportional decrease in level 1 variance compared to Model 1. [d] Compares each model to the previous model.

Figure 6

QQ-plot and predicted value-standardised residual scatterplot for freedom of assembly.



Note: The standardised residuals are standardised with country-level standard deviations.

Protection of life and physical integrity

For protection of life and physical integrity, Model 1 including only random country intercepts estimates between-country variance to be 460,35 while within-country variance is 153,77. This suggests that variation in protection of life and physical integrity mostly occurs between countries (*ICC* = 0,75). The general mean for protection of life and physical integrity is 35,25 (*SE* = 2,07, 98,75% CI [29,98; 40,52]), close to that in Table 1. While the year-fixed effects added in Model 2 significantly improve model fit ($\chi^2(54) = 199{,}63$, $p < 0{,}001$), individual differences compared to 1947 (*b* = 39,16, *SE* = 4,58, 98,75% CI [26,72; 51,60]) are not significant. Both within-country variance (147,23) and between-country variance (452,32) are a little lower in Model 2 than in Model 1 ($R_1^2 = 0{,}02$). These variances give an intra-class correlation of 0,75.

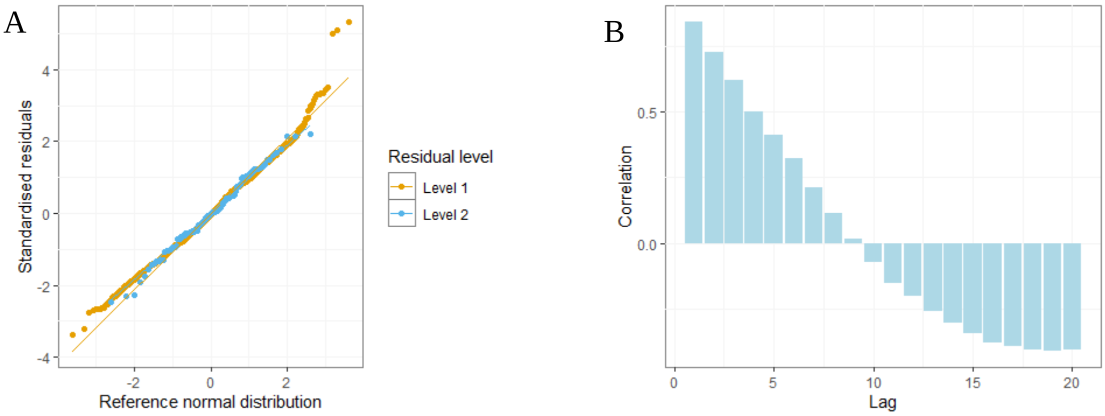
Model 3 shows that countries with the highest possible level of personalism have a level of protection of life and physical integrity that is 17,48 points lower on average (*SE* = 3,94, 98,75% CI [-27,70; -7,26]) than the level in countries with the lowest possible level of personalism, given a country's persistent influences on protection of life and physical integrity and the mean level of protection of life and physical integrity in a year across countries. This effect is somewhat small, taking into account that the estimated coefficient represents the largest difference personalism can be associated with. Model 4 introduces control variables into the specification, and these control variables together significantly contribute to improving model fit ($\chi^2(9) = 481{,}15$, $p < 0{,}001$). The relation between personalism and protection of life and physical integrity in the Model 4 remains fairly similar to that in Model 3 (*b* = -16,96, *SE* = 3,47, 98,75% CI [-25,97; -7,95]), now controlling for the control variables as well as across-country year means and country effects. Overall, this is in line with my hypothesis that greater personalism is associated to less protection of life and physical integrity.

Model 5 adds the rigour and impartiality of the public administration. Greater levels of rigour and impartiality of the public administration tend to be accompanied by greater protection of life and physical integrity (*b* = 11,23, *SE* = 1,53, 98,75% CI [7,23; 15,24]) controlling for the level of personalism, the values for the control variables in the previous year, time-persistent differences between countries and the mean level of protection of life and physical integrity in a year. This is a rather large effect, as a shift from the minimum to the maximum of the range for the rigour and impartiality of the public administration (as with personalism) would amount to a shift of 74,95 points on the 100-point theoretical range in protection of life and physical integrity. The relation between personalism and protection of life and physical integrity is almost halved (b = -9,33, SE = 2,78, 98,75% CI [-16,54; -2,11]), suggesting that this relation runs in part through the rigour and impartiality of the public administration. Model 5 has significantly improved fit compared to Model 4 ($\chi^2(1)$ = 1146,39, p < 0,001), with a decrease in total variance compared to Model 1 by 34,64 percent.

Analysing the residuals of Model 5 for protection of life and physical integrity, both the level 1 residuals and the level 2 residuals seem fairly normally distributed judging by Panel A of Figure 7. The level 1 residuals do seem to have a slight right skew. The autocorrelation plot of Panel B suggests that values within the same country remain strongly related over time. Predicted value-residual plots of the level 1 and level 2 residuals showed a fairly strong downward slope in the level 2 residuals.

Figure 7

QQ-plot and autocorrelation plot for protection of life and physical integrity.



Note: The standardised residuals are standardised with country-level standard deviations.

Table 5

Linear random intercept model with protection of life and physical integrity as dependent variable and year-fixed effects (N = 3406, n = 109, T = 1-55).

	Model 3		Model 4		Model 5	
Variable	b (SE) [a]	p	b (SE) [a]	p	b (SE) [a]	p
Intercept	45,08 (4,68)	<0,01	131,78 (45,18)	0,01	173,37 (38,53)	<0,01
Personalism	-17,48 (3,94)	<0,01	-16,96 (3,47)	<0,01	-9,33 (2,78)	<0,01
Rig. & Impart. Pub. Admin.					11,23 (1,53)	<0,01
Political Violence$_{t-1}$			-4,36 (1,09)	<0,01	-2,44 (1,00)	0,02
$\log_{10}$(Population)$_{t-1}$			-15,08 (6,64)	0,04	-17,13 (5,60)	0,01
ln(GDP per cap)$_{t-1}$			1,27 (2,03)	0,54	-1,89 (1,73)	0,28
GDP Growth			-5,38 (2,51)	0,16	-4,02 (1,88)	0,11
Internal conflict (1 = conflict)$_{t-1}$			-1,59 (1,46)	0,26	-0,20 (1,25)	0,74
International conflict (1= conflict)$_{t-1}$			-1,00 (1,44)	0,29	-1,25 (1,69)	0,29
Regime type$_{t-1}$						
Personalist (ref)						
Military			-1,75 (4,06)	0,67	-2,45 (3,64)	0,57
Single-party			1,75 (5,07)	0,73	0,20 (3,50)	0,93
Monarchy			18,44 (5,53)	0,01	9,74 (3,60)	0,05
Random effect						
Between-country	423,04		333,09		316,61	
Within-country	137,09		119,74		84,75	
R_1^2 [c]	0,09		0,26		0,35	
ICC	0,76		0,74		0,79	
$LR\ \chi^2$ Year-fixed [b]	248,25	<0,01	323,82	<0,01	320,39	<0,01
$LR\ \chi^2$ [d]	243,69	<0,01	481,15	<0,01	1146,39	<0,01

Note: [a] Country-clustered standard errors. [b] Compares model with year effects to an equivalent model without them. [c] Proportional decrease in level 1 variance compared to Model 1. [d] Compares each model to the previous model.

Freedom of movement

The first two models are once again reported in the running text only. Model 1 with only random country intercepts for freedom of movement estimates within-country variance to be 0,20 while the estimated between-country variance is 1,23 (*ICC* = 0,86). The between-country mean does not differ significantly from zero (b < 0,01, SE = 0,11, 98,75% CI [-0,27; 0,27]). Model 2 introduces year-fixed effects, which contribute significantly to model fit (χ^2(54) = 382,52, p < 0,001) and show that the mean level of freedom of movement is significantly higher than it was in 1946 from 1990 onwards. Model 2 has a within-group variance of 0,18 and a between-group variance of 1,22, giving an intra-country correlation of 0,87 and decreasing total variance by 2,37 percent (R_1^2 = 0,02) compared to Model 1.

The first model reported in Table 6 is Model 3, which introduces the personalism variable. Countries with the highest level of personalism are estimated to have a level of freedom of movement that is 0,40 points lower (*SE* = 0,12, 98,75% CI [-0,70; -0,10]) than countries with the lowest possible level of personalism, given time-persistent differences between countries and across-country yearly changes in freedom of movement. This is a small-to-medium effect, taking into account the variation in freedom of movement and the fact that it compares the extremes of the personalism scale. The decrease in total variance compared to Model 1 is just 4,30 percent.

Model 4 is second in Table 6, and this model adds the lagged control variables. None of these control variables has a statistically significant relation to freedom of movement at α = 0,0125. The control variables do seem to contribute to model fit together ($\chi^2(9)$ = 405,26, $p <$ 0,001), but the decrease in level 1 variance compared to Model 1 is just 5,65 percent. The estimated relation of personalism to freedom of movement rises somewhat (*b* = -0,47, *SE* = 0,10, 98,75% CI [-0,74; -0,20]) when controlling for the control variables in addition to time-persistent differences between countries and across-country developments in a year. The negative relation between personalism and freedom of movement fits my hypothesis.

Model 5 is the final model in Table 6, adding the rigour and impartiality of the public administration to test for mediation. The total variance of this model is quite a lot lower than that of Model 1 (R_1^2 = 0,18). Higher levels of rigour and impartiality of the public administration tend to be accompanied by greater freedom of movement (*b* = 0,34, *SE* = 0,06, 98,75% CI [0,19; 0,50]) controlling for personalism, the control variables, across-country changes by year and time-persistent differences between countries. This effect is rather large and larger than that of personalism. Controlled for the rigour and impartiality of the public administration as well as time-persistent differences between countries and across-country changes by year, the effect of personalism becomes smaller and statistically indistinguishable from zero (*b* = -0,24, *SE* = 0,10, 98,75% CI [-0,49; 0,01]). However, the confidence intervals for personalism in Model 4 and Model 5 still contain the point estimate of the other model, such that this change is not that meaningful.

For the residual diagnostics of freedom of movement, Panel A of Figure 8 shows that the level 1 residuals have a great deal of excess kurtosis. This means that the standard errors calculated for Model 5 (and probably the other models) are under-estimated. The level 2 residuals have a stretched lower tail, but resemble a normally distributed variable a lot more. The autocorrelation plot of Panel B suggests that values within the same country remain strongly related over time. Predicted value-residual plots of the level 1 and level 2 residuals showed a slight downward slope in the level 2 residuals.

Table 6

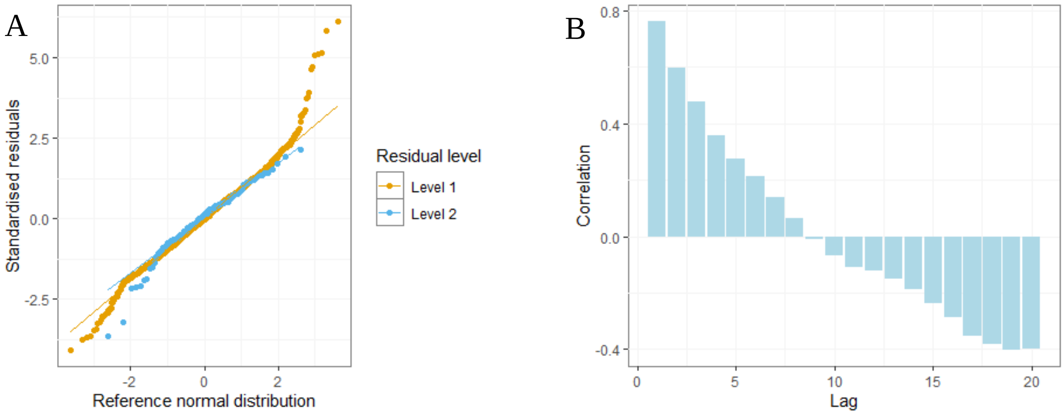
Linear random intercept model with freedom of movement as dependent variable and year-fixed effects (N = 3251, n = 107, T = 1-54).

	Model 3		Model 4		Model 5	
Variable	*b* (*SE*) [a]	*p*	*b* (*SE*) [a]	*p*	*b* (*SE*) [a]	*p*
Intercept	0,06 (0,14)	0,68	1,00 (3,47)	0,77	2,37 (2,78)	0,40
Personalism	-0,40 (0,12)	<0,01	-0,47 (0,10)	<0,01	-0,24 (0,10)	0,01
Rig. & Impart. Pub. Admin.					0,34 (0,06)	<0,01
Political Violence			-0,10 (0,05)	0,03	-0,04 (0,04)	0,31
$\log_{10}$(Population)			-0,33 (0,49)	0,50	-0,38 (0,39)	0,34
ln(GDP per cap)			0,19 (0,11)	0,08	0,10 (0,09)	0,27
GDP Growth			-0,11 (0,12)	0,40	-0,07 (0,09)	0,49
Internal conflict (1 = conflict)			-0,15 (0,07)	0,03	-0,11 (0,05)	0,05
International conflict (1= conflict)			-0,08 (0,04)	0,09	-0,08 (0,04)	0,06
Regime type						
Personalist (ref)						
Military			-0,26 (0,09)	0,01	-0,28 (0,09)	<0,01
Single-party			-0,37 (0,14)	0,01	-0,41 (0,12)	<0,01
Monarchy			0,24 (0,34)	0,49	-0,03 (0,19)	0,89
Random effect						
Between-country	1,20		1,20		1,05	
Within-country	0,18		0,16		0,12	
R_1^2 [c]	0,04		0,06		0,18	
ICC	0,87		0,88		0,89	
LR χ^2 Year-fixed [b]	433,70	<0,01	277,68	<0,01	261,13	<0,01
LR χ^2 [d]	99,89	<0,01	405,26	<0,01	762,32	<0,01

Note: [a] Country-clustered standard errors. [b] Compares model with year effects to an equivalent model without them. [c] Proportional decrease in level 1 variance compared to Model 1. [d] Compares each model to the previous model.

Figure 8

QQ-plot and autocorrelation plot for freedom of movement.



Note: The standardised residuals are standardised with country-level standard deviations.

Rigour and impartiality of the public administration

In the case of the rigour and impartiality of the public administration, Model 1 with only random country intercepts has a within-country variance of 0,33 and a between-country variance of 0,91 (*ICC* = 0,73). The between-country mean level of the rigour and impartiality of the public administration is -0,60 (*SE* = 0,09, 98,75% CI [-0,84; -0,37]). Model 2 adds year-fixed effects, but in this case doing so does little to improve model fit ($\chi^2(54) = 76{,}85$, $p = 0{,}022$). Indeed, total variance in Model 2 is just 0,14 percent lower than in Model 1.

Model 3 adds personalism, and this does improve model fit ($\chi^2(1) = 199{,}28$, $p < 0{,}001$) with the level 1 variance decreasing by 3,23 percent. The point estimate estimates the rigour and impartiality to be 0,75 lower (*SE* = 0,15, 98,75% CI [-1,14; -0,35]) in countries with the highest possible level of personalism as compared to the lowest possible level of personalism controlling for time-persistent differences between countries and across-country yearly changes. This is a small effect, and the estimate is rather uncertain. The year-fixed effects still do not significantly contribute to model fit ($\chi^2(54) = 70{,}79$, $p = 0{,}062$).

Model 4 controls the relation between personalism and the rigour and impartiality of the public administration for the same control variables used in the other models. Total variance decreases quite strongly ($R_1^2 = 0{,}16$). The estimated relation between personalism and the rigour and impartiality of the public administration controlling for time-persistent country differences and across-country year effects as well as these control variables is somewhat smaller than that in Model 3 (*b* = -0,67, *SE* = 0,15, 98,75% CI [-1,06; -0,28]), but due to the uncertainty of the estimate this change does not seem overly meaningful. Higher levels of political violence in the previous year are associated with lower levels of rigour and impartiality of the public administration in this year (*b* = -0,17, *SE* = 0,06, 98,75% CI [-0,32; -0,02]) given the level of personalism, the other control variables, time-persistent differences between countries and across-country differences over years, but once again the size of the effect is hard to determine. In this model the year-fixed effects are statistically significant (χ^2(54) = 109,65, *p* < 0,001), but since they only first become so in this model it is questionable how much weight should be given to that.

The level 1 residuals of Model 5 of the rigour and impartiality of the public administration seem to be right-skewed, shown in Panel A of Figure 9. Although less severe, such a right-skew also seems to appear in the level 2 residuals. Panel B of Figure 9 shows the autocorrelation plot of level 1 residuals, and this suggests that residuals are still very strongly related to each other over time within a country. Predicted value-residual plots of the level 1 and level 2 residuals showed a slight downward slope in the level 2 residuals.

Table 7

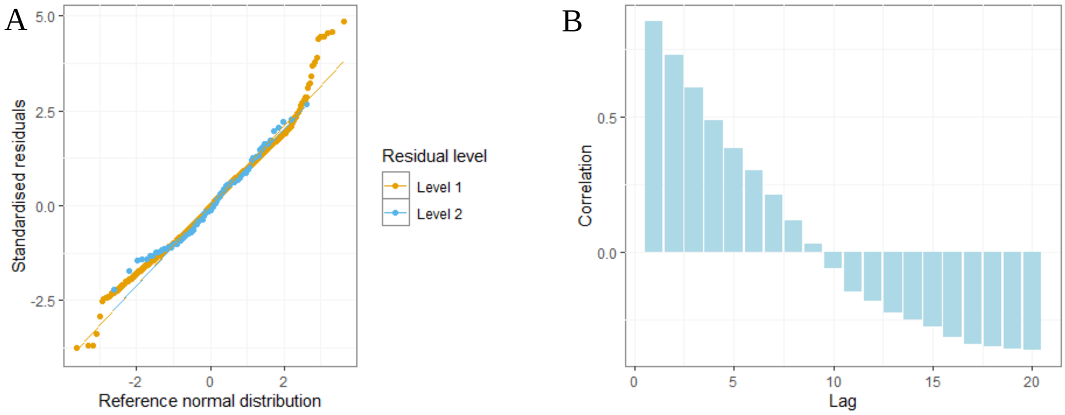
Linear random intercept model with rigour and impartiality of the public administration as dependent variable and year-fixed effects (*N* = 3406, *n* = 109, *T* = 1-55).

	Model 3		Model 4	
Variable	*b* (*SE*) [a]	*p*	*b* (*SE*) [a]	*p*
Intercept	-0,46 (0,16)	0,15	-2,28 (2,49)	0,36
Personalism	-0,71 (0,15)	<0,01	-0,67 (0,15)	<0,01
Political Violence			-0,17 (0,06)	0,01
$\log_{10}$(Population)			-0,02 (0,33)	0,95
ln(GDP per cap)			0,27 (0,13)	0,06
GDP Growth			-0,12 (0,16)	0,45
Internal conflict (1 = conflict)			-0,13 (0,11)	0,27
International conflict (1= conflict)			0,02 (0,08)	0,78
Regime type				
Personalist (ref)				
Military			0,06 (0,15)	0,69
Single-party			0,14 (0,22)	0,54
Monarchy			0,77 (0,48)	0,15
Random effect				
Between-country	0,89		0,76	
Within-country	0,31		0,27	
R_1^2 [c]	0,03		0,16	
ICC	0,74		0,74	
LR χ^2 Year-fixed [b]	70,79	0,06	109,65	<0,01
LR χ^2 [d]	199,28	<0,01	429,39	<0,01

Note: [a] Country-clustered standard errors. [b] Compares model with year effects to an equivalent model without them. [c] Proportional decrease in level 1 variance compared to Model 1. [d] Compares each model to the previous model.

Figure 9

QQ-plot and autocorrelation plot for rigour and impartiality of the public administration.



Note: The standardised residuals are standardised with country-level standard deviations.

Conclusion & Discussion:

Based on the results of the random intercept regressions, higher levels of personalism in a country tend to go along with somewhat lower freedom of expression, freedom of assembly, protection of life and physical integrity and freedom of movement. It might be that this is partly because higher levels of personalism in a country are associated to lesser competence of coercive institutions, but my evidence for this remains rather inconclusive. I need to reject my hypothesis of a curvilinear association between personalism and freedom of expression, but the extent to which the theory underlying the other hypotheses is corroborated is also limited. This means that my results are in line with the findings of Frantz et al. (2019) and broaden them somewhat by looking at civil liberties instead of repression.

I am not able to come to all too firm conclusions with regards to the role of the competence of coercive institutions in explaining reduced civil liberties in more personalised regimes. However, that my results at least suggest an effect does give me access to evidence for a mechanism which Frantz et al. (2019) do not give. This deepens their results, and in order to understand why personalism is associated to worse conditions for the inhabitants of a country it is important to continue looking into evidence for possible mechanisms. The result that my operationalisation of the competence of coercive institutions, the rigour and impartiality of the public administration, is related to greater civil liberties is itself in line with the predictions of, among others, Greitens (2016).

A conceptually important issue is that the operationalisation of personalism I use (Geddes et al., 2018; Wright, 2021) leans heavily on concrete actions connected with personalisation, such as the creation of a support party or purges of the military. Frantz et al. (2019) suggest that this operationalisation is at least not intrinsically connected to repression by including indicators of repression as indicators of personalism. Since I operationalise the competence of coercive institutions with a measure of rigour and impartiality of the public administration, which measures the respect of public officials for the law and unbiased administration of it (Coppedge et al., 2021a, pp. 175-176), this measure should also be distinct from personalism. The personalism measure is based on the relation between the dictator and his government, while the rigour and impartiality of the public administration operationalises the relation between the government and the general populace.

However, going beyond definitions and towards my theoretical argument, if a dictator undermines the competence of coercive institutions to allow further personalisation this suggests that a dictator that does not undermine the competence of coercive institutions might not be able to personalise rule as much. This is not a necessary relation, but it muddles the direction of causality and means that personalism and the competence of coercive institutions might be hard to disentangle empirically. This might explain why I did not only find (weak) evidence for the mechanism where I expected to do so, but also where I did not expect it.

A conundrum in my research setup is that I test for the effect of personalism on four civil liberties that are theoretically and empirically distinct, but are sufficiently related that the effect of personalism on one might bias the estimated effect of personalism on another. At the same time, adding the other civil liberties to the model of any one of them would create the rather questionable theoretical setup of looking at the effect of personalism on a civil liberty in those situations where the other civil liberties remain unaffected by it. Since the effects of personalism are fairly similar across the civil liberties, while the relations between civil liberties are strong but not extremely so, the extreme case of some civil liberties only being related to personalism through their relation to other civil liberties seems unlikely. However, this does not solve the underlying theoretical problem. A strategy to address this problem in future is to use multivariate multilevel models (Snijders & Bosker, 2012, pp. 282-288), which explicitly incorporate the possibility of multiple related outcomes.

Another problem in my model specification is that I have not modelled the dependence of values within a country from one year to the next. The level 1 residuals still showed strong dependence over time. This means that the amount of information my data contains is far lower that my model assumes, such that my statistical tests give statistically significant results more often than is justified. Of course, this can very plausibly mean that my conclusion that personalism affects civil liberties through the competence of coercive institutions is not statistically justifiable.

Beyond implausibly assuming within-country independence over time, another statistical limitation is the fact that I have had to assume that observations in different countries are independent from each other. While such independence is not problematic for individual humans randomly sampled from a large population, for a dataset containing all autocracies for the period under study this seems rather less plausible. This is exacerbated by the fact that countries might not only be similar because they are affected by a shared environment (e.g., interference from democratic states), but autocratic countries also have an interest to interfere in the domestic affairs of each other. I have assumed that personalism and civil liberties in the various members of the Warsaw pact developed independently, for example, while the Berlin Uprising of 1953, the Prague Spring of 1968, and the revolutions of 1989 make this assumption seem rather implausible.

An interesting extension to test the mechanisms I have presented here is to operationalise the threat of an elite coup or the extent to which the dictator is able to monitor their elite. I have only been able to study the proposed effect of elite coup threat in a very indirect way that requires a lot of my assumptions to be correct. I could not empirically distinguish the ability of a dictator to monitor their elite at all. Being able to look into the role of these mechanisms in more depth would certainly enrich a study of the consequences of personalism.

That the effect that I find is rather small and very uncertain makes clear practical implications hard to draw. Assuming that the consequences for human well-being nonetheless recommend some type of intervention I can make a few suggestions. If personalism in autocratic countries is a threat to civil liberties, it seems wise for liberal democracies to promote a more dispersed power structure in autocratic countries instead of treating all forms of autocratic government as equally undesirable. Given that the competence of coercive institutions might be one mechanism through which personalism leads to less respect for civil liberties, a method for achieving the first goal that can reduce suffering in autocracies in and of itself might be to help and pressure autocracies to establish a professional government structure that is not dominated by clientelist practices and can exert soft power. However, for both these approaches the threat of perverse consequences is considerable. Keeping power from getting overly concentrated can itself disadvantage the population of a countries if it leads to destabilisation of a regime and possibly civil war. And a more professional government apparatus is only beneficial to civil liberties if it uses its capacities for more restrained methods of control, instead of more rigorous ones. And insofar as professionalising government conflicts with personalisation, getting a dictator to cooperate will be very difficult.

To summarise, how much power lies in the hands of a dictator seems to have some deleterious consequences for freedoms of expression, assembly, movement and the protection of life and physical integrity. This seems to happen partly because dictators supress the competence of their coercive institutions, but which other mechanisms are at play remains to be seen.

References:

Acemoglu, D., Egorov, G. & Sonin, K. (2008). Coalition formation in non-democracies. *Review of Economic Studies*, *75*(4), 987-1009. https://doi.org/10.1111/j.1467-937X.2008.00503.x

Acemoglu, D., Egorov, G., & Sonin, K. (2009). Political economy under weak institutions: Do juntas lead to personal rule? *American Economic Review*, *99*(2), 298-303. https://doi.org/10.1257/aer.99.2.298

Acemoglu, D., Egorov, G., & Sonin, K. (2012). Dynamics and stability of constitutions, coalitions, and clubs. *American Economic Review*, *102*(4), 1446-1476. https://doi.org//10.1257/aer.102.4.1446

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, *67*(1), 1–48. https://doi.org/10.18637/jss.v067.i01

Baturo, A., & Elkink, J. A. (2016). Dynamics of regime personalisation and patron-client networks in Russia, 1999–2014. *Post-Soviet Affairs*, *32*(1), 75-98. https://doi.org/10.1080/1060586x.2015.1032532

Boleslavsky, R., Shadmehr, M., & Sonin, K. (2021). Media freedom in the shadow of a coup. *Journal of the European Economic Association*, *19*(3), 1782-1815. https://doi.org/10.1093/jeea/jvaa040

Bolt, J., Inklaar, R., de Jong, H., & van Zanden, J. L. (2018). *Rebasing 'Maddison': New income comparisons and the shape of long-run economic development* (GGDC Research Memorandum 174). Groningen Growth and Development Centre. https://www.rug.nl/ggdc/html_publications/memorandum/gd174.pdf

Chin, J. (2020, May 8). *Five myths about coups: No, they're not found only in poorer nations.* The Washington Post. https://www.washingtonpost.com/outlook/five-myths/five-myths-about-coups/2020/05/07/9c64ee04-8f1d-11ea-B9c0-73b93422d691_story.html

Coppedge, M., Gerring, J., Knutsen, C. H., Lindberg, S. I., Teorell, J., Alizada, N., Altman, D., Bernhard, M., Cornell, A., Fish, M. S., Gastaldi, L., Gjerløw, H., Glynn, A., Hicken, A., Hindle, G., Ilchenko, N., Krusell, J., Lührmann, A., Maerz, S. F., … Ziblatt, D. (2021a). *V-Dem Codebook* [Code Book] (version 11.1). Varieties of Democracy (V-Dem) Project.

Coppedge, M., Gerring, J., Knutsen, C. H., Lindberg, S. I., Teorell, J., Alizada, N., Altman, D., Bernhard, M., Cornell, A., Fish, M. S., Gastaldi, L., Gjerløw, H., Glynn, A., Hicken, A., Hindle, G., Ilchenko, N., Krusell, J., Lührmann, A., Maerz, S. F., … Ziblatt, D. (2021b). *V-Dem Country–Year Dataset* [Data set] (version 11.1). Varieties of Democracy (V-Dem) Project. https://doi.org/10.23696/vdemds21

Coppedge, M., Gerring, J., Knutsen, C. H., Lindberg, S. I., Teorell, J., Alizada, N., Altman, D., Bernhard, M., Cornell, A., Fish, M. S., Gastaldi, L., Gjerløw, H., Glynn, A., Hicken, A., Hindle, G., Ilchenko, N., Krusell, J., Lührmann, A., Maerz, S. F., … Ziblatt, D. (2021c). *V-Dem Methodology* (version 11). Varieties of Democracy (V-Dem) Project. https://doi.org/10.23696/vdemds21

Davenport, C. (2007a). State repression and political order. *Annual Review of Political Science*, *10*, 1-23. https://doi.org/10.1146/annurev.polisci.10.101405.143216

Davenport, C. (2007b). State repression and the tyrannical peace. *Journal of Peace Research*, *44*(4), 485-504. https://doi.org/10.1177/0022343307078940

Davenport, C., RezaeeDaryakenari, B., & Wood, R. M. (2021). Tenure through tyranny? Repression, dissent, and leader removal in Africa and Latin America, 1990-2006.

Journal of Global Security Studies, *7*(1), Article ogab023. https://doi.org/10.1093/jogss/ogab023

Egorov, G., & Sonin, K. (2020). *The political economics of non-democracy* (Working Paper 27949). National Bureau of Economic Research. http://www.nber.org/papers/w27949

Frantz, E., Kendall-Taylor, A., Wright, J., & Xu, X. (2019). Personalisation of power and repression in dictatorships. *The Journal of Politics, 82*(1), 372-377. https://doi.org/10.1086/706049

Geddes, B., Wright, J., & Frantz, E. (2014). Autocratic breakdown and regime transitions. *Perspectives on Politics*, *12*(2), 313-331. https://doi.org/10.1017/S1537592714000851

Geddes, B., Wright, J., & Franz, E. (2018). *How dictatorships work: Power, personalisation and collapse*. Cambridge University Press. https://doi.org/10.1017/9781316336182.004

Gehlbach, S., Sonin, K., & Svolik, M. W. (2016). Formal models of nondemocratic politics. *Annual Review of Political Science*, *19*, 565-584. https://doi.org/10.1146/annurev-polisci-042114-014927

Greitens, S. C. (2016). *Dictators and their secret police: Coercive institutions and state violence.* Cambridge University Press. https://doi.org/10.1017/CBO9781316489031.002

Hill, D. W., & Jones, Z. M. (2014). An empirical evaluation of explanations for state repression. *The American Political Science Review*, *108*(3), 661-687. https://doi.org/10.1017/S000305541

Hollyer, J. R., Rosendorff, B. P., & Vreeland, J. R. (2019). Why do autocrats disclose? Economic transparency and inter-elite politics in the shadow of mass unrest. *Journal of Conflict Resolution*, *63*(6), 1488-1516. https://doi.org/10.1177/0022002718792602

Jandoc, K., & Juarez, R. (2019). An experimental study of self-enforcing coalitions. *Games*, *10*(3), Article 31. https://doi.org/10.3390/g10030031

Levitsky, S., & Way, L. A. (2016). Durable authoritarianism. I. O. Fioretos, T. G. Faletti, & A. Sheingate (Eds.), *The Oxford handbook of historical institutionalism.* Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199662814.013.12

Libman, A. (2020). Temporality of authoritarian regimes. In K. H. Goetz (Ed.), *The Oxford handbook of time and politics*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780190862084.013.29

Newson, A., & Trebbi, F. (2018). Authoritarian elites. *Canadian Journal of Economics, 51*(4), 1088-1117. https://doi.org/10.1111/caje.12362

MacKinnon, J. G., & White, H. (1985). Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. *Journal of Econometrics*, 29(3), 305-325. https://doi.org/10.1016/0304-4076(85)90158-7

McDermott, K. (2014). Stalin and Stalinism. In S. A. Smith (Ed.), *The Oxford handbook of the history of communism* (pp. 72-87). Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199602056.013.007

Møller, J., & Skaaning, S.-E. (2013a). Autocracies, democracies and the violation of civil liberties. *Democratization*, *20*(1), 82-106. https://doi.org/10.1080/13510347.2013.738863

Møller, J., & Skaaning, S.-E. (2013b). Respect for civil liberties during the third wave of democratization: Presenting a new dataset. *Social Indicators Research*, *117*(3), 1069-1087. https://doi.org/10.1007/s11205-013-0391-y

Møller, J., & Skaaning, S.-E. (2013c). Single-party autocracies, ideology, and repression. *Statsvetenskaplig Tidskrift*, *115*(4), 345-363. https://journals.lub.lu.se/st/article/view/9742

Newson, A., & Trebbi, F. (2018). Authoritarian elites. *Canadian Journal of Economics*, *51*(4), 1088-1117. https://doi.org/10.1111/caje.12362

Pemstein, D., Marquardt, K. L., Tzelgov, E., Wang, Y.-t., Medzihorsky, J., Krusell, J., Miri, F., & van Römer, J. (2021). *The V-Dem measurement model: Latent variable analysis for cross-national and cross-temporal expert-coded data* (Working Paper No. 21, 6th ed.). Varieties of Democracy (V-Dem) Project. https://web.archive.org/web/20211021062126/https://www.v-dem.net/media/filer_public/d4/d2/d4d27532-96a7-4f8c-8028-4c356fb1b2ca/wp_21_6th_edition.pdf

Pustejovsky, J. E., & Tipton, E. (2016). Small-sample methods for cluster-robust variance estimation and hypothesis testing in fixed effects models. *Journal of Business & Economic Statistics*, *36*(4), 672-683. https://doi.org/10.1080/07350015.2016.1247004

R Core Team. (2021). *R: A language and environment for statistical computing.* (Version 4.1.1) [Computer software]. R Foundation for Statistical Computing https://www.R-project.org/

Sinkkonen, E. (2021). Dynamic dictators: improving the research agenda on autocratization and authoritarian resilience. *Democratization*, *28*(6), 1172-1190. https://doi.org/10.1080/13510347.2021.1903881

Snijders, T. A. B., & Bosker, R. J. (2012). *Multilevel analysis: An introduction to basic and advanced multilevel modeling*. Sage.

Svolik, M. W. (2012). *The politics of authoritarian rule.* Cambridge University Press. https://doi.org/10.1017/CBO9781139176040.003

Turner, T. & Young, C. (1985). *The rise and decline of the Zairian State*. University of Wisconsin Press.

Universal Declaration of Human Rights, December 10, 1948, https://www.un.org/sites/un2.un.org/files/udhr.pdf

Wright, J. (2021). The latent characteristics that structure autocratic rule. *Political Science Research and Methods*, *9*(1), 1-19. https://doi.org/10.1017/psrm.2019.50

Zu, Z. (2020). *Logic of selective repression: How do elite purges affect state violence in authoritarian regimes* [Master’s Thesis, Duke University]. DukeSpace. https://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/20790/Zu_duke_0066N_15620.pdf?sequence=1&isAllowed=y

Appendix: R script

This appendix contains the R script used to run the analyses for this thesis. The script starts at the next page. The datasets used by the script are "GWF+personalism-scores.csv" which can be retrieved from https://tinyurl.com/277vebkf, "GWF Autocratic Regimes.xlsx", which can be retrieved from https://tinyurl.com/2cpwmwvt, and "GWFtscs.txt", which can be retrieved from https://tinyurl.com/23krvz7a.

The shortened URLs above are more useful for manual entry. In case they do not work or the full URL is preferred, these are:

V-Dem: www.v-dem.net/media/datasets/Country_Year_V-Dem_Full_others_R_v11.1.zip

Geddes et al. (2018): www.sites.psu.edu/dictators/wp-content/uploads/sites/12570/2016/05/GWF-Autocratic-Regimes-1.2.zip

Wright (2021): www.sites.psu.edu/wright/files/2019/11/GWF-time-vary-personalism.zip

```
# Author:   Siebren Kooistra
# Date:     05-06-2022
# Goal:     Construct a dataset on personalism and civil liberties using
#           Geddes et al. (2014, 2018) and V-Dem (Coppedge et al., 2021) and
#           carry out analyses on the associations between personalism and civil
#           liberties

# Preparations: clean workspace and mount libraries.
rm(list = ls())
graphics.off()
gc()
library(DescTools)
library(readxl)
library(tidyverse)
library(GGally)
library(lmtest)
library(ggrepel)
library(mice)
library(naniar)
library(lme4)
library(arsenal)
library(plm)
library(clubSandwich)

#  Create visual elements for later use
blue_light <- theme(panel.background = element_rect(fill = "white",
                                                    colour = "grey50"),
                    panel.grid.major = element_line(colour = "whitesmoke"),
                    panel.grid.minor = element_line(colour = "whitesmoke"),
                    legend.key = element_rect(fill = "white",
                                              colour = "grey50"))
lines_palette <- c("darkgreen", "blue", "lightblue", "lightblue", "red")

################################################################################
###################### DATASET ASSEMBLY AND DESCRIPTION  #######################
################################################################################

######################  Dataset assembly  ######################################

# First dataset: GWF with personalism

#   Load GWF autocracies dataset with personalism data and specify
#   which variables to use
GWF_personalism <- read.csv("GWF+personalism-scores.csv")
GWF_personalism_variables_of_interest <- c("cowcode", "gwf_country", "year",
                                           "dataID", "latent_personalism",
                                           "paramil_pers", "sectyapp_pers",
                                           "officepers", "partyrbrstmp",
                                           "militparty_newparty", "milnotrial",
                                           "milmerit_pers", "partyexcom_pers")

#     Define a new variable dataID to match V-Dem country-years to GWF
#     country-years. In this case, V-Dem takes the separation of Yemen into two
#     countries from 1918 to 1989 into account by giving the unified Yemen
#     existing until 1918 and from 1990 a COW code of 679. Since this is more
#     informative than the constant GWF code of 678, the GWF code is overwritten
#     as 679 from 1990 to 2010.
GWF_personalism$cowcode[GWF_personalism$cowcode == 678 &
                          GWF_personalism$year %in% 1990:2010] <- 679
```

```
# Second dataset: GWF regime types

#   Load the GWF regime type dataset and specify which variables to use
GWF_regime_type <- read.delim("GWFtscs.txt")
GWF_regime_type_variables_of_interest <- c("dataID", "gwf_party",
                                           "gwf_personal", "gwf_military",
                                           "gwf_monarch")

#   Apply the same correction for South Yemen as used in the personalism
#   dataset to the regime type dataset
GWF_regime_type$cowcode[GWF_regime_type$cowcode == 678 &
                          GWF_regime_type$year %in% 1990:2010] <- 679

# Third dataset:  V-Dem

#   Load V-Dem datafile and specify which variables to use
VDem <- readRDS("V-Dem-CY-Full+Others-v11.1.rds")
VDem_variables_of_interest <- c("COWcode", "dataID", "v2x_clphy", "v2cltort",
                                "v2clkill", "v2caassemb", "v2clfmove",
                                "v2cldmovem", "v2cldmovew", "v2x_freexp",
                                "v2mecenefm", "v2meharjrn", "v2meslfcen",
                                "v2xcl_disc", "v2clacfree", "e_mipopula",
                                "e_miinteco", "e_miinterc", "e_migdppcln",
                                "e_migdpgro", "v2caviol", "v2clrspct",
                                "v2cldiscm", "v2cldiscw", "lag_e_miinteco",
                                "lag_e_miinterc", "lag_e_migdppcln",
                                "lag_e_migdpgro", "lag_v2caviol",
                                "lag_e_mipopula")

#   Create variables to match the three datasets
GWF_regime_type$dataID <- paste(GWF_regime_type$cowcode, GWF_regime_type$year)
GWF_personalism$dataID <- paste(GWF_personalism$cowcode, GWF_personalism$year)
VDem$dataID <- paste(VDem$COWcode, VDem$year)

#   Lag variables
VDempdata <- pdata.frame(VDem, index = c("COWcode", "year"))
VDem$lag_e_miinteco <- plm::lag(VDem$e_miinteco)
VDem$lag_e_miinterc <- plm::lag(VDem$e_miinterc)
VDem$lag_e_migdppcln <- plm::lag(VDem$e_migdppcln)
VDem$lag_e_migdpgro <- plm::lag(VDem$e_migdpgro)
VDem$lag_v2caviol <- plm::lag(VDem$v2caviol)
VDem$lag_e_mipopula <- plm::lag(VDem$e_mipopula)

# Combine the GWF data and the V-Dem data

#   Subset the part of the V-Dem dataset that is relevant for the GWF data
VDem_Repression <- VDem[VDem$dataID %in% GWF_personalism$dataID,
                        c("historical_date",
                          VDem_variables_of_interest)]

#   Combine the datasets
autocracy_data <- cbind(GWF_personalism[order(GWF_personalism$dataID),
                                        GWF_personalism_variables_of_interest],
                        GWF_regime_type[order(GWF_regime_type$dataID),
                                        GWF_regime_type_variables_of_interest],
                        VDem_Repression[order(VDem_Repression$dataID),
                                        VDem_variables_of_interest])
```

```
#   Check for sorting errors.
TEST <- all(autocracy_data$cowcode == autocracy_data$COWcode)

# Variable operations
#   Create a log-10 population variable
autocracy_data$log10pop <- log10(autocracy_data$e_mipopula * 1000)
autocracy_data$lag_log10pop <- log10(autocracy_data$lag_e_mipopula * 1000)

#   Compute the scale variable for freedom of movement
autocracy_data$freedom_movement <-
  rowMeans(autocracy_data[, c("v2clfmove", "v2cldmovem", "v2cldmovew")])

#   Multiply freedom of expression and protection of life and physical integrity
#   scales by 100 to avoid errors due to floating point arithmetic
autocracy_data$free_expr_x100 <- autocracy_data$v2x_freexp * 100
autocracy_data$life_phys_x100 <- autocracy_data$v2x_clphy * 100

# Tidy up dataset
autocracy_data <- autocracy_data[, colnames(autocracy_data) %nin% c("dataID",
                                                                    "COWcode")]
# Subset complete cases
datacomplete <- autocracy_data[complete.cases(autocracy_data),]

# Write dataset into csv file
write.csv(autocracy_data,
          "autocracies_personalism_civilrights_data_S_Kooistra.csv")
write.csv(datacomplete,
          "complete_cases_autocracies_personalism_civilrights_S_Kooistra.csv")

# Clean up to save memory.
rm(GWF_personalism, GWF_personalism_variables_of_interest,
   GWF_regime_type, GWF_regime_type_variables_of_interest,
   VDem, VDem_Repression, VDem_variables_of_interest,
   datacomplete, autocracy_data) # Also remove dataframes to re-load and
                                 # remove pdata.frame properties
gc()

# Re-load dataframes
autocracy_data <-
  read.csv("autocracies_personalism_civilrights_data_S_Kooistra.csv")
datacomplete <-
  read.csv("complete_cases_autocracies_personalism_civilrights_S_Kooistra.csv")

######################   Personalism descriptives   ###########################

#   Item description and scale evaluation

#     Military promotions

#       Table and chi-square test
milmerit_pers_table <- table(autocracy_data$milmerit_pers, autocracy_data$year)
milmerit_pers_table
summary(milmerit_pers_table)

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(milmerit_pers))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Military promotions") +
  scale_fill_manual(values = c("tomato", "lightblue", "darkgreen"),
```

```
  labels = c("No forced retirement, \nno loyalty-based promotion, \nor no
military",
             "Loyalty-based promotion",
             "In-group promotion \nor forced retirement"))

#     Recode military promotion variable to dummy
autocracy_data$milmerit_pers_twocat <- recode(autocracy_data$milmerit_pers,
                                              `0` = 0, `1` = 0, `2` = 1)

#     Investigate distribution of dummy
milmerit_pers_twocat_table <- table(autocracy_data$milmerit_pers_twocat,
                                    autocracy_data$year)
milmerit_pers_twocat_table
summary(milmerit_pers_twocat_table)

#       Stacked barplot per year for dummy
ggplot(autocracy_data, aes(x = year, fill = as.factor(milmerit_pers_twocat))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Military promotions") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("No forced retirement, \nno loyalty-based
promotion, \nor no military, \nor loyalty-based promotion",
                    "In-group promotion \nor forced retirement"))

#     Military purges

#       Table and chi-square test
milnotrial_table <- table(autocracy_data$milnotrial, autocracy_data$year)
milnotrial_table
summary(milnotrial_table)
chisq.test(autocracy_data$milnotrial, autocracy_data$year)$stdres

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(milnotrial))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Military purges") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("No imprisonment or killing, \nno military, \nor
foreign military",
                               "Officer imprisonment and/or killing"))

#     Support party

#       Table and chi-square test
militparty_newparty_table <- table(autocracy_data$militparty_newparty,
                                   autocracy_data$year)
militparty_newparty_table
summary(militparty_newparty_table)
chisq.test(autocracy_data$militparty_newparty, autocracy_data$year)$stdres
chisq.test(autocracy_data$militparty_newparty, autocracy_data$year)$expected
fisher.test(autocracy_data$militparty_newparty, autocracy_data$year,
            simulate.p.value = TRUE, B = 10000)

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(militparty_newparty))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Party creation") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("No support party created",
```

```
                              "Support party created"))

#     Executive committee appointments

#       Table and chi-square test
partyexcom_pers_table <- table(autocracy_data$partyexcom_pers,
                                  autocracy_data$year)
partyexcom_pers_table
summary(partyexcom_pers_table)
chisq.test(autocracy_data$partyexcom_pers, autocracy_data$year)$stdres

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(partyexcom_pers))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion",
       fill = "Party executive committee control") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("Regime leader does not choose \nparty executive 
committee",
                               "Regime leader chooses \nparty executive 
committee"))

#     Executive committee rubberstamp

#       Check coding
cor(autocracy_data$partyrbrstmp,
    autocracy_data[, c("latent_personalism", "paramil_pers",
                       "sectyapp_pers", "officepers", "partyrbrstmp",
                       "militparty_newparty", "milnotrial",
                       "milmerit_pers", "partyexcom_pers")])

#       Table and chi-square test
partyrbrstmp_table <- table(autocracy_data$partyrbrstmp,
                               autocracy_data$year)
partyrbrstmp_table
summary(partyrbrstmp_table)

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(partyrbrstmp))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion",
       fill = "Party executive committee independence") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("The party executive committee \nhas some policy 
independence \nfrom the regime leader",
                               "The party executive committee \nis a 'rubber 
stamp' \nor does not exist"))

#     Discretion over high office appointments

#       Table and chi-square test
officepers_table <- table(autocracy_data$officepers,
                            autocracy_data$year)
officepers_table
summary(officepers_table)
chisq.test(autocracy_data$officepers, autocracy_data$year)$stdres

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(officepers))) +
```

```
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion",
       fill = "Regime leader discretion over high office appointments") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("Regime leader does not have discretion \nover
appointments to high office",
                               "Regime leader has discretion over \nhigh office
appointments or appoints \nrelatives to these positions"))

#     Personalised control over the security apparatus

#       Table and chi-square test
sectyapp_pers_table <- table(autocracy_data$sectyapp_pers,
                          autocracy_data$year)
sectyapp_pers_table
summary(sectyapp_pers_table)
chisq.test(autocracy_data$sectyapp_pers, autocracy_data$year)$stdres

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(sectyapp_pers))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion",
       fill = "Regime leader's personal control over the security apparatus") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("Security apparatus is not controlled \npersonally
by the regime leader",
                               "Security apparatus is controlled \npersonally by
the regime leader"))

#     Loyal paramilitary forces

#       Table and chi-square test
paramil_pers_table <- table(autocracy_data$paramil_pers,
                             autocracy_data$year)
paramil_pers_table
summary(paramil_pers_table)
chisq.test(autocracy_data$paramil_pers, autocracy_data$year)$stdres

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(paramil_pers))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion",
       fill = "Creation of paramilitary forces loyal to the regime leader") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("Regime leader does not create \nparamilitary
forces, \na president's guard, \nor new security forces \napparently loyal to
himself",
                               "Regime leader creates \nparamilitary forces, \na
president's guard, \nor new security forces \napparently loyal to himself"))

#     Reliability analysis

#       Calculate Cronbach's Alpha by year
personalism_cronbachalphas <-
  by(data = autocracy_data[, c("milmerit_pers_twocat", "milnotrial",
                               "militparty_newparty", "partyexcom_pers",
                               "partyrbrstmp", "officepers",
                               "sectyapp_pers", "paramil_pers")],
     INDICES = autocracy_data$year,
```

```
      FUN = CronbachAlpha, cond = TRUE)

#       Create table for calculated Cronbach's Alpha
table_personalism_cronbachalphas <-
  tibble(year = sort(unique(datacomplete$year)),
         "All items" = NA,
         "Military promotion strategies " = NA,
         "Military purges" = NA,
         "Support party creation" = NA,
         "Party executive committee control" = NA,
         "Party executive committee rubberstamp" = NA,
         "Regime leader discretion over high office appointments" = NA,
         "Personalised control over security apparatus" = NA,
         "Loyal paramilitary forces" = NA
         )

#       Fill table
for (y in sort(unique(autocracy_data$year))) {
  table_personalism_cronbachalphas[table_personalism_cronbachalphas$year == y,
                                   "All items"] <-
    personalism_cronbachalphas[[as.character(y)]][["unconditional"]]
  table_personalism_cronbachalphas[table_personalism_cronbachalphas$year == y,
                                   "Military promotion strategies"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][1, "Cronbach Alpha"]
  table_personalism_cronbachalphas[table_personalism_cronbachalphas$year == y,
                                   "Military purges"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][2, "Cronbach Alpha"]
  table_personalism_cronbachalphas[table_personalism_cronbachalphas$year == y,
                                   "Support party creation"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][3, "Cronbach Alpha"]
  table_personalism_cronbachalphas[table_personalism_cronbachalphas$year == y,
                                   "Party executive committee control"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][4, "Cronbach Alpha"]
  table_personalism_cronbachalphas[table_personalism_cronbachalphas$year == y,
                                   "Party executive committee rubberstamp"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][5, "Cronbach Alpha"]
  table_personalism_cronbachalphas[
    table_personalism_cronbachalphas$year == y,
    "Regime leader discretion over high office appointments"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][6, "Cronbach Alpha"]
  table_personalism_cronbachalphas[
    table_personalism_cronbachalphas$year == y,
    "Personalised control over security apparatus"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][7, "Cronbach Alpha"]
  table_personalism_cronbachalphas[table_personalism_cronbachalphas$year == y,
                                   "Loyal paramilitary forces"] <-
    personalism_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
                                                    ]][8, "Cronbach Alpha"]
}

#       Pivot table for use in plotting
longtab_personalism_cronbachalphas <-
```

```
  pivot_longer(table_personalism_cronbachalphas, 2:10, names_to = "Type",
               values_to = "Value")

#       Plot Cronbach's Alpha by year
ggplot(longtab_personalism_cronbachalphas, aes(x = year, y = Value,
                                               colour = Type)) +
  geom_line() + blue_light

#       Calculate range of yearly Cronbach's Alphas
range(table_personalism_cronbachalphas$"All items" )

#   Scale descriptives

#     Entire dataset

#     Draw time series for Argentina, China, Congo/Zaire, Persia/Iran, Portugal
#     and the Soviet Union and Russia
ggplot(autocracy_data[autocracy_data$cowcode %in%
                        c(160, 710, 490, 630, 235, 365),],
       aes(x = year, y = latent_personalism, colour = gwf_country)) +
  geom_step() + facet_grid(rows = vars(cowcode)) + blue_light +
  labs(x = "Year", y = "Latent personalism")  +
  scale_color_manual(name='Country', labels = vars(gwf_country),
                     values = palette("Okabe-Ito"))

#       Heatmap of personalism distribution per year
ggplot(data = autocracy_data, mapping = aes(y = latent_personalism,
                                            x = year)) +
  geom_bin2d(binwidth = c(1, 0.1)) + blue_light +
  labs(y = "Latent personalism", x = "Year")

#       Calculation of country means
latent_personalism_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$latent_personalism,
                       autocracy_data$cowcode,
                       mean)
  )
#       Histogram and QQ-plot of country means
ggplot(data = latent_personalism_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue") +
  labs(x = "Latent personalism", y = "Count") + blue_light
ggplot(data = latent_personalism_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Latent personalism") +
  blue_light

#       Compute summary statistics per year
latent_personalism_full_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$latent_personalism,
                       autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$latent_personalism,
                     autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$latent_personalism,
                         autocracy_data$year, median),
         Q3 = tapply(autocracy_data$latent_personalism,
```

```
                     autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$latent_personalism,
                     autocracy_data$year, sd),
         Skew = tapply(autocracy_data$latent_personalism,
                     autocracy_data$year, FUN = Skew),
         Kurtosis = tapply(autocracy_data$latent_personalism,
                       autocracy_data$year, FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
latent_personalism_full_summary_longtable <-
  pivot_longer(latent_personalism_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = latent_personalism_full_summary_longtable[
    latent_personalism_full_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                               "Median", "Q3",
                                                               "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
    geom_line() + blue_light + labs(x = "Year", y = "Latent personalism"))
)

#       Plot skew and kurtosis over time
ggplot(data = latent_personalism_full_summary_longtable[
  latent_personalism_full_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() +
  labs(x = "Year", y = "Latent personalism") +
  blue_light

#     Complete cases

#       Heatmap of personalism distribution per year
ggplot(data = datacomplete, mapping = aes(y = latent_personalism,
                                          x = year)) +
  geom_bin2d(binwidth = c(1, 0.1)) + blue_light +
  labs(y = "Latent personalism", x = "Year")

#     Calculation of country means
latent_personalism_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$latent_personalism, datacomplete$cowcode,
                       mean)
)

#       Histogram and QQ-plot of country means
ggplot(data = latent_personalism_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue") +
  labs(x = "Latent personalism", y = "Count") + blue_light
ggplot(data = latent_personalism_complete_countrymeans,
       aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Latent personalism") +
```

```r
  blue_light

#       Compute summary statistics per year
latent_personalism_complete_summary <-
  tibble(year =  unique(datacomplete$year)[order(unique(
    datacomplete$year))],
         Mean = tapply(datacomplete$latent_personalism, datacomplete$year,
                       mean),
         Q1 = tapply(datacomplete$latent_personalism, datacomplete$year,
                     quantile, prob = 0.25),
         Median = tapply(datacomplete$latent_personalism,
                         datacomplete$year, median),
         Q3 = tapply(datacomplete$latent_personalism,
                     datacomplete$year,
                     quantile, prob = 0.75),
         SD = tapply(datacomplete$latent_personalism,
                     datacomplete$year, sd),
         Skew = tapply(datacomplete$latent_personalism,
                       datacomplete$year, FUN = Skew),
         Kurtosis = tapply(datacomplete$latent_personalism,
                       datacomplete$year, FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
latent_personalism_complete_summary_longtable <-
  pivot_longer(latent_personalism_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = latent_personalism_complete_summary_longtable[
    latent_personalism_complete_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                                   "Median", "Q3",
                                                                   "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Latent personalism"))
)

#       Plot skew and kurtosis over time
ggplot(data = latent_personalism_complete_summary_longtable[
  latent_personalism_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Latent personalism") + blue_light

######################   Freedom of expression descriptives  ##################

#   Item description and scale evaluation

#     Government censorship effort

#       Establish observed range
range(autocracy_data$v2mecenefm)

#       Heatmap of government censorship effort distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2mecenefm, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Government censorship effort", x = "Year")
```

```
#       Calculation of country means
government_censorship_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2mecenefm, autocracy_data$cowcode, mean)
)
#       Histogram and QQ-plot of country means
ggplot(data = government_censorship_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Government censorship effort", y = "Count") + blue_light
ggplot(data = government_censorship_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Government censorship effort") + blue_light

#       Compute summary statistics per year
government_censorship_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2mecenefm, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2mecenefm, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2mecenefm, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$v2mecenefm, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2mecenefm, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2mecenefm, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2mecenefm, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics for use in plotting
government_censorship_summary_longtable <-
  pivot_longer(government_censorship_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = government_censorship_summary_longtable[
    government_censorship_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                              "Median", "Q3",
                                                              "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() +
      blue_light + labs(x = "Year", y = "Government censorship effort"))
)

#       Plot skew and kurtosis over time
ggplot(data = government_censorship_summary_longtable[
  government_censorship_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Government censorship effort") + blue_light

#     Harassment of journalists

#       Establish observed range
range(autocracy_data$v2meharjrn)
```

```
# 	Heatmap of personalism distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2meharjrn, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Journalist harassment", x = "Year")

# 	Calculation of country means
journalist_harassment_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2meharjrn, autocracy_data$cowcode,
                       mean)
)
# 	Histogram and QQ-plot of country means
ggplot(data = journalist_harassment_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Journalist harassment", y = "Count") + blue_light
ggplot(data = journalist_harassment_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") +
  geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Journalist harassment") +
  blue_light

# 	Compute summary statistics per year
journalist_harassment_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2meharjrn, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2meharjrn, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2meharjrn, autocracy_data$year, median),
         Q3 = tapply(autocracy_data$v2meharjrn, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2meharjrn, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2meharjrn, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2meharjrn, autocracy_data$year,
                           FUN = Kurt) )

# 	 Pivot summary statistics table for use in plotting
journalist_harassment_summary_longtable <-
  pivot_longer(journalist_harassment_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

# 	Plot Mean, median, first quartile, third quartile
# 	and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = journalist_harassment_summary_longtable[
    journalist_harassment_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                             "Median", "Q3",
                                                             "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year",
                                      y = "Journalist harassment"))
)

# 	Plot skew and kurtosis over time
ggplot(data = journalist_harassment_summary_longtable[
  journalist_harassment_summary_longtable$Statistic %in%
```

```
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Journalist harassment") + blue_light

#     Media self-censorship

#       Establish observed range
range(autocracy_data$v2meslfcen)

#       Heatmap of media self-censorship distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2meslfcen,
                                            x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Media self-censorship", x = "Year")

#       Calculation of country means
media_selfcensor_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2meslfcen, autocracy_data$cowcode,
                       mean)
)
#       Histogram and QQ-plot of country means
ggplot(data = media_selfcensor_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Media self-censorship", y = "Count") + blue_light
ggplot(data = media_selfcensor_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Media self-censorship") + blue_light

#       Compute summary statistics per year
media_selfcensor_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2meslfcen, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2meslfcen, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2meslfcen, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$v2meslfcen, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2meslfcen, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2meslfcen, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2meslfcen, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
media_selfcensor_summary_longtable <-
  pivot_longer(media_selfcensor_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = media_selfcensor_summary_longtable[
    media_selfcensor_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                        "Median", "Q3",
```

```
                                                              "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Media self-censorship"))
)

#       Plot skew and kurtosis over time
ggplot(data = media_selfcensor_summary_longtable[
  media_selfcensor_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Media self-censorship") + blue_light

#     Freedom of discussion

#       Freedom of discussion for men

#         Establish observed range
range(autocracy_data$v2cldiscm)

#         Heatmap of media freedom of discussion for men distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2cldiscm, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of discussion for men", x = "Year")

#         Calculation of country means
free_disc_men_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2cldiscm, autocracy_data$cowcode,
                       mean)
)
#         Histogram and QQ-plot of country means
ggplot(data = free_disc_men_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of discussion for men", y = "Count") + blue_light
ggplot(data = free_disc_men_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Freedom of discussion for men") + blue_light

#         Compute summary statistics per year
free_disc_men_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2cldiscm, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2cldiscm, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2cldiscm, autocracy_data$year, median),
         Q3 = tapply(autocracy_data$v2cldiscm, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2cldiscm, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2cldiscm, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2cldiscm, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_disc_men_summary_longtable <-
  pivot_longer(free_disc_men_summary, 2:8, names_to = "Statistic",
               values_to = "Value")
```

```
#         Plot Mean, median, first quartile, third quartile
#         and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_disc_men_summary_longtable[
    free_disc_men_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                        "Median", "Q3",
                                                        "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Freedom of discussion for men"))
)

#         Plot skew and kurtosis over time
ggplot(data = free_disc_men_summary_longtable[
  free_disc_men_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom of discussion for men") + blue_light

#       Freedom of discussion for women

#         Establish observed range
range(autocracy_data$v2cldiscw)

#         Heatmap of freedom of discussion for women distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2cldiscw, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of discussion for women", x = "Year")

#       Calculation of country means
free_disc_women_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2cldiscw,
                       autocracy_data$cowcode,
                       mean)
)
#         Histogram and QQ-plot of country means
ggplot(data = free_disc_women_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of discussion for women", y = "Count") + blue_light
ggplot(data = free_disc_women_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Freedom of discussion for women") + blue_light

#         Compute summary statistics per year
free_disc_women_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2cldiscw, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2cldiscw, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2cldiscw, autocracy_data$year, median),
         Q3 = tapply(autocracy_data$v2cldiscw, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2cldiscw, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2cldiscw, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2cldiscw, autocracy_data$year,
```

```
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_disc_women_summary_longtable <-
  pivot_longer(free_disc_women_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#         Plot Mean, median, first quartile, third quartile
#         and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_disc_women_summary_longtable[
    free_disc_women_summary_longtable$Statistic %in% c("Mean", "Q1", "Median",
                                                       "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year",
                                      y = "Freedom of discussion for women"))
)

#         Plot skew and kurtosis over time
ggplot(data = free_disc_women_summary_longtable[
  free_disc_women_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom of discussion for women") + blue_light

#       Sub-item intercorrelation and freedom of discussion description

#         Calculate intercorrelation by year
free_disc_cors <-
  by(data = autocracy_data[, c("v2cldiscm", "v2cldiscw")],
     INDICES = autocracy_data$year, FUN = cor)

#         Create table for correlations
table_free_disc_cors <- tibble(Year = sort(unique(autocracy_data$year)),
                               "Correlation" = NA)

#         Fill table
for (y in sort(unique(autocracy_data$year))) {
  table_free_disc_cors[table_free_disc_cors$Year == y,
                       "Correlation"] <-
    free_disc_cors[[as.character(y)]][2,1]
}

#         Plot correlations by year
ggplot(table_free_disc_cors, aes(x = Year, y = Correlation)) +
  geom_line(colour = "tomato") + blue_light

#         Establish range of correlations
range(table_free_disc_cors$"Correlation")

#         Establish observed range for scale
range(autocracy_data$v2xcl_disc)

#         Heatmap of freedom of discussion distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2xcl_disc, x = year)) +
  geom_bin2d(binwidth = c(1, 0.1)) + blue_light +
  labs(y = "Freedom of discussion", x = "Year")
```

```
#       Calculation of country means
free_disc_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2xcl_disc, autocracy_data$cowcode,
                       mean)
)
#         Histogram and QQ-plot of country means
ggplot(data = free_disc_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue") +
  labs(x = "Freedom of discussion", y = "Count") + blue_light
ggplot(data = free_disc_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Freedom of discussion") + blue_light

#       Compute scale summary statistics per year
free_disc_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2xcl_disc, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2xcl_disc, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2xcl_disc, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$v2xcl_disc, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2xcl_disc, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2xcl_disc, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2xcl_disc, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_disc_summary_longtable <-
  pivot_longer(free_disc_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot mean, median, first quartile, third quartile
#       and standard deviation of scale over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_disc_summary_longtable[
    free_disc_summary_longtable$Statistic %in% c("Mean", "Q1", "Median", "Q3",
                                                 "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year",
                                      y = "Freedom of discussion"))
)

#       Plot skew and kurtosis of scale over time
ggplot(data = free_disc_summary_longtable[
  free_disc_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom of discussion") + blue_light


#     Freedom of academic and cultural expression
```

```
#       Establish observed range
range(autocracy_data$v2clacfree)

#       Heatmap of media self-censorship distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2clacfree, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of acad. and cult. expr.", x = "Year")

#       Calculation of country means
free_acad_cult_expr_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2clacfree, autocracy_data$cowcode,
                       mean)
)
#       Histogram and QQ-plot of country means
ggplot(data = free_acad_cult_expr_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of acad. and cult. expr.", y = "Count") + blue_light
ggplot(data = free_acad_cult_expr_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Freedom of acad. and cult. expr.") + blue_light

#       Compute summary statistics per year
free_acad_cult_expr_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2clacfree, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2clacfree, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2clacfree, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$v2clacfree, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2clacfree, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2clacfree, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2clacfree, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_acad_cult_expr_summary_longtable <-
  pivot_longer(free_acad_cult_expr_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_acad_cult_expr_summary_longtable[
    free_acad_cult_expr_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                           "Median", "Q3",
                                                           "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Freedom of acad. and cult. expr."))
)

#       Plot skew and kurtosis over time
ggplot(data = free_acad_cult_expr_summary_longtable[
```

```
    free_acad_cult_expr_summary_longtable$Statistic %in%
      c("Skew", "Kurtosis"), ],
    aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
    labs(x = "Year", y = "Freedom of acad. and cult. expr.") + blue_light

#     Reliability analysis

#       Calculate Cronbach's Alpha by year
free_expr_cronbachalphas <-
  by(data = autocracy_data[, c("v2mecenefm", "v2meharjrn", "v2meslfcen",
                               "v2xcl_disc", "v2clacfree")],
     INDICES = autocracy_data$year, FUN = CronbachAlpha, cond = TRUE)

#       Create table for Cronbach's Alpha values
table_free_expr_cronbachalphas <-
  tibble(year = sort(unique(autocracy_data$year)),
         "All items" = NA, "Government censorship effort" = NA,
         "Journalist harassment" = NA, "Media self-censorship" = NA,
         "Freedom of discussion" = NA, "Freedom of acad. and cult. expr." = NA
  )

#       Fill in table
for (y in sort(unique(autocracy_data$year))) {
  table_free_expr_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "All items"] <-
    free_expr_cronbachalphas[[as.character(y)]][["unconditional"]]
  table_free_expr_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Government censorship effort"] <-
    free_expr_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][1, "Cronbach Alpha"]
  table_free_expr_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Journalist harassment"] <-
    free_expr_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][2, "Cronbach Alpha"]
  table_free_expr_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Media self-censorship"] <-
    free_expr_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][3, "Cronbach Alpha"]
  table_free_expr_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Freedom of discussion"] <-
    free_expr_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][4, "Cronbach Alpha"]
  table_free_expr_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Freedom of acad. and cult. expr."] <-
    free_expr_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][5, "Cronbach Alpha"]
}

#     Pivot table for use in plotting
longtab_free_expr_cronbachalphas <-
  pivot_longer(table_free_expr_cronbachalphas, 2:7, names_to = "Type",
               values_to = "Value")

#     Plot Cronbach's Alpha by year
ggplot(longtab_free_expr_cronbachalphas, aes(x = year, y = Value,
                                             colour = Type)) +
  geom_line() + blue_light

#     Calculate range of yearly Cronbach's Alpha
```

```
range(table_free_expr_cronbachalphas$"All items")

#   Original scale

#     Heatmap of freedom of expression distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2x_freexp, x = year)) +
  geom_bin2d(binwidth = c(1, 0.1)) + blue_light +
  labs(y = "Freedom of expression", x = "Year")

#     Calculation of country means
free_expr_unscaled_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2x_freexp, autocracy_data$cowcode,
                       mean)
)
#     Histogram and QQ-plot of country means
ggplot(data = free_expr_unscaled_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue") +
  labs(x = "Freedom of expression", y = "Count") + blue_light
ggplot(data = free_expr_unscaled_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Freedom of expression") +
  blue_light

#     Compute summary statistics per year
free_expr_unscaled_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2x_freexp, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2x_freexp, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2x_freexp, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$v2x_freexp, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2x_freexp, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2x_freexp, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2x_freexp, autocracy_data$year,
                           FUN = Kurt) )

#       Pivot summary statistics table for use in plotting
free_expr_unscaled_summary_longtable <-
  pivot_longer(free_expr_unscaled_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#     Plot Mean, median, first quartile, third quartile
#     and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_expr_unscaled_summary_longtable[
    free_expr_unscaled_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                          "Median", "Q3",
                                                          "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom of expression"))
)

#     Plot skew and kurtosis over time
```

```r
ggplot(data = free_expr_unscaled_summary_longtable[
  free_expr_unscaled_summary_longtable$Statistic %in% c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom of expression") + blue_light


#   Final scale descriptives

#     Entire dataset

#       Heatmap of freedom of expression distribution per year
ggplot(data = autocracy_data, mapping = aes(y = free_expr_x100, x = year)) +
  geom_bin2d(binwidth = c(1, 10)) + blue_light +
  labs(y = "Freedom of expression", x = "Year")

#       Calculation of country means
free_expr_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$free_expr_x100, autocracy_data$cowcode,
                       mean)
)
#       Histogram and QQ-plot of country means
ggplot(data = free_expr_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 10, fill = "lightblue") +
  labs(x = "Freedom of expression", y = "Count") + blue_light
ggplot(data = free_expr_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Freedom of expression") +
  blue_light

#       Compute summary statistics per year
free_expr_full_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$free_expr_x100, autocracy_data$year,
                       mean),
         Q1 = tapply(autocracy_data$free_expr_x100, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$free_expr_x100, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$free_expr_x100, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$free_expr_x100, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$free_expr_x100, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$free_expr_x100, autocracy_data$year,
                           FUN = Kurt) )

#        Pivot summary statistics table for use in plotting
free_expr_full_summary_longtable <-
  pivot_longer(free_expr_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_expr_full_summary_longtable[
    free_expr_full_summary_longtable$Statistic %in% c("Mean", "Q1",
```

```
                                                                "Median", "Q3",
                                                                "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom of expression"))
)

#       Plot skew and kurtosis over time
ggplot(data = free_expr_full_summary_longtable[
  free_expr_full_summary_longtable$Statistic %in% c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom of expression") + blue_light

#     Complete cases

#       Heatmap of freedom of expression distribution per year
ggplot(data = datacomplete, mapping = aes(y = free_expr_x100, x = year)) +
  geom_bin2d(binwidth = c(1, 10)) + blue_light +
  labs(y = "Freedom of expression", x = "Year")

#     Calculation of country means
free_expr_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$free_expr_x100,
                       datacomplete$cowcode, mean)
)

#       Histogram and QQ-plot of country means
ggplot(data = free_expr_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 10, fill = "lightblue") +
  labs(x = "Freedom of expression", y = "Count") + blue_light
ggplot(data = free_expr_complete_countrymeans,
       aes(sample = countrymean)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + labs(x = "Reference normal distribution",
                        y = "Freedom of expression") + blue_light

#       Compute summary statistics per year
free_expr_complete_summary <-
  tibble(year =  unique(datacomplete$year)[order(unique(
    datacomplete$year))],
    Mean = tapply(datacomplete$free_expr_x100, datacomplete$year, mean),
    Q1 = tapply(datacomplete$free_expr_x100, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$free_expr_x100, datacomplete$year, median),
    Q3 = tapply(datacomplete$free_expr_x100, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$free_expr_x100, datacomplete$year, sd),
    Skew = tapply(datacomplete$free_expr_x100, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$free_expr_x100, datacomplete$year,
                      FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_expr_complete_summary_longtable <-
  pivot_longer(free_expr_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
```

```
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_expr_complete_summary_longtable[
    free_expr_complete_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                           "Median", "Q3",
                                                           "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom of expression"))
)

#       Plot skew and kurtosis over time
ggplot(data = free_expr_complete_summary_longtable[
  free_expr_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Freedom of expression") + blue_light

#####################   Freedom of assembly descriptives  ####################

# Entire dataset

#   Establish observed range
range(autocracy_data$v2caassemb, na.rm = TRUE)

#   Heatmap of the orgininal freedom of assembly variable
#   distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2caassemb, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of assembly", x = "Year")

#     Calculation of country means
free_assemb_org_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2caassemb, autocracy_data$cowcode,
                       mean, na.rm = TRUE)
)

#     Histogram and QQ-plot of country means
ggplot(data = free_assemb_org_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of assembly", y = "Count") + blue_light
ggplot(data = free_assemb_org_countrymeans,
       aes(sample = countrymean)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + labs(x = "Reference normal distribution",
                        y = "Freedom of assembly") + blue_light

#     Compute summary statistics per year
free_assemb_org_summary <-
  tibble(year =  unique(autocracy_data$year)[order(unique(
    autocracy_data$year))],
    Mean = tapply(autocracy_data$v2caassemb, autocracy_data$year, mean,
                  na.rm = TRUE),
    Q1 = tapply(autocracy_data$v2caassemb, autocracy_data$year,
                quantile, prob = 0.25, na.rm = TRUE),
    Median = tapply(autocracy_data$v2caassemb, autocracy_data$year, median,
                    na.rm = TRUE),
    Q3 = tapply(autocracy_data$v2caassemb, autocracy_data$year,
                quantile, prob = 0.75, na.rm = TRUE),
    SD = tapply(autocracy_data$v2caassemb, autocracy_data$year, sd,
                na.rm = TRUE),
```

```
    Skew = tapply(autocracy_data$v2caassemb, autocracy_data$year, FUN = Skew,
                  na.rm = TRUE),
    Kurtosis = tapply(autocracy_data$v2caassemb, autocracy_data$year,
                      FUN = Kurt, na.rm = TRUE) )

#         Pivot summary statistics table for use in plotting
free_assemb_org_summary_longtable <-
  pivot_longer(free_assemb_org_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#     Plot Mean, median, first quartile, third quartile
#     and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_assemb_org_summary_longtable[
    free_assemb_org_summary_longtable$Statistic %in% c("Mean", "Q1", "Median",
                                                       "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom of assembly")) )

#       Plot skew and kurtosis over time
ggplot(data = free_assemb_org_summary_longtable[
  free_assemb_org_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Freedom of assembly") + blue_light

# Complete cases

#   Heatmap of freedom of assembly distribution per year
ggplot(data = datacomplete, mapping = aes(y = v2caassemb, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of assembly", x = "Year")

#   Calculation of country means
free_assemb_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$v2caassemb, datacomplete$cowcode, mean) )

#   Histogram and QQ-plot of country means
ggplot(data = free_assemb_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of assembly", y = "Count") + blue_light
ggplot(data = free_assemb_complete_countrymeans,
       aes(sample = countrymean)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + labs(x = "Reference normal distribution",
                        y = "Freedom of assembly") + blue_light

#   Compute summary statistics per year
free_assemb_complete_summary <-
  tibble(year =  unique(datacomplete$year)[order(unique(
    datacomplete$year))],
    Mean = tapply(datacomplete$v2caassemb, datacomplete$year, mean),
    Q1 = tapply(datacomplete$v2caassemb, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$v2caassemb, datacomplete$year, median),
    Q3 = tapply(datacomplete$v2caassemb, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$v2caassemb, datacomplete$year, sd),
```

```
    Skew = tapply(datacomplete$v2caassemb, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$v2caassemb, datacomplete$year, FUN = Kurt) )

#        Pivot summary statistics table for use in plotting
free_assemb_complete_summary_longtable <-
  pivot_longer(free_assemb_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#   Plot Mean, median, first quartile, third quartile and standard deviation
#   over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_assemb_complete_summary_longtable[
    free_assemb_complete_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                            "Median", "Q3",
                                                            "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom of assembly")) )

#   Plot skew and kurtosis over time
ggplot(data = free_assemb_complete_summary_longtable[
  free_assemb_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Freedom of assembly") + blue_light

######################   Freedom of movement descriptives    ##################

# Item descriptives

#   Freedom of foreign movement

#     Establish the observed range for the variable
range(autocracy_data$v2clfmove)

#       Heatmap of freedom of foreign movement per year
ggplot(data = autocracy_data, mapping = aes(y = v2clfmove,
                                            x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of foreign movement", x = "Year")

#       Calculation of country means
free_move_foreign_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2clfmove, autocracy_data$cowcode,
                       mean) )

#       Histogram and QQ-plot of country means
ggplot(data = free_move_foreign_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of foreign movement", y = "Count") + blue_light
ggplot(data = free_move_foreign_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Freedom of foreign movement") +
  blue_light

#       Compute summary statistics per year
free_move_foreign_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
```

```
          Mean = tapply(autocracy_data$v2clfmove, autocracy_data$year, mean),
          Q1 = tapply(autocracy_data$v2clfmove, autocracy_data$year,
                      quantile, prob = 0.25),
          Median = tapply(autocracy_data$v2clfmove, autocracy_data$year, median),
          Q3 = tapply(autocracy_data$v2clfmove, autocracy_data$year,
                      quantile, prob = 0.75),
          SD = tapply(autocracy_data$v2clfmove, autocracy_data$year, sd),
          Skew = tapply(autocracy_data$v2clfmove, autocracy_data$year,
                        FUN = Skew),
          Kurtosis = tapply(autocracy_data$v2clfmove, autocracy_data$year,
                            FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_move_foreign_summary_longtable <-
  pivot_longer(free_move_foreign_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_move_foreign_summary_longtable[
    free_move_foreign_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                         "Median", "Q3",
                                                         "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Freedom of foreign movement")) )

#       Plot skew and kurtosis over time
ggplot(data = free_move_foreign_summary_longtable[
  free_move_foreign_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Freedom of foreign movement") + blue_light

#   Freedom of domestic movement for men

#     Establish the observed range for the variable
range(autocracy_data$v2cldmovem)

#       Heatmap of freedom of domestic movement for men distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2cldmovem, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of domestic movement for men", x = "Year")

#       Calculation of country means
free_move_men_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2cldmovem, autocracy_data$cowcode,
                       mean)
)
#       Histogram and QQ-plot of country means
ggplot(data = free_move_men_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of domestic movement for men", y = "Count") + blue_light
ggplot(data = free_move_men_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Freedom of domestic movement for men") + blue_light
```

```
#       Compute summary statistics per year
free_move_men_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2cldmovem, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2cldmovem, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2cldmovem, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$v2cldmovem, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2cldmovem, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2cldmovem, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2cldmovem, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_move_men_summary_longtable <-
  pivot_longer(free_move_men_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_move_men_summary_longtable[
    free_move_men_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                     "Median", "Q3",
                                                     "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Freedom of domestic movement for men")) )

#       Plot skew and kurtosis over time
ggplot(data = free_move_men_summary_longtable[
  free_move_men_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],  aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Freedom of domestic movement for men") +
  blue_light

#   Freedom of domestic movement for women

#     Establish the observed range for the variable
range(autocracy_data$v2cldmovew)

#       Heatmap of freedom of domestic movement for women distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2cldmovew, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of domestic movement for women", x = "Year")

#       Calculation of country means
free_move_women_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2cldmovew, autocracy_data$cowcode,
                       mean) )

#       Histogram and QQ-plot of country means
ggplot(data = free_move_women_countrymeans, aes(x = countrymean)) +
```

```
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of domestic movement for women", y = "Count") +
  blue_light
ggplot(data = free_move_women_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Freedom of domestic movement for women") + blue_light

#       Compute summary statistics per year
free_move_women_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2cldmovew, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2cldmovew, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2cldmovew, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$v2cldmovew, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2cldmovew, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2cldmovew, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2cldmovew, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_move_women_summary_longtable <-
  pivot_longer(free_move_women_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_move_women_summary_longtable[
    free_move_women_summary_longtable$Statistic %in% c("Mean", "Q1", "Median",
                                                       "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light +
      labs(x = "Year", y = "Freedom of domestic movement for women")) )

#       Plot skew and kurtosis over time
ggplot(data = free_move_women_summary_longtable[
  free_move_women_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Freedom of domestic movement for women") +
  blue_light

#     Reliability analysis

#       Calculate Cronbach's Alpha by year
free_move_cronbachalphas <-
  by(data = autocracy_data[, c("v2clfmove", "v2cldmovem", "v2cldmovew")],
     INDICES = autocracy_data$year, FUN = CronbachAlpha, cond = TRUE,
     na.rm = TRUE)

#       Create table for yearly Cronbach's Alpha values
table_free_move_cronbachalphas <-
  tibble(year = sort(unique(autocracy_data$year)),
         "All items" = NA, "Freedom of foreign movement" = NA,
```

```
         "Freedom of domestic movement for men" = NA,
         "Freedom of domestic movement for women" = NA)

#     Fill table
for (y in sort(unique(autocracy_data$year))) {
  table_free_move_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "All items"] <-
    free_move_cronbachalphas[[as.character(y)]][["unconditional"]]
  table_free_move_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Freedom of foreign movement"] <-
    free_move_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][1, "Cronbach Alpha"]
  table_free_move_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Freedom of domestic movement for men"] <-
    free_move_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][2, "Cronbach Alpha"]
  table_free_move_cronbachalphas[table_free_expr_cronbachalphas$year == y,
                                 "Freedom of domestic movement for women"] <-
    free_move_cronbachalphas[[as.character(y)]][["condCronbachAlpha"
    ]][3, "Cronbach Alpha"]
}

#       Pivot table for use in plotting
longtab_free_move_cronbachalphas <-
  pivot_longer(table_free_move_cronbachalphas, 2:5, names_to = "Type",
               values_to = "Value")

#       Plot Cronbach's Alpha by year
ggplot(longtab_free_move_cronbachalphas, aes(x = year, y = Value,
                                             colour = Type)) +
  geom_line() + blue_light

#       Calculate range of yearly Cronbach's Alpha values
range(table_free_move_cronbachalphas$"All items")

#   Scale descriptives

#     Entire dataset

#       Heatmap of freedom of movement distribution per year
ggplot(data = autocracy_data, mapping = aes(y = freedom_movement, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of movement", x = "Year")

#       Calculation of country means
free_move_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$freedom_movement, autocracy_data$cowcode,
                       mean)
)
#       Histogram and QQ-plot of country means
ggplot(data = free_move_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of movement", y = "Count") + blue_light
ggplot(data = free_move_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Freedom of movement") +
  blue_light
```

```
#       Compute summary statistics per year
free_move_full_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$freedom_movement, autocracy_data$year,
                       mean),
         Q1 = tapply(autocracy_data$freedom_movement, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$freedom_movement, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$freedom_movement, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$freedom_movement, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$freedom_movement, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$freedom_movement, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_move_full_summary_longtable <-
  pivot_longer(free_move_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_move_full_summary_longtable[
    free_move_full_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                      "Median", "Q3",
                                                      "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom of movement")) )

#       Plot skew and kurtosis over time
ggplot(data = free_move_full_summary_longtable[
  free_move_full_summary_longtable$Statistic %in% c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom of movement") + blue_light

#     Complete cases

#       Heatmap of freedom of movement distribution per year
ggplot(data = datacomplete, mapping = aes(y = freedom_movement, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom of movement", x = "Year")

#     Calculation of country means
free_move_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$freedom_movement, datacomplete$cowcode,
                       mean) )

#       Histogram and QQ-plot of country means
ggplot(data = free_move_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom of movement", y = "Count") + blue_light
ggplot(data = free_move_complete_countrymeans, aes(sample = countrymean)) +
```

```
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Freedom of movement") +
  blue_light

#       Compute summary statistics per year
free_move_complete_summary <-
  tibble(year =  unique(datacomplete$year)[order(unique(
    datacomplete$year))],
    Mean = tapply(datacomplete$freedom_movement, datacomplete$year, mean),
    Q1 = tapply(datacomplete$freedom_movement, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$freedom_movement, datacomplete$year, median),
    Q3 = tapply(datacomplete$freedom_movement, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$freedom_movement, datacomplete$year, sd),
    Skew = tapply(datacomplete$freedom_movement, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$freedom_movement, datacomplete$year,
                      FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_move_complete_summary_longtable <-
  pivot_longer(free_move_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_move_complete_summary_longtable[
    free_move_complete_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                          "Median", "Q3",
                                                          "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom of movement")) )

#       Plot skew and kurtosis over time
ggplot(data = free_move_complete_summary_longtable[
  free_move_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Freedom of movement") + blue_light

##########   Protection of life and physical integrity descriptives    #########

# Item descriptives

#   Freedom from torture

#     Establish observed range
range(autocracy_data$v2cltort)

#         Heatmap of media freedom of discussion for men distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2cltort, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Freedom from torture", x = "Year")

#         Calculation of country means
free_tort_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
```

```
  countrymean = tapply(autocracy_data$v2cltort, autocracy_data$cowcode,
                       mean)
)
#         Histogram and QQ-plot of country means
ggplot(data = free_tort_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom from torture", y = "Count") + blue_light
ggplot(data = free_tort_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Freedom from torture") +
  blue_light

#         Compute summary statistics per year
free_tort_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2cltort, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2cltort, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2cltort, autocracy_data$year, median),
         Q3 = tapply(autocracy_data$v2cltort, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2cltort, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2cltort, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2cltort, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_tort_summary_longtable <-
  pivot_longer(free_tort_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#         Plot Mean, median, first quartile, third quartile
#         and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_tort_summary_longtable[
    free_tort_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                 "Median", "Q3",
                                                 "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Freedom from torture")) )

#         Plot skew and kurtosis over time
ggplot(data = free_tort_summary_longtable[
  free_tort_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom from torture") + blue_light


#   Freedom from political killings

#         Establish observed range
range(autocracy_data$v2clkill)

#         Heatmap of freedom of discussion for women distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2clkill, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
```

```
  labs(y = "Freedom from political killings", x = "Year")

#       Calculation of country means
free_kill_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2clkill, autocracy_data$cowcode,
                       mean) )
#         Histogram and QQ-plot of country means
ggplot(data = free_kill_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Freedom from political killings", y = "Count") + blue_light
ggplot(data = free_kill_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Freedom from political killings") + blue_light

#         Compute summary statistics per year
free_kill_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2clkill, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2clkill, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2clkill, autocracy_data$year, median),
         Q3 = tapply(autocracy_data$v2clkill, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2clkill, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2clkill, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2clkill, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
free_kill_summary_longtable <-
  pivot_longer(free_kill_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#         Plot Mean, median, first quartile, third quartile
#         and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = free_kill_summary_longtable[
    free_kill_summary_longtable$Statistic %in% c("Mean", "Q1", "Median", "Q3",
                                                 "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year",
                                      y = "Freedom from political killings")) )

#         Plot skew and kurtosis over time
ggplot(data = free_kill_summary_longtable[
  free_kill_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Freedom from political killings") + blue_light

#       Item intercorrelation

#         Calculate correlations by year
life_phys_cors <-
```

```
  by(data = autocracy_data[, c("v2cltort", "v2clkill")],
     INDICES = autocracy_data$year, FUN = cor)

#         Create table for correlations
table_life_phys_cors <-
  tibble(Year = sort(unique(autocracy_data$year)), Correlation = NA)

#         Fill table
for (y in sort(unique(autocracy_data$year))) {
  table_life_phys_cors[table_life_phys_cors$Year == y,
                       "Correlation"] <- life_phys_cors[[as.character(y)]][2,1]
  }

#         Plot yearly correlations
ggplot(table_life_phys_cors, aes(x = Year, y = Correlation)) +
  geom_line(colour = "tomato") + blue_light

#         Establish range for yearly correlations
range(table_life_phys_cors$Correlation)

# Scale descriptives

#   Entire dataset

#     Establish observed range for scale
range(autocracy_data$life_phys_x100)

#     Heatmap of protection of life and physical integrity distribution per year
ggplot(data = autocracy_data, mapping = aes(y = life_phys_x100, x = year)) +
  geom_bin2d(binwidth = c(1, 10)) + blue_light +
  labs(y = "Protection of life and physical integrity", x = "Year")

#     Calculation of country means
full_life_phys_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$life_phys_x100, autocracy_data$cowcode,
                       mean)
)
#     Histogram and QQ-plot of country means
ggplot(data = full_life_phys_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 10, fill = "lightblue") +
  labs(x = "Protection of life and physical integrity", y = "Count") +
  blue_light
ggplot(data = full_life_phys_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Protection of life and physical integrity") + blue_light

#     Compute scale summary statistics per year
full_life_phys_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$life_phys_x100, autocracy_data$year,
                       mean),
         Q1 = tapply(autocracy_data$life_phys_x100, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$life_phys_x100, autocracy_data$year,
                         median),
         Q3 = tapply(autocracy_data$life_phys_x100, autocracy_data$year,
```

```
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$life_phys_x100, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$life_phys_x100, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$life_phys_x100, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
full_life_phys_summary_longtable <-
  pivot_longer(full_life_phys_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#     Plot mean, median, first quartile, third quartile
#     and standard deviation of scale over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = full_life_phys_summary_longtable[
    full_life_phys_summary_longtable$Statistic %in% c("Mean", "Q1", "Median",
                                                       "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Protection of life and physical integrity")) )

#     Plot skew and kurtosis of scale over time
ggplot(data = full_life_phys_summary_longtable[
  full_life_phys_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Protection of life and physical integrity") + blue_light

#   Complete cases

#     Establish observed range for scale
range(datacomplete$life_phys_x100)

#     Heatmap of protection of life and physical integrity distribution per year
ggplot(data = datacomplete, mapping = aes(y = life_phys_x100, x = year)) +
  geom_bin2d(binwidth = c(1, 10)) + blue_light +
  labs(y = "Protection of life and physical integrity", x = "Year")

#     Calculation of country means
complete_life_phys_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$life_phys_x100, datacomplete$cowcode,
                       mean)
)
#     Histogram and QQ-plot of country means
ggplot(data = complete_life_phys_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 10, fill = "lightblue") +
  labs(x = "Protection of life and physical integrity", y = "Count") +
  blue_light
ggplot(data = complete_life_phys_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution",
       y = "Protection of life and physical integrity") + blue_light

#     Compute scale summary statistics per year
complete_life_phys_summary <-
  tibble(year = sort(unique(datacomplete$year)),
```

```
          Mean = tapply(datacomplete$life_phys_x100, datacomplete$year, mean),
          Q1 = tapply(datacomplete$life_phys_x100, datacomplete$year,
                      quantile, prob = 0.25),
          Median = tapply(datacomplete$life_phys_x100, datacomplete$year,
                          median),
          Q3 = tapply(datacomplete$life_phys_x100, datacomplete$year,
                      quantile, prob = 0.75),
          SD = tapply(datacomplete$life_phys_x100, datacomplete$year, sd),
          Skew = tapply(datacomplete$life_phys_x100, datacomplete$year,
                        FUN = Skew),
          Kurtosis = tapply(datacomplete$life_phys_x100, datacomplete$year,
                            FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
complete_life_phys_summary_longtable <-
  pivot_longer(complete_life_phys_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#     Plot mean, median, first quartile, third quartile
#     and standard deviation of scale over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = complete_life_phys_summary_longtable[
    complete_life_phys_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                          "Median", "Q3",
                                                          "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Protection of life and physical integrity")) )

#     Plot skew and kurtosis of scale over time
ggplot(data = complete_life_phys_summary_longtable[
  complete_life_phys_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Protection of life and physical integrity") + blue_light

######################   Population descriptives    ###########################

# Original variable

#       Heatmap of population distribution per year
ggplot(data = autocracy_data, mapping = aes(y = e_mipopula, x = year)) +
  geom_bin2d(binwidth = c(1, 10000)) + blue_light +
  labs(y = "Population (thousands)", x = "Year")
ggplot(data = autocracy_data[autocracy_data$cowcode == 710,],
       mapping = aes(y = e_mipopula, x = year)) +
  geom_bin2d(binwidth = c(1, 10000)) + blue_light +
  labs(y = "Population (thousands)", x = "Year")

#       Calculation of country means
population_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$e_mipopula, autocracy_data$cowcode,
                       mean, na.rm = TRUE)
)
#       Histogram and QQ-plot of country means
ggplot(data = population_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 10000, fill = "lightblue") +
```

```
  labs(x = "Population (thousands)", y = "Count") + blue_light
ggplot(data = population_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Population (thousands)") +
  blue_light

#       Compute summary statistics per year
population_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$e_mipopula, autocracy_data$year, mean,
                       na.rm = TRUE),
         Q1 = tapply(autocracy_data$e_mipopula, autocracy_data$year,
                     quantile, prob = 0.25, na.rm = TRUE),
         Median = tapply(autocracy_data$e_mipopula, autocracy_data$year,
                         median, na.rm = TRUE),
         Q3 = tapply(autocracy_data$e_mipopula, autocracy_data$year,
                     quantile, prob = 0.75, na.rm = TRUE),
         SD = tapply(autocracy_data$e_mipopula, autocracy_data$year,
                     sd, na.rm = TRUE),
         Skew = tapply(autocracy_data$e_mipopula, autocracy_data$year,
                       FUN = Skew, na.rm = TRUE),
         Kurtosis = tapply(autocracy_data$e_mipopula, autocracy_data$year,
                           FUN = Kurt, na.rm = TRUE) )

#         Pivot summary statistics table for use in plotting
population_summary_longtable <-
  pivot_longer(population_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = population_summary_longtable[
    population_summary_longtable$Statistic %in% c("Mean", "Q1", "Median", "Q3",
                                                   "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Population (thousands)"))
)

#       Plot skew and kurtosis over time
ggplot(data = population_summary_longtable[
  population_summary_longtable$Statistic %in% c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Population (thousands)") + blue_light

# Logged variable

#   Entire dataset

#       Heatmap of population (log 10) distribution per year
ggplot(data = autocracy_data, mapping = aes(y = log10pop, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Population (log 10)", x = "Year")

#       Calculation of country means
log_population_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
```

```
  countrymean = tapply(autocracy_data$log10pop, autocracy_data$cowcode,
                       mean, na.rm = TRUE)
)
#       Histogram and QQ-plot of country means
ggplot(data = log_population_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Population (log 10)", y = "Count") + blue_light
ggplot(data = log_population_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Population (log 10)") +
  blue_light

#       Compute summary statistics per year
log_population_full_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$log10pop, autocracy_data$year, mean,
                       na.rm = TRUE),
         Q1 = tapply(autocracy_data$log10pop, autocracy_data$year,
                     quantile, prob = 0.25, na.rm = TRUE),
         Median = tapply(autocracy_data$log10pop, autocracy_data$year, median,
                         na.rm = TRUE),
         Q3 = tapply(autocracy_data$log10pop, autocracy_data$year,
                     quantile, prob = 0.75, na.rm = TRUE),
         SD = tapply(autocracy_data$log10pop, autocracy_data$year, sd,
                     na.rm = TRUE),
         Skew = tapply(autocracy_data$log10pop, autocracy_data$year, FUN = Skew,
                       na.rm = TRUE),
         Kurtosis = tapply(autocracy_data$log10pop, autocracy_data$year,
                           FUN = Kurt, na.rm = TRUE) )

#         Pivot summary statistics table for use in plotting
log_population_full_summary_longtable <-
  pivot_longer(log_population_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = log_population_full_summary_longtable[
    log_population_full_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                           "Median", "Q3",
                                                           "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +  geom_line() + blue_light +
      labs(x = "Year", y = "Population (log 10)")) )

#       Plot skew and kurtosis over time
ggplot(data = log_population_full_summary_longtable[
  log_population_full_summary_longtable$Statistic %in% c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Population (log 10)") + blue_light

#   Complete cases

#       Heatmap of population (log 10) distribution per year
ggplot(data = datacomplete, mapping = aes(y = log10pop, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Population (log 10)", x = "Year")
```

```
#     Calculation of country means
log_population_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$log10pop, datacomplete$cowcode,
                       mean) )

#      Histogram and QQ-plot of country means
ggplot(data = log_population_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Population (log 10)", y = "Count") + blue_light
ggplot(data = log_population_complete_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Population (log 10)") +
  blue_light

# Calculate mean population level
mean(tapply(10^(datacomplete$log10pop), datacomplete$cowcode, mean,
            na.rm = TRUE), na.rm = TRUE)

#      Compute summary statistics per year
log_population_complete_summary <-
  tibble(year =  unique(datacomplete$year)[order(unique(
    datacomplete$year))],
    Mean = tapply(datacomplete$log10pop, datacomplete$year, mean),
    Q1 = tapply(datacomplete$log10pop, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$log10pop, datacomplete$year, median),
    Q3 = tapply(datacomplete$log10pop, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$log10pop, datacomplete$year, sd),
    Skew = tapply(datacomplete$log10pop, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$log10pop, datacomplete$year, FUN = Kurt) )

#        Pivot summary statistics table for use in plotting
log_population_complete_summary_longtable <-
  pivot_longer(log_population_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#      Plot Mean, median, first quartile, third quartile
#      and standard deviation over time

withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = log_population_complete_summary_longtable[
    log_population_complete_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                               "Median", "Q3",
                                                               "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Population (log 10)")) )

#      Plot skew and kurtosis over time
ggplot(data = log_population_complete_summary_longtable[
  log_population_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Population (log 10)") + blue_light

####################   GDP per capita (ln) descriptives    ####################

```

```
# Entire dataset

#       Heatmap of GDP per capita (ln) distribution per year
ggplot(data = autocracy_data, mapping = aes(y = e_migdppcln, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "GDP per capita (ln)", x = "Year")

#       Calculation of country means
GDP_cap_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$e_migdppcln, autocracy_data$cowcode,
                       mean, na.rm = TRUE) )

#       Histogram and QQ-plot of country means
ggplot(data = GDP_cap_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "GDP per capita (ln)", y = "Count") + blue_light
ggplot(data = GDP_cap_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "GDP per capita (ln)") +
  blue_light

#       Compute summary statistics per year
GDP_cap_full_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$e_migdppcln, autocracy_data$year, mean,
                       na.rm = TRUE),
         Q1 = tapply(autocracy_data$e_migdppcln, autocracy_data$year,
                     quantile, prob = 0.25, na.rm = TRUE),
         Median = tapply(autocracy_data$e_migdppcln, autocracy_data$year,
                         median, na.rm = TRUE),
         Q3 = tapply(autocracy_data$e_migdppcln, autocracy_data$year,
                     quantile, prob = 0.75, na.rm = TRUE),
         SD = tapply(autocracy_data$e_migdppcln, autocracy_data$year, sd, na.rm =
TRUE),
         Skew = tapply(autocracy_data$e_migdppcln, autocracy_data$year,
                       FUN = Skew, na.rm = TRUE),
         Kurtosis = tapply(autocracy_data$e_migdppcln, autocracy_data$year,
                           FUN = Kurt, na.rm = TRUE) )

#         Pivot summary statistics table for use in plotting
GDP_cap_full_summary_longtable <-
  pivot_longer(GDP_cap_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = GDP_cap_full_summary_longtable[
    GDP_cap_full_summary_longtable$Statistic %in% c("Mean", "Q1", "Median",
                                                    "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "GDP per capita (ln)")) )

#       Plot skew and kurtosis over time
ggplot(data = GDP_cap_full_summary_longtable[
  GDP_cap_full_summary_longtable$Statistic %in% c("Skew", "Kurtosis"), ],
```

```
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "GDP per capita (ln)") + blue_light

#     Complete cases

#       Heatmap of GDP per capita (ln) distribution per year
ggplot(data = datacomplete, mapping = aes(y = e_migdppcln, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "GDP per capita (ln)", x = "Year")

#     Calculation of country means
GDP_cap_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$e_migdppcln, datacomplete$cowcode,
                       mean)
)

#       Histogram and QQ-plot of country means
ggplot(data = GDP_cap_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "GDP per capita (ln)", y = "Count") + blue_light
ggplot(data = GDP_cap_complete_countrymeans,
       aes(sample = countrymean)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + labs(x = "Reference normal distribution",
                        y = "GDP per capita (ln)") + blue_light

# Calculate mean country mean GDP per capita
mean(tapply(exp(datacomplete$e_migdppcln), datacomplete$cowcode, mean,
            na.rm = TRUE), na.rm = TRUE)

#       Compute summary statistics per year
GDP_cap_complete_summary <-
  tibble(year =  unique(datacomplete$year)[order(unique(
    datacomplete$year))],
    Mean = tapply(datacomplete$e_migdppcln, datacomplete$year, mean),
    Q1 = tapply(datacomplete$e_migdppcln, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$e_migdppcln, datacomplete$year, median),
    Q3 = tapply(datacomplete$e_migdppcln, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$e_migdppcln, datacomplete$year, sd),
    Skew = tapply(datacomplete$e_migdppcln, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$e_migdppcln, datacomplete$year, FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
GDP_cap_complete_summary_longtable <-
  pivot_longer(GDP_cap_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time

withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = GDP_cap_complete_summary_longtable[
    GDP_cap_complete_summary_longtable$Statistic %in% c("Mean", "Q1", "Median",
                                                        "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
```

```
      labs(x = "Year", y = "GDP per capita (ln)")) )

#       Plot skew and kurtosis over time
ggplot(data = GDP_cap_complete_summary_longtable[
  GDP_cap_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "GDP per capita (ln)") + blue_light

#####################   GDP growth descriptives    ############################

# Entire dataset

#   Heatmap of GDP growth distribution per year
ggplot(data = autocracy_data, mapping = aes(y = e_migdpgro, x = year)) +
  geom_bin2d(binwidth = c(1, 0.01)) + blue_light +
  labs(y = "GDP growth", x = "Year")

#   View outliers
autocracy_data[abs(autocracy_data$e_migdpgro) > 0.5 &
                 is.na(autocracy_data$e_migdpgro) == FALSE,
               c("cowcode", "year", "gwf_country", "e_migdpgro")]

#   Calculation of country means
GDP_growth_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$e_migdpgro, autocracy_data$cowcode,
                       mean, na.rm = TRUE) )

#   Histogram and QQ-plot of country means
ggplot(data = GDP_growth_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.01, fill = "lightblue") +
  labs(x = "GDP growth", y = "Count") + blue_light
ggplot(data = GDP_growth_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "GDP growth") + blue_light

#   Compute summary statistics per year
GDP_growth_full_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$e_migdpgro, autocracy_data$year, mean,
                       na.rm = TRUE),
         Q1 = tapply(autocracy_data$e_migdpgro, autocracy_data$year,
                     quantile, prob = 0.25, na.rm = TRUE),
         Median = tapply(autocracy_data$e_migdpgro, autocracy_data$year, median,
                         na.rm = TRUE),
         Q3 = tapply(autocracy_data$e_migdpgro, autocracy_data$year,
                     quantile, prob = 0.75, na.rm = TRUE),
         SD = tapply(autocracy_data$e_migdpgro, autocracy_data$year, sd,
                     na.rm = TRUE),
         Skew = tapply(autocracy_data$e_migdpgro, autocracy_data$year,
                       FUN = Skew, na.rm = TRUE),
         Kurtosis = tapply(autocracy_data$e_migdpgro, autocracy_data$year,
                           FUN = Kurt, na.rm = TRUE) )

#        Pivot summary statistics table for use in plotting
GDP_growth_full_summary_longtable <-
  pivot_longer(GDP_growth_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")
```

```
#   Plot Mean, median, first quartile, third quartile and standard deviation
#   over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = GDP_growth_full_summary_longtable[
    GDP_growth_full_summary_longtable$Statistic %in% c("Mean", "Q1", "Median",
                                                       "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +  geom_line() + blue_light +
      labs(x = "Year", y = "GDP growth")) )

#   Plot skew and kurtosis over time
ggplot(data = GDP_growth_full_summary_longtable[
  GDP_growth_full_summary_longtable$Statistic %in% c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "GDP growth") + blue_light

# Complete cases

#   Heatmap of GDP growth distribution per year
ggplot(data = datacomplete, mapping = aes(y = e_migdpgro, x = year)) +
  geom_bin2d(binwidth = c(1, 0.01)) + blue_light +
  labs(y = "GDP growth", x = "Year")

#   See which outliers remain
datacomplete[abs(datacomplete$e_migdpgro) > 0.5 &
                 is.na(datacomplete$e_migdpgro) == FALSE,
               c("cowcode", "year", "gwf_country", "e_migdpgro")]


#   Calculation of country means
GDP_growth_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$e_migdpgro, datacomplete$cowcode,
                       mean) )

#   Histogram and QQ-plot of country means
ggplot(data = GDP_growth_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.01, fill = "lightblue") +
  labs(x = "GDP growth", y = "Count") + blue_light
ggplot(data = GDP_growth_complete_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "GDP growth") + blue_light

#   Compute summary statistics per year
GDP_growth_complete_summary <-
  tibble(year =  unique(datacomplete$year)[order(unique(
    datacomplete$year))],
    Mean = tapply(datacomplete$e_migdpgro, datacomplete$year, mean),
    Q1 = tapply(datacomplete$e_migdpgro, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$e_migdpgro, datacomplete$year, median),
    Q3 = tapply(datacomplete$e_migdpgro, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$e_migdpgro, datacomplete$year, sd),
    Skew = tapply(datacomplete$e_migdpgro, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$e_migdpgro, datacomplete$year, FUN = Kurt) )
```

```
#         Pivot summary statistics table for use in plotting
GDP_growth_complete_summary_longtable <-
  pivot_longer(GDP_growth_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#   Plot Mean, median, first quartile, third quartile and standard deviation
#   over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = GDP_growth_complete_summary_longtable[
    GDP_growth_complete_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                           "Median", "Q3",
                                                           "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "GDP growth")) )

#   Plot skew and kurtosis over time
ggplot(data = GDP_growth_complete_summary_longtable[
  GDP_growth_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "GDP growth") + blue_light

######################   Regime type descriptives   ###########################

# Check for coding errors in regime type
table(list(autocracy_data$gwf_party, autocracy_data$gwf_personal,
           autocracy_data$gwf_military, autocracy_data$gwf_monarch))

# Create a single regime type factor
autocracy_data$regime_type[autocracy_data$gwf_party == 1] <- "Single-party"
autocracy_data$regime_type[autocracy_data$gwf_personal == 1] <- "Personalist"
autocracy_data$regime_type[autocracy_data$gwf_military == 1] <- "Military"
autocracy_data$regime_type[autocracy_data$gwf_monarch == 1] <- "Monarch"
datacomplete$regime_type[datacomplete$gwf_party == 1] <- "Single-party"
datacomplete$regime_type[datacomplete$gwf_personal == 1] <- "Personalist"
datacomplete$regime_type[datacomplete$gwf_military == 1] <- "Military"
datacomplete$regime_type[datacomplete$gwf_monarch == 1] <- "Monarch"

# Entire dataset

#       Table and chi-square test
full_regime_type_table <- table(autocracy_data$regime_type, autocracy_data$year)
full_regime_type_table
summary(full_regime_type_table)
chisq.test(autocracy_data$regime_type, autocracy_data$year)$stdres
chisq.test(autocracy_data$regime_type, autocracy_data$year)$expected
fisher.test(autocracy_data$regime_type, autocracy_data$year,
            simulate.p.value = TRUE, B = 10000)

#       Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = regime_type)) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Regime type")

# Complete cases

#       Table and chi-square test
complete_regime_type_table <- table(datacomplete$regime_type, datacomplete$year)
complete_regime_type_table
```

```
summary(complete_regime_type_table)
chisq.test(datacomplete$regime_type, datacomplete$year)$expected
fisher.test(datacomplete$regime_type, datacomplete$year,
            simulate.p.value = TRUE, B = 10000)

#       Stacked barplot per year
ggplot(datacomplete, aes(x = year, fill = regime_type)) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Regime type")

######################   International conflict descriptives    ################

# Entire dataset

#   Table and chi-square test
full_international_conflict_table <- table(autocracy_data$e_miinteco,
                                           autocracy_data$year)
full_international_conflict_table
chisq.test(autocracy_data$e_miinteco, autocracy_data$year)
chisq.test(autocracy_data$e_miinteco, autocracy_data$year)$expected
fisher.test(autocracy_data$e_miinteco, autocracy_data$year,
            simulate.p.value = TRUE, B = 10000)

#   Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(e_miinteco))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "International conflict") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("No international conflict",
                               "International conflict"))

# Complete cases

#   Table and chi-square test
complete_international_conflict_table <- table(datacomplete$e_miinteco,
                                               datacomplete$year)
complete_international_conflict_table
summary(complete_international_conflict_table)
chisq.test(datacomplete$e_miinteco, datacomplete$year)$expected
fisher.test(datacomplete$e_miinteco, datacomplete$year,
            simulate.p.value = TRUE, B = 10000)

#   Stacked barplot per year
ggplot(datacomplete, aes(x = year, fill = as.factor(e_miinteco))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "International conflict") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("No international conflict",
                               "International conflict"))

######################   Internal conflict descriptives   ######################

# Entire dataset

#   Table and chi-square test
full_internal_conflict_table <- table(autocracy_data$e_miinterc,
                                      autocracy_data$year)
full_internal_conflict_table
chisq.test(autocracy_data$e_miinterc, autocracy_data$year)
```

```
chisq.test(autocracy_data$e_miinterc, autocracy_data$year)$expected

#   Stacked barplot per year
ggplot(autocracy_data, aes(x = year, fill = as.factor(e_miinterc))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Internal conflict") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("No internal conflict",
                               "Internal conflict"))

# Complete cases

#   Table and chi-square test
complete_internal_conflict_table <- table(datacomplete$e_miinterc,
                                          datacomplete$year)
complete_internal_conflict_table
summary(complete_internal_conflict_table)
chisq.test(datacomplete$e_miinterc, datacomplete$year)$expected
fisher.test(autocracy_data$e_miinterc, autocracy_data$year,
            simulate.p.value = TRUE, B = 10000)

#   Stacked barplot per year
ggplot(datacomplete, aes(x = year, fill = as.factor(e_miinterc))) +
  geom_bar(position = "fill") + blue_light +
  labs(x = "Year", y = "Proportion", fill = "Internal conflict") +
  scale_fill_manual(values = c("lightblue", "darkgreen"),
                    labels = c("No internal conflict", "Internal conflict"))

######################   Political violence descriptives    ####################

# Entire dataset

#       Heatmap of political violence distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2caviol, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Political violence", x = "Year")

#       Calculation of country means
political_violence_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2caviol, autocracy_data$cowcode,
                       mean, na.rm = TRUE)
)
#       Histogram and QQ-plot of country means
ggplot(data = political_violence_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Political violence", y = "Count") + blue_light
ggplot(data = political_violence_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") +  geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Political violence") +
  blue_light

#       Compute summary statistics per year
political_violence_full_summary <-
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2caviol, autocracy_data$year, mean,
                       na.rm = TRUE),
         Q1 = tapply(autocracy_data$v2caviol, autocracy_data$year,
```

```
                     quantile, prob = 0.25, na.rm = TRUE),
         Median = tapply(autocracy_data$v2caviol, autocracy_data$year, median,
                         na.rm = TRUE),
         Q3 = tapply(autocracy_data$v2caviol, autocracy_data$year,
                     quantile, prob = 0.75, na.rm = TRUE),
         SD = tapply(autocracy_data$v2caviol, autocracy_data$year, sd,
                     na.rm = TRUE),
         Skew = tapply(autocracy_data$v2caviol, autocracy_data$year, FUN = Skew,
                       na.rm = TRUE),
         Kurtosis = tapply(autocracy_data$v2caviol, autocracy_data$year,
                           FUN = Kurt, na.rm = TRUE) )

#         Pivot summary statistics table for use in plotting
political_violence_full_summary_longtable <-
  pivot_longer(political_violence_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = political_violence_full_summary_longtable[
    political_violence_full_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                               "Median", "Q3",
                                                               "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) +
      geom_line() + blue_light + labs(x = "Year", y = "Political violence")) )

#       Plot skew and kurtosis over time
ggplot(data = political_violence_full_summary_longtable[
  political_violence_full_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ],
  aes(x = year, y = Value, linetype = Statistic)) + geom_line() +
  labs(x = "Year", y = "Political violence") + blue_light

#     Complete cases

#       Heatmap of political violence distribution per year
ggplot(data = datacomplete, mapping = aes(y = v2caviol, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Political violence", x = "Year")

#     Calculation of country means
political_violence_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$v2caviol, datacomplete$cowcode, mean) )

#       Histogram and QQ-plot of country means
ggplot(data = political_violence_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Political violence", y = "Count") + blue_light
ggplot(data = political_violence_complete_countrymeans,
       aes(sample = countrymean)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + labs(x = "Reference normal distribution",
                        y = "Political violence") + blue_light

#       Compute summary statistics per year
political_violence_complete_summary <-
```

```
  tibble(year =  unique(datacomplete$year)[order(unique(datacomplete$year))],
    Mean = tapply(datacomplete$v2caviol, datacomplete$year, mean),
    Q1 = tapply(datacomplete$v2caviol, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$v2caviol, datacomplete$year, median),
    Q3 = tapply(datacomplete$v2caviol, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$v2caviol, datacomplete$year, sd),
    Skew = tapply(datacomplete$v2caviol, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$v2caviol, datacomplete$year, FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
political_violence_complete_summary_longtable <-
  pivot_longer(political_violence_complete_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = political_violence_complete_summary_longtable[
    political_violence_complete_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                                   "Median",
                                                                   "Q3", "SD"),
    ], aes(x = year, y = Value, colour = Statistic)) + geom_line() +
      blue_light + labs(x = "Year", y = "Political violence")) )

#       Plot skew and kurtosis over time
ggplot(data = political_violence_complete_summary_longtable[
  political_violence_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Political violence") + blue_light

#######   Rigour and impartiality public administration descriptives    ########

# Entire dataset

#       Heatmap of political violence distribution per year
ggplot(data = autocracy_data, mapping = aes(y = v2clrspct, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Rig. & impart. pub. admin.", x = "Year")

#       Calculation of country means
rigour_impartiality_full_countrymeans <- tibble(
  countrycode = unique(autocracy_data$cowcode)[order(unique(
    autocracy_data$cowcode))],
  countrymean = tapply(autocracy_data$v2clrspct, autocracy_data$cowcode, mean) )

#       Histogram and QQ-plot of country means
ggplot(data = rigour_impartiality_full_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Rig. & impart. pub. admin.", y = "Count") + blue_light
ggplot(data = rigour_impartiality_full_countrymeans, aes(sample = countrymean)) +
  geom_qq(colour = "lightblue") + geom_qq_line() +
  labs(x = "Reference normal distribution", y = "Rig. & impart. pub. admin.") +
  blue_light

#       Compute summary statistics per year
rigour_impartiality_full_summary <-
```

```
  tibble(year = sort(unique(autocracy_data$year)),
         Mean = tapply(autocracy_data$v2clrspct, autocracy_data$year, mean),
         Q1 = tapply(autocracy_data$v2clrspct, autocracy_data$year,
                     quantile, prob = 0.25),
         Median = tapply(autocracy_data$v2clrspct, autocracy_data$year, median),
         Q3 = tapply(autocracy_data$v2clrspct, autocracy_data$year,
                     quantile, prob = 0.75),
         SD = tapply(autocracy_data$v2clrspct, autocracy_data$year, sd),
         Skew = tapply(autocracy_data$v2clrspct, autocracy_data$year,
                       FUN = Skew),
         Kurtosis = tapply(autocracy_data$v2clrspct, autocracy_data$year,
                           FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
rigour_impartiality_full_summary_longtable <-
  pivot_longer(rigour_impartiality_full_summary, 2:8, names_to = "Statistic",
               values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = rigour_impartiality_full_summary_longtable[
    rigour_impartiality_full_summary_longtable$Statistic %in% c("Mean", "Q1",
                                                                "Median", "Q3",
                                                                "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Rig. & impart. pub. admin.")) )

#       Plot skew and kurtosis over time
ggplot(data = rigour_impartiality_full_summary_longtable[
  rigour_impartiality_full_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Rig. & impart. pub. admin.") + blue_light

#     Complete cases

#       Heatmap of political violence distribution per year
ggplot(data = datacomplete, mapping = aes(y = v2clrspct, x = year)) +
  geom_bin2d(binwidth = c(1, 0.5)) + blue_light +
  labs(y = "Rig. & impart. pub. admin.", x = "Year")

#     Calculation of country means
rigour_impartiality_complete_countrymeans <- tibble(
  countrycode = unique(datacomplete$cowcode)[order(unique(
    datacomplete$cowcode))],
  countrymean = tapply(datacomplete$v2clrspct, datacomplete$cowcode, mean) )

#       Histogram and QQ-plot of country means
ggplot(data = rigour_impartiality_complete_countrymeans, aes(x = countrymean)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue") +
  labs(x = "Rig. & impart. pub. admin.", y = "Count") + blue_light
ggplot(data = rigour_impartiality_complete_countrymeans,
       aes(sample = countrymean)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + labs(x = "Reference normal distribution",
                        y = "Rig. & impart. pub. admin.") + blue_light

#       Compute summary statistics per year
rigour_impartiality_complete_summary <-
```

```
  tibble(year =  unique(datacomplete$year)[order(unique(datacomplete$year))],
    Mean = tapply(datacomplete$v2clrspct, datacomplete$year, mean),
    Q1 = tapply(datacomplete$v2clrspct, datacomplete$year,
                quantile, prob = 0.25),
    Median = tapply(datacomplete$v2clrspct, datacomplete$year, median),
    Q3 = tapply(datacomplete$v2clrspct, datacomplete$year,
                quantile, prob = 0.75),
    SD = tapply(datacomplete$v2clrspct, datacomplete$year, sd),
    Skew = tapply(datacomplete$v2clrspct, datacomplete$year, FUN = Skew),
    Kurtosis = tapply(datacomplete$v2clrspct, datacomplete$year, FUN = Kurt) )

#         Pivot summary statistics table for use in plotting
rigour_impartiality_complete_summary_longtable <-
  pivot_longer(rigour_impartiality_complete_summary, 2:8,
               names_to = "Statistic", values_to = "Value")

#       Plot Mean, median, first quartile, third quartile
#       and standard deviation over time
withr::with_options(
  list(ggplot2.discrete.colour = lines_palette),
  print(ggplot(data = rigour_impartiality_complete_summary_longtable[
    rigour_impartiality_complete_summary_longtable$Statistic %in%
      c("Mean", "Q1", "Median", "Q3", "SD"), ],
    aes(x = year, y = Value, colour = Statistic)) + geom_line() + blue_light +
      labs(x = "Year", y = "Rig. & impart. pub. admin.")) )

#       Plot skew and kurtosis over time
ggplot(data = rigour_impartiality_complete_summary_longtable[
  rigour_impartiality_complete_summary_longtable$Statistic %in%
    c("Skew", "Kurtosis"), ], aes(x = year, y = Value, linetype = Statistic)) +
  geom_line() + labs(x = "Year", y = "Rig. & impart. pub. admin.") + blue_light

######################   Missing data analysis    ##############################

#   Remove variables that are uninformative during missing data analysis
autocracy_data_clean <- autocracy_data[, c("cowcode","gwf_country","year",
                                           "latent_personalism","regime_type",
                                           "v2x_clphy", "free_expr_x100",
                                           "e_miinteco", "e_miinterc",
                                           "e_migdppcln", "e_migdpgro",
                                           "v2caviol", "v2clrspct", "log10pop",
                                           "v2caassemb",
                                           "freedom_movement")]
datacomplete_clean <- datacomplete[, c("cowcode","gwf_country","year",
                                       "latent_personalism","regime_type",
                                       "v2x_clphy", "free_expr_x100",
                                       "e_miinteco", "e_miinterc",
                                       "e_migdppcln", "e_migdpgro",
                                       "v2caviol", "v2clrspct", "log10pop",
                                       "v2caassemb", "freedom_movement")]

# Tabulate the distribution of values over years and cowcodes
table(autocracy_data$year)
table(autocracy_data$cowcode)

#   Single out countries observed in all years
table(autocracy_data$cowcode)[table(autocracy_data$cowcode) == 65]

#   Visualise the distribution of values over years and cowcodes
```

```
ggplot(autocracy_data_clean, aes(x = year)) + geom_bar(fill = "lightblue") +
  labs(x = "Year", y = "Count") + blue_light
ggplot(autocracy_data_clean, aes(x = as.factor(cowcode))) +
  geom_bar(fill = "lightblue") + labs(x = "Country code", y = "Count") +
  guides(x = guide_axis(angle = 90)) + blue_light

# Table of missing values per variable
missing_value_table <- tibble(Variable =
                                variable.names(autocracy_data_clean)[4:16],
                              N = NA, "N%" = NA, n = NA, "n%" = NA, T_min = NA,
                              T_max = NA)

for(V in variable.names(autocracy_data_clean)[4:16]) {
  missing_value_table[missing_value_table$Variable == V, "N"] <-
    nrow(autocracy_data[complete.cases(autocracy_data[, c("cowcode", "year",
                                                          V)]), ])
  missing_value_table[missing_value_table$Variable == V, "T_min"] <-
    min(table(autocracy_data$cowcode[complete.cases(autocracy_data[, V])]))
  missing_value_table[missing_value_table$Variable == V, "T_max"] <-
    max(table(autocracy_data$cowcode[complete.cases(autocracy_data[, V])]))
  missing_value_table[missing_value_table$Variable == V, "n"] <-
    length(unique(autocracy_data$cowcode[
      complete.cases(autocracy_data[, c("cowcode", "year", V)])]))
}
missing_value_table$`N%` <- (missing_value_table$N / 4591) * 100
missing_value_table$`n%` <- (missing_value_table$n / 119) * 100

# Create a variable listing which variables a case misses
autocracy_data_clean$which_miss <- "Missing:"

for (V in variable.names(autocracy_data_clean)) {
  autocracy_data_clean[is.na(autocracy_data_clean[, V]),"which_miss"] <-
    paste(autocracy_data_clean[is.na(autocracy_data_clean[, V]),"which_miss"],
          V, sep = " ")
}

# Visualise missingness over countries and time
ggplot(autocracy_data_clean, aes(x = year, y = as.factor(cowcode),
                           fill = which_miss)) +
  labs(x = "Year", y = "Country code") +
  scale_fill_discrete(name = "Missing",
                      labels = c("None", "GDP growth",
                                 "GDP growth &\nPopulation (log 10)",
                                 "GDP growth,\nPolitical violence, \nFreedom of
assembly",
                                 "GDP per capita (ln)",
                                  "GDP per capita (ln) &\nGDP growth",
                                  "GDP per capita (ln),\nGDP growth, &\nPopulation
(log 10)",
                                  "GDP per capita (ln),\nGDP growth,\nPolitical
violence,\nFreedom of assembly",
                                  "Internal & International conflict", "GDP
growth,\nInternal, & International conflict",
                                  "GDP per capita (ln),\nGDP growth,\nInternal &
International conflict",
                                  "GDP per capita (ln),\nGDP growth,\nPopulation
(log 10),\nInternal & International conflict",
```

```
                                  "GDP per capita (ln),\nGDP growth,\nPolitical
violence,\nPopulation (log 10),\nFreedom of assembly,\nInternal & International
conflict",
                                  "Population (log 10),\nInternal & International
conflict",
                                  "Political violence,\nPopulation (log
10),\nFreedom of assembly,\nInternal & International conflict",
                                  "Population (log 10)",
                                  "Political violence",
                                  "Political violence &\nFreedom of assembly") ) +
  guides(x = guide_axis(angle = 90)) + blue_light + geom_tile()

# Compare distributions in full dataset and subset of complete cases

#   Summary statistics table

#     Create table
comparison_table <- tibble(Variable = rep(c("latent_personalism",
                                            "life_phys_x100",
                                            "free_expr_x100", "e_migdppcln",
                                            "e_migdpgro", "v2caviol",
                                            "v2clrspct", "log10pop",
                                            "freedom_movement",
                                            "v2caassemb"), each = 2),
                           Dataset = rep(c("Full", "Complete"), times = 10),
                           Mean = NA, SD = NA, Minimum = NA, Maximum = NA,
                           Skew = NA, Kurt = NA, ICC = NA)

#     Fill table
for (V in c("latent_personalism", "life_phys_x100", "free_expr_x100",
            "e_migdppcln", "e_migdpgro", "v2caviol", "v2clrspct", "log10pop",
            "freedom_movement", "v2caassemb")) {
  comparison_table[comparison_table$Variable == V &
                   comparison_table$Dataset == "Full", "Mean"] <-
    mean(tapply(autocracy_data[, V], autocracy_data$cowcode, mean,
                na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                   comparison_table$Dataset == "Complete", "Mean"] <-
    mean(tapply(datacomplete[, V], datacomplete$cowcode, mean,
                na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                   comparison_table$Dataset == "Full", "SD"] <-
    sd(tapply(autocracy_data[, V], autocracy_data$cowcode, mean,
                na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                   comparison_table$Dataset == "Complete", "SD"] <-
    sd(tapply(datacomplete[, V], datacomplete$cowcode, mean,
                na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                   comparison_table$Dataset == "Full", "Minimum"] <-
    min(tapply(autocracy_data[, V], autocracy_data$cowcode, mean,
              na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                   comparison_table$Dataset == "Complete", "Minimum"] <-
    min(tapply(datacomplete[, V], datacomplete$cowcode, mean,
              na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                   comparison_table$Dataset == "Full", "Maximum"] <-
    max(tapply(autocracy_data[, V], autocracy_data$cowcode, mean,
```

```
              na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                     comparison_table$Dataset == "Complete", "Maximum"] <-
    max(tapply(datacomplete[, V], datacomplete$cowcode, mean,
              na.rm = TRUE), na.rm = TRUE)
  comparison_table[comparison_table$Variable == V &
                     comparison_table$Dataset == "Full", "Skew"] <-
    Skew(tapply(autocracy_data[, V], autocracy_data$cowcode, mean,
                na.rm = TRUE)[is.finite(tapply(autocracy_data[, V],
                                               autocracy_data$cowcode, mean,
                                               na.rm = TRUE))])
  comparison_table[comparison_table$Variable == V &
                     comparison_table$Dataset == "Complete", "Skew"] <-
    Skew(tapply(datacomplete[, V], datacomplete$cowcode, mean, na.rm = TRUE),
         na.rm = TRUE)
#       Because some country means for GDP growth are NaNs, the GDP growth
#       would also be NaN if these are not removed. The values are removed
#       using is.finite(), but doing so distorts the estimated kurtosis.
  comparison_table[comparison_table$Variable == V &
                     comparison_table$Dataset == "Full", "Kurt"] <-
    Kurt(tapply(autocracy_data[, V], autocracy_data$cowcode, mean,
                na.rm = TRUE)[is.finite(tapply(autocracy_data[, V],
                                               autocracy_data$cowcode, mean,
                                               na.rm = TRUE))])
  comparison_table[comparison_table$Variable == V &
                     comparison_table$Dataset == "Complete", "Kurt"] <-
    Kurt(tapply(datacomplete[, V], datacomplete$cowcode, mean,
                na.rm = TRUE), na.rm = TRUE)
  full_intercept_model <- lmer(autocracy_data[, V] ~ (1|cowcode),
                               data = autocracy_data)
  full_intercept_model_ICC_frame <-
    as.data.frame(VarCorr(full_intercept_model))
  comparison_table[comparison_table$Variable == V &
                     comparison_table$Dataset == "Full", "ICC"] <-
    full_intercept_model_ICC_frame[1, 4] /
    (full_intercept_model_ICC_frame[1, 4] +
       full_intercept_model_ICC_frame[2, 4])
  complete_intercept_model <- lmer(datacomplete[, V] ~ (1|cowcode),
                               data = datacomplete)
  complete_intercept_model_ICC_frame <-
    as.data.frame(VarCorr(complete_intercept_model))
  comparison_table[comparison_table$Variable == V &
                     comparison_table$Dataset == "Complete", "ICC"] <-
    complete_intercept_model_ICC_frame[1, 4] /
    (complete_intercept_model_ICC_frame[1, 4] +
       complete_intercept_model_ICC_frame[2, 4])
}

##########################  Bivariate analyses  ################################

# Collect over-time country means
countrymean_table <-
  tibble(cowcode = latent_personalism_complete_countrymeans$countrycode,
         latent_personalism = latent_personalism_complete_countrymeans$
           countrymean,
         freedom_assembly = free_assemb_complete_countrymeans$countrymean,
         freedom_move = free_move_complete_countrymeans$countrymean,
         protection_life_phys = complete_life_phys_countrymeans$countrymean,
         freedom_expression = free_expr_complete_countrymeans$countrymean,
```

```
         log_population = log_population_complete_countrymeans$countrymean,
         log_GDP_per_cap = GDP_cap_complete_countrymeans$countrymean,
         GDP_growth = GDP_growth_complete_countrymeans$countrymean,
         rigour_impartiality = rigour_impartiality_complete_countrymeans$
           countrymean,
         political_violence = political_violence_complete_countrymeans$
           countrymean,
         personal_regime = tapply(datacomplete$gwf_personal,
                                  datacomplete$cowcode, mean),
         party_regime = tapply(datacomplete$gwf_party,
                               datacomplete$cowcode, mean),
         military_regime = tapply(datacomplete$gwf_military,
                               datacomplete$cowcode, mean),
         monarch_regime = tapply(datacomplete$gwf_monarch,
                               datacomplete$cowcode, mean),
         international_conflict = tapply(datacomplete$e_miinteco,
                                         datacomplete$cowcode, mean),
         internal_conflict = tapply(datacomplete$e_miinterc,
                                    datacomplete$cowcode, mean) )

# Calculate correlations between over-time country means
cor(countrymean_table[, 2:17])

#   establish significance threshold
a <- exp((2*1.96)/sqrt(109-3))
(a - 1) / (a + 1)

# Calculate intra-class correlations with year as class

#   Personalism
personalism_ICC_model <- lmer(latent_personalism ~ (1|cowcode),
                             data = datacomplete)
personalism_ICC_frame <-
  as.data.frame(VarCorr(personalism_ICC_model))
personalism_ICC_frame[1, 4] / (personalism_ICC_frame[1, 4] +
                                 personalism_ICC_frame[2, 4])

#   Freedom of movement
free_move_ICC_model <- lmer(freedom_movement ~ (1|cowcode),
                              data = datacomplete)
free_move_ICC_frame <-
  as.data.frame(VarCorr(free_move_ICC_model))
free_move_ICC_frame[1, 4] / (free_move_ICC_frame[1, 4] +
                               free_move_ICC_frame[2, 4])

#   Freedom of assembly
free_assemb_ICC_model <- lmer(v2caassemb ~ (1|cowcode), data = datacomplete)
free_assemb_ICC_frame <- as.data.frame(VarCorr(free_assemb_ICC_model))
free_assemb_ICC_frame[1, 4] / (free_assemb_ICC_frame[1, 4] +
                                 free_assemb_ICC_frame[2, 4])

#   Freedom of expression
free_expr_ICC_model <- lmer(free_expr_x100 ~ (1|cowcode), data = datacomplete)
free_expr_ICC_frame <- as.data.frame(VarCorr(free_expr_ICC_model))
free_expr_ICC_frame[1, 4] / (free_expr_ICC_frame[1, 4] +
                               free_expr_ICC_frame[2, 4])

#   Protection of life and physical integrity
life_phys_ICC_model <- lmer(life_phys_x100 ~ (1|cowcode), data = datacomplete)
```

```
life_phys_ICC_frame <- as.data.frame(VarCorr(life_phys_ICC_model))
life_phys_ICC_frame[1, 4] / (life_phys_ICC_frame[1, 4] +
                               life_phys_ICC_frame[2, 4])

#   Political violence
political_violence_ICC_model <- lmer(v2caviol ~ (1|cowcode),
                                     data = datacomplete)
political_violence_ICC_frame <-
  as.data.frame(VarCorr(political_violence_ICC_model))
political_violence_ICC_frame[1, 4] / (political_violence_ICC_frame[1, 4] +
                                        political_violence_ICC_frame[2, 4])

#   GDP per capita
GDP_cap_ICC_model <- lmer(e_migdppcln ~ (1|cowcode), data = datacomplete)
GDP_cap_ICC_frame <- as.data.frame(VarCorr(GDP_cap_ICC_model))
GDP_cap_ICC_frame[1, 4] / (GDP_cap_ICC_frame[1, 4] + GDP_cap_ICC_frame[2, 4])

#   GDP growth
GDP_growth_ICC_model <- lmer(e_migdpgro ~ (1|cowcode), data = datacomplete)
GDP_growth_ICC_frame <- as.data.frame(VarCorr(GDP_growth_ICC_model))
GDP_growth_ICC_frame[1, 4] / (GDP_growth_ICC_frame[1, 4] +
                                GDP_growth_ICC_frame[2, 4])

#   Rigour and impartiality of the public administration
rig_impart_ICC_model <- lmer(v2clrspct ~ (1|cowcode), data = datacomplete)
rig_impart_ICC_frame <- as.data.frame(VarCorr(rig_impart_ICC_model))
rig_impart_ICC_frame[1, 4] / (rig_impart_ICC_frame[1, 4] +
                                rig_impart_ICC_frame[2, 4])

#   Population (log 10)
population_ICC_model <- lmer(log10pop ~ (1|cowcode), data = datacomplete)
population_ICC_frame <- as.data.frame(VarCorr(population_ICC_model))
population_ICC_frame[1, 4] / (population_ICC_frame[1, 4] +
                                population_ICC_frame[2, 4])

################################################################################
############################  REGRESSION MODELLING  ############################
################################################################################

############################## Freedom of expression ###########################

#   Intercept-only model
free_expr_intercept <- lmer(free_expr_x100 ~ (1|cowcode), data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_expr_intercept, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_expr_intercept, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_expr_intercept_ICC_frame <- as.data.frame(VarCorr(free_expr_intercept))
free_expr_intercept_ICC_frame

#     Calculate deviance
deviance_free_expr_intercept <- deviance(lmer(free_expr_x100 ~ (1|cowcode),
                                              data = datacomplete,
                                              REML = FALSE))
```

```
#       ICC
free_expr_intercept_ICC_frame[1, 4] / (free_expr_intercept_ICC_frame[1, 4] +
                                         free_expr_intercept_ICC_frame[2, 4])

#   Model with year-fixed effects
free_expr_yearfixed <- lmer(free_expr_x100 ~ (1|cowcode) + as.factor(year),
                            data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_expr_yearfixed, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_expr_yearfixed, vcov = "CR2", level = 0.9875,
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_expr_yearfixed_ICC_frame <- as.data.frame(VarCorr(free_expr_yearfixed))
free_expr_yearfixed_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_expr_yearfixed_ICC_frame[,4])/
  sum(free_expr_intercept_ICC_frame[,4])

#       ICC
free_expr_yearfixed_ICC_frame[1, 4] / (free_expr_yearfixed_ICC_frame[1, 4] +
                                         free_expr_yearfixed_ICC_frame[2, 4])

#     LR-test

#       LR-test against intercept-only model
deviance_free_expr_yearfixed <- deviance(lmer(free_expr_x100 ~ (1|cowcode) +
                                                as.factor(year),
                                              data = datacomplete,
                                              REML = FALSE))
deviance_free_expr_intercept - deviance_free_expr_yearfixed
pchisq(deviance_free_expr_intercept - deviance_free_expr_yearfixed,
       df = 54, lower.tail = FALSE)

#   Model with personalism
free_expr_personalism <- lmer(free_expr_x100 ~ (1|cowcode) + as.factor(year) +
                                latent_personalism, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_expr_personalism, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_expr_personalism, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_expr_personalism_ICC_frame <- as.data.frame(VarCorr(free_expr_personalism))
free_expr_personalism_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_expr_personalism_ICC_frame[,4])/
  sum(free_expr_intercept_ICC_frame[,4])

#       ICC
```

```
free_expr_personalism_ICC_frame[1, 4] /
  (free_expr_personalism_ICC_frame[1, 4] +
     free_expr_personalism_ICC_frame[2, 4])

#     Removing year-fixed effects
free_expr_personalism_no_yearfixed <- lmer(free_expr_x100 ~ (1|cowcode) +
                                             latent_personalism,
                                           data = datacomplete)
coef_test(free_expr_personalism_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-tests

#       LR-test against model with year-fixed effects
deviance_free_expr_personalism <- deviance(lmer(free_expr_x100 ~ (1|cowcode) +
                                                  as.factor(year) +
                                                  latent_personalism,
                                                data = datacomplete,
                                                REML = FALSE))

deviance_free_expr_yearfixed - deviance_free_expr_personalism
pchisq(deviance_free_expr_yearfixed - deviance_free_expr_personalism,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_expr_personalism_no_yearfixed <-
  deviance(lmer(free_expr_x100 ~ (1|cowcode) + latent_personalism,
                data = datacomplete, REML = FALSE))
deviance_free_expr_personalism_no_yearfixed - deviance_free_expr_personalism
pchisq(deviance_free_expr_personalism_no_yearfixed -
         deviance_free_expr_personalism, df = 54, lower.tail = FALSE)

#   Model with personalism and its square
free_expr_personalism_squared <- lmer(free_expr_x100 ~ (1|cowcode) +
                                        as.factor(year) +
                                        poly(latent_personalism, 2,
                                             raw = TRUE), data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_expr_personalism_squared, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_expr_personalism_squared, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_expr_personalism_squared_ICC_frame <-
  as.data.frame(VarCorr(free_expr_personalism_squared))
free_expr_personalism_squared_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_expr_personalism_squared_ICC_frame[,4])/
  sum(free_expr_intercept_ICC_frame[,4])

#       ICC
free_expr_personalism_squared_ICC_frame[1, 4] /
  (free_expr_personalism_squared_ICC_frame[1, 4] +
     free_expr_personalism_squared_ICC_frame[2, 4])
```

```
#     LR-test

#       LR-test against personalism-only model
deviance_free_expr_personalism_squared <-
  deviance(lmer(free_expr_x100 ~ (1|cowcode) + as.factor(year) +
                  poly(latent_personalism, 2, raw = TRUE), data = datacomplete,
                REML = FALSE))
deviance_free_expr_personalism - deviance_free_expr_personalism_squared
pchisq(deviance_free_expr_personalism - deviance_free_expr_personalism_squared,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_expr_personalism_squared_no_yearfixed <-
  deviance(lmer(free_expr_x100 ~ (1|cowcode) + poly(latent_personalism, 2,
                                                     raw = TRUE),
                data = datacomplete, REML = FALSE))
deviance_free_expr_personalism_squared_no_yearfixed -
  deviance_free_expr_personalism_squared
pchisq(deviance_free_expr_personalism_squared_no_yearfixed -
         deviance_free_expr_personalism_squared, df = 54, lower.tail = FALSE)

#     Compare level 1 residuals with linear and quadratic curves

#       Quadratic model
ggplot(datacomplete, aes(x = latent_personalism,
                         y = resid(free_expr_personalism_squared))) +
  geom_point(colour = "darkblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Personalism", y = "Level 1 residual")

#       Linear model
ggplot(datacomplete, aes(x = latent_personalism,
                         y = resid(free_expr_personalism))) +
  geom_point(colour = "darkblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Personalism", y = "Level 1 residual")

#   Model with personalism and controls
free_expr_control <- lmer(free_expr_x100 ~ (1|cowcode) + as.factor(year) +
                            latent_personalism + lag_e_miinteco +
                            lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
                            lag_v2caviol + lag_log10pop + gwf_monarch +
                            gwf_military + gwf_party, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_expr_control, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_expr_control, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_expr_control_ICC_frame <- as.data.frame(VarCorr(free_expr_control))
free_expr_control_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_expr_control_ICC_frame[,4])/ sum(free_expr_intercept_ICC_frame[,4])
```

```
#       ICC
free_expr_control_ICC_frame[1, 4] / (free_expr_control_ICC_frame[1, 4] +
                                       free_expr_control_ICC_frame[2, 4])

#     Removing year-fixed effects
free_expr_control_no_yearfixed <- lmer(free_expr_x100 ~ (1|cowcode) +
                                         latent_personalism +
                                         lag_e_miinteco + lag_e_miinterc +
                                         lag_e_migdppcln + lag_e_migdpgro +
                                         lag_v2caviol + lag_log10pop +
                                         gwf_monarch + gwf_military + gwf_party,
                                       data = datacomplete)
coef_test(free_expr_control_no_yearfixed, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against personalism-only model
deviance_free_expr_control <-
  deviance(lmer(free_expr_x100 ~ (1|cowcode) + as.factor(year) +
                  latent_personalism + lag_e_miinteco + lag_e_miinterc +
                  lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
                  lag_log10pop + gwf_monarch + gwf_military + gwf_party,
                data = datacomplete, REML = FALSE))
deviance_free_expr_personalism - deviance_free_expr_control
pchisq(deviance_free_expr_personalism - deviance_free_expr_control,
       df = 9, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_expr_control_no_yearfixed <-
  deviance(lmer(free_expr_x100 ~ (1|cowcode) + latent_personalism +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_free_expr_control_no_yearfixed - deviance_free_expr_control
pchisq(deviance_free_expr_control_no_yearfixed - deviance_free_expr_control,
       df = 54, lower.tail = FALSE)

#     Residual diagnostics

#       Add predicted values to dataset
datacomplete$free_expr_predict <- predict(free_expr_control)

#       Add residuals to dataset

#         Calculate level 1 residuals
datacomplete$free_expr_resid_level_1 <- residuals(free_expr_control)

#         Standardise level 1 residuals
for(c in unique(datacomplete$cowcode)) {
  datacomplete$free_expr_country_resid_sds[datacomplete$cowcode == c] <-
    sd(datacomplete$free_expr_resid_level_1[datacomplete$cowcode == c])
}
datacomplete$free_expr_stand_resid_level_1 <-
  datacomplete$free_expr_resid_level_1 /
  datacomplete$free_expr_country_resid_sds

#       Calculate level 2 residuals
```

```
free_expr_resid_level_2 <- as.data.frame(ranef(free_expr_control))
free_expr_resid_level_2$cowcode <-
  as.numeric(levels(free_expr_resid_level_2$grp))[free_expr_resid_level_2$grp]

#       Standardise level 2 residuals
free_expr_resid_level_2$free_expr_stand_resid_level_2 <-
  free_expr_resid_level_2$condval /
  free_expr_control_ICC_frame$sdcor[1]

#       Add level 2 residuals to dataset
datacomplete <- left_join(datacomplete,
            free_expr_resid_level_2[, c("cowcode", "condval",
                                        "free_expr_stand_resid_level_2")],
            by = c("cowcode"))
datacomplete <- rename(datacomplete, free_expr_resid_level_2 = condval)

#       Autocorrelation

#         Create dataframe with just the residuals
free_expr_resid_pdata <-
  pdata.frame(datacomplete[, c("year", "cowcode", "free_expr_resid_level_1")],
              index = c("cowcode", "year"))
#         Create lagged residuals
for (j in 1:20) {
  # Create the name of the lagged variable
  var <- paste("free_expr_lag", j ,"_resid_level_1", sep = "")
  # Generate the lagged variable via plm's implementation of "lag"
  free_expr_resid_pdata[, var] <-
    plm::lag(free_expr_resid_pdata$free_expr_resid_level_1, k= j)
}

#         Calculate correlations between present values and lags
autocors_free_expr_resid_level_1 <-
  cor(free_expr_resid_pdata$free_expr_resid_level_1,
      free_expr_resid_pdata[, 4:ncol(free_expr_resid_pdata)],
      use = "complete.obs")

#         Reschape the correlations into a more workable format
autocors_free_expr_resid_level_1 <-
  tibble(Lag = 1:20, Correlation = t(autocors_free_expr_resid_level_1))

#         Create an autocorrelation plot
ggplot(data = autocors_free_expr_resid_level_1, aes(x = Lag, y = Correlation)) +
  geom_col(fill = "lightblue") + blue_light

#       Linearity

#         Plot predicted values against level 1 residuals
ggplot(datacomplete, aes(x = free_expr_predict,
                         y = free_expr_stand_resid_level_1)) +
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Predicted value", y = "Level 1 residual (standardised)")

#       Plot predicted values from fixed effects against level 2 residuals
ggplot(datacomplete, aes(x = free_expr_predict - free_expr_resid_level_2,
                         y = free_expr_stand_resid_level_2)) +
  geom_point(colour = "lightblue") +
```

```
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Fixed part predictions", y = "Level 2 residual (standardised)")

#       Normality

#         Assign residual levels to the residuals for combination
rbind_resids_level_2 <-
  rename(free_expr_resid_level_2[, c("cowcode",
                                     "free_expr_stand_resid_level_2")],
         resid = free_expr_stand_resid_level_2)
rbind_resids_level_2$level <- "Level 2"
rbind_resids_level_1 <-
  rename(datacomplete[, c("cowcode", "free_expr_stand_resid_level_1")],
         resid = free_expr_stand_resid_level_1)
rbind_resids_level_1$level <- "Level 1"

#         Create long dataset of all residuals
free_expr_resids <- rbind(rbind_resids_level_1, rbind_resids_level_2)
rm(list = c("rbind_resids_level_1", "rbind_resids_level_2"))

#         Plot residuals in dual QQ-plot
withr::with_options(
  list(ggplot2.discrete.colour = c("#E69F00", "#56B4E9", "#009E73", "#F0E442",
                                   "#0072B2", "#D55E00", "#CC79A7")),
  print(ggplot(data = free_expr_resids, aes(sample = resid,
                                            colour = as.factor(level))) +
          geom_qq() + geom_qq_line() + blue_light +
          scale_colour_discrete(name = "Residual level") +
          labs(y = "Standardised residuals",
               x = "Reference normal distribution")) )

#         Plot residuals in single QQ-plots

#             Level 1
ggplot(data = free_expr_resids[free_expr_resids$level == "Level 1", ],
       aes(sample = resid)) + geom_qq(colour = "lightblue") + geom_qq_line() +
  blue_light + labs(y = "Standardised residuals",
                  x = "Reference normal distribution")

#             Level 2
ggplot(data = free_expr_resids[free_expr_resids$level == "Level 2", ],
       aes(sample = resid)) + geom_qq(colour = "lightblue") + geom_qq_line() +
  blue_light + labs(y = "Standardised residuals", x = "Reference normal
distribution")

#   Model with rigour and impartiality of the public administration
free_expr_rig_impart <- lmer(free_expr_x100 ~ (1|cowcode) + as.factor(year) +
                               latent_personalism + v2clrspct + lag_e_miinteco +
                               lag_e_miinterc + lag_e_migdppcln +
                               lag_e_migdpgro + lag_v2caviol + lag_log10pop +
                               gwf_monarch + gwf_military + gwf_party,
                               data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_expr_rig_impart, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_expr_rig_impart, vcov = "CR2", level = 0.9875,
```

```
        test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_expr_rig_impart_ICC_frame <- as.data.frame(VarCorr(free_expr_rig_impart))
free_expr_rig_impart_ICC_frame

#     Removing year-fixed effects
free_expr_rig_impart_no_yearfixed <- lmer(free_expr_x100 ~ (1|cowcode) +
                                            latent_personalism + v2clrspct +
                                            lag_e_miinteco + lag_e_miinterc +
                                            lag_e_migdppcln + lag_e_migdpgro +
                                            lag_v2caviol + lag_log10pop +
                                            gwf_monarch + gwf_military +
                                            gwf_party, data = datacomplete)
coef_test(free_expr_rig_impart_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test and ICC

#       LR-test against personalism-only model
deviance_free_expr_rig_impart <- deviance(lmer(free_expr_x100 ~ (1|cowcode) +
                                                 as.factor(year) +
                                                 latent_personalism +
                                                 v2clrspct + lag_e_miinteco +
                                                 lag_e_miinterc +
                                                 lag_e_migdppcln +
                                                 lag_e_migdpgro +
                                                 lag_v2caviol + lag_log10pop +
                                                 gwf_monarch + gwf_military +
                                                 gwf_party,
                                               data = datacomplete, REML = FALSE))
deviance_free_expr_control - deviance_free_expr_rig_impart
pchisq(deviance_free_expr_control - deviance_free_expr_rig_impart,
       df = 9, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_expr_rig_impart_no_yearfixed <-
  deviance(lmer(free_expr_x100 ~ (1|cowcode) + latent_personalism + v2clrspct +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_free_expr_rig_impart_no_yearfixed - deviance_free_expr_rig_impart
pchisq(deviance_free_expr_rig_impart_no_yearfixed -
         deviance_free_expr_rig_impart, df = 54, lower.tail = FALSE)

#       ICC
free_expr_rig_impart_ICC_frame[1, 4] / (free_expr_rig_impart_ICC_frame[1, 4] +
                                          free_expr_rig_impart_ICC_frame[2, 4])

############################## Freedom of assembly ############################

#   Intercept-only model
free_assemb_intercept <- lmer(v2caassemb ~ (1|cowcode), data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_assemb_intercept, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_assemb_intercept, vcov = "CR2", level = 0.9875,
```

```
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_assemb_intercept_ICC_frame <- as.data.frame(VarCorr(free_assemb_intercept))
free_assemb_intercept_ICC_frame

#     Calculate deviance
deviance_free_assemb_intercept <- deviance(lmer(v2caassemb ~ (1|cowcode),
                                                data = datacomplete,
                                                REML = FALSE))

#       ICC
free_assemb_intercept_ICC_frame[1, 4] /
  (free_assemb_intercept_ICC_frame[1, 4] +
     free_assemb_intercept_ICC_frame[2, 4])

#   Model with year-fixed effects
free_assemb_yearfixed <- lmer(v2caassemb ~ (1|cowcode) + as.factor(year),
                              data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_assemb_yearfixed, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_assemb_yearfixed, vcov = "CR2", level = 0.9875,
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_assemb_yearfixed_ICC_frame <- as.data.frame(VarCorr(free_assemb_yearfixed))
free_assemb_yearfixed_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_assemb_yearfixed_ICC_frame[,4])/
  sum(free_assemb_intercept_ICC_frame[,4])

#       ICC
free_assemb_yearfixed_ICC_frame[1, 4] /
  (free_assemb_yearfixed_ICC_frame[1, 4] +
     free_assemb_yearfixed_ICC_frame[2, 4])

#     LR-test

#       LR-test against intercept-only model
deviance_free_assemb_yearfixed <- deviance(lmer(v2caassemb ~ (1|cowcode) +
                                                  as.factor(year),
                                                data = datacomplete,
                                                REML = FALSE))
deviance_free_assemb_intercept - deviance_free_assemb_yearfixed
pchisq(deviance_free_assemb_intercept - deviance_free_assemb_yearfixed,
       df = 54, lower.tail = FALSE)

#   Model with personalism
free_assemb_personalism <- lmer(v2caassemb ~ (1|cowcode) + as.factor(year) +
                                  latent_personalism, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_assemb_personalism, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
```

```
conf_int(free_assemb_personalism, vcov = "CR2", level = 0.9875,
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_assemb_personalism_ICC_frame <-
  as.data.frame(VarCorr(free_assemb_personalism))
free_assemb_personalism_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_assemb_personalism_ICC_frame[,4])/
  sum(free_assemb_intercept_ICC_frame[,4])

#       ICC
free_assemb_personalism_ICC_frame[1, 4] /
  (free_assemb_personalism_ICC_frame[1, 4] +
     free_assemb_personalism_ICC_frame[2, 4])

#     Removing year-fixed effects
free_assemb_rig_personalism_no_yearfixed <- lmer(v2caassemb ~ (1|cowcode) +
                                                   latent_personalism,
                                                 data = datacomplete)
coef_test(free_assemb_rig_personalism_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against model with year-fixed effects
deviance_free_assemb_personalism <- deviance(lmer(v2caassemb ~ (1|cowcode) +
                                                    as.factor(year) +
                                                    latent_personalism,
                                                  data = datacomplete,
                                                  REML = FALSE))

deviance_free_assemb_yearfixed - deviance_free_assemb_personalism
pchisq(deviance_free_assemb_yearfixed - deviance_free_assemb_personalism,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_assemb_personalism_no_yearfixed <-
  deviance(lmer(v2caassemb ~ (1|cowcode) + latent_personalism,
                data = datacomplete, REML = FALSE))
deviance_free_assemb_personalism_no_yearfixed - deviance_free_assemb_personalism
pchisq(deviance_free_assemb_personalism_no_yearfixed -
         deviance_free_assemb_personalism,
       df = 54, lower.tail = FALSE)

#   Model with personalism and controls
free_assemb_control <- lmer(v2caassemb ~ (1|cowcode) + as.factor(year) +
                              latent_personalism + lag_e_miinteco +
                              lag_e_miinterc + lag_e_migdppcln +
                              lag_e_migdpgro + lag_v2caviol + lag_log10pop +
                              gwf_monarch + gwf_military + gwf_party,
                            data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_assemb_control, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_assemb_control, vcov = "CR2", level = 0.9875,
```

```
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_assemb_control_ICC_frame <- as.data.frame(VarCorr(free_assemb_control))
free_assemb_control_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_assemb_control_ICC_frame[,4])/
  sum(free_assemb_intercept_ICC_frame[,4])

#       ICC
free_assemb_control_ICC_frame[1, 4] / (free_assemb_control_ICC_frame[1, 4] +
                                         free_assemb_control_ICC_frame[2, 4])

#     Removing year-fixed effects
free_assemb_control_no_yearfixed <-
  lmer(v2caassemb ~ (1|cowcode) + latent_personalism + lag_e_miinteco +
         lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
         lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)
coef_test(free_assemb_control_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against personalism-only model
deviance_free_assemb_control <-
  deviance(lmer(v2caassemb ~ (1|cowcode) + as.factor(year) +
                  latent_personalism + lag_e_miinteco + lag_e_miinterc +
                  lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
                  lag_log10pop + gwf_monarch + gwf_military + gwf_party,
                data = datacomplete, REML = FALSE))
deviance_free_assemb_personalism - deviance_free_assemb_control
pchisq(deviance_free_assemb_personalism - deviance_free_assemb_control,
       df = 9, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_assemb_control_no_yearfixed <-
  deviance(lmer(v2caassemb ~ (1|cowcode) + latent_personalism +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_free_assemb_control_no_yearfixed - deviance_free_assemb_control
pchisq(deviance_free_assemb_control_no_yearfixed - deviance_free_assemb_control,
       df = 54, lower.tail = FALSE)

#     Residual diagnostics

#       Add predicted values to dataset
datacomplete$free_assemb_predict <- predict(free_assemb_control)

#       Add residuals to dataset

#         Calculate level 1 residuals
datacomplete$free_assemb_resid_level_1 <- residuals(free_assemb_control)

#         Standardise level 1 residuals
for(c in unique(datacomplete$cowcode)) {
  datacomplete$free_assemb_country_resid_sds[datacomplete$cowcode == c] <-
```

```
    sd(datacomplete$free_assemb_resid_level_1[datacomplete$cowcode == c])
}
datacomplete$free_assemb_stand_resid_level_1 <-
  datacomplete$free_assemb_resid_level_1 /
  datacomplete$free_assemb_country_resid_sds

#         Calculate level 2 residuals
free_assemb_resid_level_2 <- as.data.frame(ranef(free_assemb_control))
free_assemb_resid_level_2$cowcode <-
  as.numeric(levels(free_assemb_resid_level_2$grp))[free_assemb_resid_level_2$
                                                      grp]

#         Standardise level 2 residuals
free_assemb_resid_level_2$free_assemb_stand_resid_level_2 <-
  free_assemb_resid_level_2$condval / free_assemb_control_ICC_frame$sdcor[1]

#         Add level 2 residuals to dataset
datacomplete <-
  left_join(datacomplete,
            free_assemb_resid_level_2[, c("cowcode", "condval",
                                          "free_assemb_stand_resid_level_2")],
            by = c("cowcode"))
datacomplete <- rename(datacomplete, free_assemb_resid_level_2 = condval)

#       Autocorrelation

#         Create dataframe with just the residuals
free_assemb_resid_pdata <-
  pdata.frame(datacomplete[, c("year", "cowcode", "free_assemb_resid_level_1")],
              index = c("cowcode", "year"))
#         Create lagged residuals
for (j in 1:20) {
  # Create the name of the lagged variable
  var <- paste("free_assemb_lag", j ,"_resid_level_1", sep = "")
  # Generate the lagged variable via plm's implementation of "lag"
  free_assemb_resid_pdata[, var] <-
    plm::lag(free_assemb_resid_pdata$free_assemb_resid_level_1, k= j)
}

#         Calculate correlations between present values and lags
autocors_free_assemb_resid_level_1 <-
  cor(free_assemb_resid_pdata$free_assemb_resid_level_1,
      free_assemb_resid_pdata[, 4:ncol(free_assemb_resid_pdata)],
      use = "complete.obs")

#         Reschape the correlations into a more workable format
autocors_free_assemb_resid_level_1 <-
  tibble(Lag = 1:20, Correlation = t(autocors_free_assemb_resid_level_1))

#         Create an autocorrelation plot
ggplot(data = autocors_free_assemb_resid_level_1,
       aes(x = Lag, y = Correlation)) + geom_col(fill = "lightblue") +
  blue_light

#       Linearity

#         Plot predicted values against level 1 residuals
ggplot(datacomplete, aes(x = free_assemb_predict,
                         y = free_assemb_stand_resid_level_1)) +
```

```
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) +
  blue_light + labs(x = "Predicted value",
                    y = "Level 1 residual (standardised)")

#       Plot values predicted by fixed part against level 2 residuals
ggplot(datacomplete, aes(x = free_assemb_predict - free_assemb_resid_level_2,
                         y = free_assemb_stand_resid_level_2)) +
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) +
  blue_light + labs(x = "Fixed part predictions",
                    y = "Level 2 residual (standardised)")

#       Normality

#         Assign residual levels to the residuals for combination
rbind_resids_level_2 <-
  rename(free_assemb_resid_level_2[, c("cowcode",
                                       "free_assemb_stand_resid_level_2")],
         resid = free_assemb_stand_resid_level_2)
rbind_resids_level_2$level <- "Level 2"
rbind_resids_level_1 <-
  rename(datacomplete[, c("cowcode", "free_assemb_stand_resid_level_1")],
         resid = free_assemb_stand_resid_level_1)
rbind_resids_level_1$level <- "Level 1"

#         Create long dataset of all residuals
free_assemb_resids <- rbind(rbind_resids_level_1, rbind_resids_level_2)
rm(list = c("rbind_resids_level_1", "rbind_resids_level_2"))

#         Plot residuals in dual QQ-plot
withr::with_options(
  list(ggplot2.discrete.colour = c("#E69F00", "#56B4E9", "#009E73", "#F0E442",
                                   "#0072B2", "#D55E00", "#CC79A7")),
  print(ggplot(data = free_assemb_resids, aes(sample = resid,
                                              colour = as.factor(level))) +
          geom_qq() + geom_qq_line() + blue_light +
          scale_colour_discrete(name = "Residual level") +
          labs(y = "Standardised residuals",
               x = "Reference normal distribution"))
)

#         Plot residuals in single QQ-plots

#             Level 1
ggplot(data = free_assemb_resids[free_assemb_resids$level == "Level 1", ],
       aes(sample = resid)) + geom_qq(colour = "lightblue") + geom_qq_line() +
  blue_light + labs(y = "Standardised residuals",
                    x = "Reference normal distribution")

#             Level 2
ggplot(data = free_assemb_resids[free_assemb_resids$level == "Level 2", ],
       aes(sample = resid)) + geom_qq(colour = "lightblue") + geom_qq_line() +
  blue_light + labs(y = "Standardised residuals",
                    x = "Reference normal distribution")

#   Model with rigour and impartiality of the public administration
```

```
free_assemb_rig_impart <-
  lmer(v2caassemb ~ (1|cowcode) + as.factor(year) + latent_personalism +
         v2clrspct + lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
         lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
         gwf_military + gwf_party, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_assemb_rig_impart, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_assemb_rig_impart, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_assemb_rig_impart_ICC_frame <-
  as.data.frame(VarCorr(free_assemb_rig_impart))
free_assemb_rig_impart_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_assemb_rig_impart_ICC_frame[,4])/
  sum(free_assemb_intercept_ICC_frame[,4])

#       ICC
free_assemb_rig_impart_ICC_frame[1, 4] /
  (free_assemb_rig_impart_ICC_frame[1, 4] +
     free_assemb_rig_impart_ICC_frame[2, 4])

#     Removing year-fixed effects
free_assemb_rig_impart_no_yearfixed <-
  lmer(v2caassemb ~ (1|cowcode) + latent_personalism + v2clrspct +
         lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
         lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)
coef_test(free_assemb_rig_impart_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against model with controls
deviance_free_assemb_rig_impart <-
  deviance(lmer(v2caassemb ~ (1|cowcode) + as.factor(year) +
                  latent_personalism + v2clrspct + lag_e_miinteco +
                  lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
                  lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military +
                  gwf_party, data = datacomplete, REML = FALSE))
deviance_free_assemb_control - deviance_free_assemb_rig_impart
pchisq(deviance_free_assemb_control - deviance_free_assemb_rig_impart,
       df = 9, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_assemb_rig_impart_no_yearfixed <-
  deviance(lmer(v2caassemb ~ (1|cowcode) + latent_personalism + v2clrspct +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_free_assemb_rig_impart_no_yearfixed - deviance_free_assemb_rig_impart
pchisq(deviance_free_assemb_rig_impart_no_yearfixed -
         deviance_free_assemb_rig_impart, df = 54, lower.tail = FALSE)
```

```
################## Protection of life and physical integrity ###################

#   Intercept-only model
life_phys_intercept <- lmer(life_phys_x100 ~ (1|cowcode), data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(life_phys_intercept, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(life_phys_intercept, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
life_phys_intercept_ICC_frame <- as.data.frame(VarCorr(life_phys_intercept))
life_phys_intercept_ICC_frame

#     Calculate deviance
deviance_life_phys_intercept <- deviance(lmer(life_phys_x100 ~ (1|cowcode),
                                              data = datacomplete,
                                              REML = FALSE))

#       ICC
life_phys_intercept_ICC_frame[1, 4] / (life_phys_intercept_ICC_frame[1, 4] +
                                         life_phys_intercept_ICC_frame[2, 4])

#   Model with year-fixed effects
life_phys_yearfixed <- lmer(life_phys_x100 ~ (1|cowcode) + as.factor(year),
                            data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(life_phys_yearfixed, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(life_phys_yearfixed, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition and REML criterion
life_phys_yearfixed_ICC_frame <- as.data.frame(VarCorr(life_phys_yearfixed))
life_phys_yearfixed_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(life_phys_yearfixed_ICC_frame[,4])/
  sum(life_phys_intercept_ICC_frame[,4])

#       ICC
life_phys_yearfixed_ICC_frame[1, 4] / (life_phys_yearfixed_ICC_frame[1, 4] +
                                         life_phys_yearfixed_ICC_frame[2, 4])

#     LR-test and ICC

#       LR-test against intercept-only model
deviance_life_phys_yearfixed <- deviance(lmer(life_phys_x100 ~ (1|cowcode) +
                                                as.factor(year),
                                              data = datacomplete,
                                              REML = FALSE))
deviance_life_phys_intercept - deviance_life_phys_yearfixed
pchisq(deviance_life_phys_intercept - deviance_life_phys_yearfixed,
       df = 54, lower.tail = FALSE)
```

```
#   Model with personalism
life_phys_personalism <- lmer(life_phys_x100 ~ (1|cowcode) + as.factor(year) +
                                latent_personalism, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(life_phys_personalism, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(life_phys_personalism, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
life_phys_personalism_ICC_frame <- as.data.frame(VarCorr(life_phys_personalism))
life_phys_personalism_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(life_phys_personalism_ICC_frame[,4])/
  sum(life_phys_intercept_ICC_frame[,4])

#       ICC
life_phys_personalism_ICC_frame[1, 4] /
  (life_phys_personalism_ICC_frame[1, 4] +
     life_phys_personalism_ICC_frame[2, 4])

#     Removing year-fixed effects
life_phys_rig_personalism_no_yearfixed <-
  lmer(life_phys_x100 ~ (1|cowcode) + latent_personalism, data = datacomplete)
coef_test(life_phys_rig_personalism_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against model with year-fixed effects
deviance_life_phys_personalism <-
  deviance(lmer(life_phys_x100 ~ (1|cowcode) + as.factor(year) +
                  latent_personalism, data = datacomplete, REML = FALSE))

deviance_life_phys_yearfixed - deviance_life_phys_personalism
pchisq(deviance_life_phys_yearfixed - deviance_life_phys_personalism,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_life_phys_personalism_no_yearfixed <-
  deviance(lmer(life_phys_x100 ~ (1|cowcode) + latent_personalism,
                data = datacomplete, REML = FALSE))
deviance_life_phys_personalism_no_yearfixed - deviance_life_phys_personalism
pchisq(deviance_life_phys_personalism_no_yearfixed -
         deviance_life_phys_personalism, df = 54, lower.tail = FALSE)

#   Model with personalism and controls
life_phys_control <-
  lmer(life_phys_x100 ~ (1|cowcode) + as.factor(year) + latent_personalism +
         lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
         lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(life_phys_control, vcov = "CR2", test = "Satterthwaite",
```

```
          coefs = "All", p_values = TRUE)
conf_int(life_phys_control, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
life_phys_control_ICC_frame <- as.data.frame(VarCorr(life_phys_control))
life_phys_control_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(life_phys_control_ICC_frame[,4])/ sum(life_phys_intercept_ICC_frame[,4])

#       ICC
life_phys_control_ICC_frame[1, 4] / (life_phys_control_ICC_frame[1, 4] +
                                       life_phys_control_ICC_frame[2, 4])

#     Removing year-fixed effects
life_phys_rig_control_no_yearfixed <-
  lmer(life_phys_x100 ~ (1|cowcode) + latent_personalism + lag_e_miinteco +
         lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
         lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)
coef_test(life_phys_rig_control_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against personalism-only model
deviance_life_phys_control <-
  deviance(lmer(life_phys_x100 ~ (1|cowcode) + as.factor(year) +
                  latent_personalism + lag_e_miinteco + lag_e_miinterc +
                  lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
                  lag_log10pop + gwf_monarch + gwf_military + gwf_party,
                data = datacomplete, REML = FALSE))
deviance_life_phys_personalism - deviance_life_phys_control
pchisq(deviance_life_phys_personalism - deviance_life_phys_control,
       df = 9, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_life_phys_control_no_yearfixed <-
  deviance(lmer(life_phys_x100 ~ (1|cowcode) + latent_personalism +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_life_phys_control_no_yearfixed - deviance_life_phys_control
pchisq(deviance_life_phys_control_no_yearfixed - deviance_life_phys_control,
       df = 54, lower.tail = FALSE)

#   Previous model with rigour and impartiality of the public administration
life_phys_rig_impart <-
  lmer(life_phys_x100 ~ (1|cowcode) + as.factor(year) + latent_personalism +
         v2clrspct + lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
         lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
         gwf_military + gwf_party, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(life_phys_rig_impart, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(life_phys_rig_impart, vcov = "CR2", level = 0.9875,
```

```
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
life_phys_rig_impart_ICC_frame <- as.data.frame(VarCorr(life_phys_rig_impart))
life_phys_rig_impart_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(life_phys_rig_impart_ICC_frame[,4])/
  sum(life_phys_intercept_ICC_frame[,4])

#       ICC
life_phys_rig_impart_ICC_frame[1, 4] / (life_phys_rig_impart_ICC_frame[1, 4] +
                                          life_phys_rig_impart_ICC_frame[2, 4])

#     Removing year-fixed effects
life_phys_rig_impart_no_yearfixed <-
  lmer(life_phys_x100 ~ (1|cowcode) + latent_personalism + v2clrspct +
         lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
         lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)
coef_test(life_phys_rig_impart_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite",
          coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against model without rigour and impartiality
#       of the public administration
deviance_life_phys_rig_impart <-
  deviance(lmer(life_phys_x100 ~ (1|cowcode) + as.factor(year) +
                  latent_personalism + v2clrspct + lag_e_miinteco +
                  lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
                  lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military +
                  gwf_party, data = datacomplete, REML = FALSE))
deviance_life_phys_control - deviance_life_phys_rig_impart
pchisq(deviance_life_phys_control - deviance_life_phys_rig_impart,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_life_phys_rig_impart_no_yearfixed <-
  deviance(lmer(life_phys_x100 ~ (1|cowcode) + latent_personalism + v2clrspct +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_life_phys_rig_impart_no_yearfixed - deviance_life_phys_rig_impart
pchisq(deviance_life_phys_rig_impart_no_yearfixed -
         deviance_life_phys_rig_impart, df = 54, lower.tail = FALSE)

#     Residual diagnostics

#       Add predicted values to dataset
datacomplete$life_phys_predict <- predict(life_phys_rig_impart)

#       Add residuals to dataset

#         Calculate level 1 residuals
datacomplete$life_phys_resid_level_1 <- residuals(life_phys_rig_impart)

#         Standardise level 1 residuals
```

```
for(c in unique(datacomplete$cowcode)) {
  datacomplete$life_phys_country_resid_sds[datacomplete$cowcode == c] <-
    sd(datacomplete$life_phys_resid_level_1[datacomplete$cowcode == c])
}
datacomplete$life_phys_stand_resid_level_1 <-
  datacomplete$life_phys_resid_level_1 /
  datacomplete$life_phys_country_resid_sds

#       Calculate level 2 residuals
life_phys_resid_level_2 <- as.data.frame(ranef(life_phys_rig_impart))
life_phys_resid_level_2$cowcode <-
  as.numeric(levels(life_phys_resid_level_2$grp))[life_phys_resid_level_2$grp]

#       Standardise level 2 residuals
life_phys_resid_level_2$life_phys_stand_resid_level_2 <-
  life_phys_resid_level_2$condval / life_phys_rig_impart_ICC_frame$sdcor[1]

#       Add level 2 residuals to dataset
datacomplete <-
  left_join(datacomplete,
            life_phys_resid_level_2[, c("cowcode", "condval",
                                        "life_phys_stand_resid_level_2")],
            by = c("cowcode"))
datacomplete <- rename(datacomplete, life_phys_resid_level_2 = condval)

#       Autocorrelation

#         Create dataframe with just the residuals
life_phys_resid_pdata <-
  pdata.frame(datacomplete[, c("year", "cowcode", "life_phys_resid_level_1")],
              index = c("cowcode", "year"))
#         Create lagged residuals
for (j in 1:20) {
  # Create the name of the lagged variable
  var <- paste("life_phys_lag", j ,"_resid_level_1", sep = "")
  # Generate the lagged variable via plm's implementation of "lag"
  life_phys_resid_pdata[, var] <-
    plm::lag(life_phys_resid_pdata$life_phys_resid_level_1, k= j)
}

#         Calculate correlations between present values and lags
autocors_life_phys_resid_level_1 <-
  cor(life_phys_resid_pdata$life_phys_resid_level_1,
      life_phys_resid_pdata[, 4:ncol(life_phys_resid_pdata)],
      use = "complete.obs")

#         Reschape the correlations into a more workable format
autocors_life_phys_resid_level_1 <-
  tibble(Lag = 1:20, Correlation = t(autocors_life_phys_resid_level_1))

#         Create an autocorrelation plot
ggplot(data = autocors_life_phys_resid_level_1, aes(x = Lag, y = Correlation)) +
  geom_col(fill = "lightblue") + blue_light

#       Linearity

#         Plot predicted values against level 1 residuals
ggplot(datacomplete, aes(x = life_phys_predict,
                         y = life_phys_stand_resid_level_1)) +
```

```
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Predicted value", y = "Level 1 residual (standardised)")

#       Plot values predicted by fixed part against level 2 residuals
ggplot(datacomplete, aes(x = life_phys_predict - life_phys_resid_level_2,
                         y = life_phys_stand_resid_level_2)) +
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Fixed part predictions", y = "Level 2 residual (standardised)")

#       Normality

#         Assign residual levels to the residuals for combination
rbind_resids_level_2 <-
  rename(life_phys_resid_level_2[, c("cowcode",
                                     "life_phys_stand_resid_level_2")],
         resid = life_phys_stand_resid_level_2)
rbind_resids_level_2$level <- "Level 2"
rbind_resids_level_1 <-
  rename(datacomplete[, c("cowcode", "life_phys_stand_resid_level_1")],
         resid = life_phys_stand_resid_level_1)
rbind_resids_level_1$level <- "Level 1"

#         Create long dataset of all residuals
life_phys_resids <- rbind(rbind_resids_level_1, rbind_resids_level_2)
rm(list = c("rbind_resids_level_1", "rbind_resids_level_2"))

#         Plot residuals in dual QQ-plot
withr::with_options(
  list(ggplot2.discrete.colour = c("#E69F00", "#56B4E9", "#009E73", "#F0E442",
                                   "#0072B2", "#D55E00", "#CC79A7")),
  print(ggplot(data = life_phys_resids,
               aes(sample = resid, colour = as.factor(level))) + geom_qq() +
          geom_qq_line() + blue_light +
          scale_colour_discrete(name = "Residual level") +
          labs(y = "Standardised residuals",
               x = "Reference normal distribution")) )

#         Plot residuals in single QQ-plots

#             Level 1
ggplot(data = life_phys_resids[life_phys_resids$level == "Level 1", ],
       aes(sample = resid)) +
  geom_qq(colour = "lightblue") + geom_qq_line() + blue_light +
  labs(y = "Standardised residuals", x = "Reference normal distribution")

#             Level 2
ggplot(data = life_phys_resids[life_phys_resids$level == "Level 2", ],
       aes(sample = resid)) +
  geom_qq(colour = "lightblue") + geom_qq_line() + blue_light +
  labs(y = "Standardised residuals", x = "Reference normal distribution")

############################## Freedom of movement ############################

#   Intercept-only model
free_move_intercept <- lmer(freedom_movement ~ (1|cowcode), data = datacomplete)
```

```
#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_move_intercept, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_move_intercept, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_move_intercept_ICC_frame <- as.data.frame(VarCorr(free_move_intercept))
free_move_intercept_ICC_frame

#       ICC
free_move_intercept_ICC_frame[1, 4] / (free_move_intercept_ICC_frame[1, 4] +
                                         free_move_intercept_ICC_frame[2, 4])

#     Calculate deviance
deviance_free_move_intercept <- deviance(lmer(freedom_movement ~ (1|cowcode),
                                              data = datacomplete,
                                              REML = FALSE))

#   Model with year-fixed effects
free_move_yearfixed <- lmer(freedom_movement ~ (1|cowcode) + as.factor(year),
                            data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_move_yearfixed, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_move_yearfixed, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_move_yearfixed_ICC_frame <- as.data.frame(VarCorr(free_move_yearfixed))
free_move_yearfixed_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_move_yearfixed_ICC_frame[,4])/
  sum(free_move_intercept_ICC_frame[,4])

#       ICC
free_move_yearfixed_ICC_frame[1, 4] / (free_move_yearfixed_ICC_frame[1, 4] +
                                         free_move_yearfixed_ICC_frame[2, 4])

#     LR-test

#       LR-test against intercept-only model
deviance_free_move_yearfixed <-
  deviance(lmer(freedom_movement ~ (1|cowcode) + as.factor(year),
                data = datacomplete, REML = FALSE))
deviance_free_move_intercept - deviance_free_move_yearfixed
pchisq(deviance_free_move_intercept - deviance_free_move_yearfixed,
       df = 54, lower.tail = FALSE)

#   Model with personalism
free_move_personalism <- lmer(freedom_movement ~ (1|cowcode) + as.factor(year) +
                                latent_personalism, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
```

```
#     intervals
coef_test(free_move_personalism, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_move_personalism, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_move_personalism_ICC_frame <- as.data.frame(VarCorr(free_move_personalism))
free_move_personalism_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_move_personalism_ICC_frame[,4])/
  sum(free_move_intercept_ICC_frame[,4])

#       ICC
free_move_personalism_ICC_frame[1, 4] /
  (free_move_personalism_ICC_frame[1, 4] +
     free_move_personalism_ICC_frame[2, 4])

#     Removing year-fixed effects
free_move_rig_personalism_no_yearfixed <-
  lmer(freedom_movement ~ (1|cowcode) + latent_personalism, data = datacomplete)
coef_test(free_move_rig_personalism_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against model with year-fixed effects
deviance_free_move_personalism <-
  deviance(lmer(freedom_movement ~ (1|cowcode) + as.factor(year) +
                  latent_personalism, data = datacomplete, REML = FALSE))

deviance_free_move_yearfixed - deviance_free_move_personalism
pchisq(deviance_free_move_yearfixed - deviance_free_move_personalism,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_move_personalism_no_yearfixed <-
  deviance(lmer(freedom_movement ~ (1|cowcode) + latent_personalism,
                data = datacomplete, REML = FALSE))
deviance_free_move_personalism_no_yearfixed - deviance_free_move_personalism
pchisq(deviance_free_move_personalism_no_yearfixed -
         deviance_free_move_personalism,
       df = 54, lower.tail = FALSE)

#   Model with personalism and controls
free_move_control <-
  lmer(freedom_movement ~ (1|cowcode) + as.factor(year) + latent_personalism +
         lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
         lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_move_control, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_move_control, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)
```

```
#     Retrieve variance decomposition
free_move_control_ICC_frame <- as.data.frame(VarCorr(free_move_control))
free_move_control_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_move_control_ICC_frame[,4])/sum(free_move_intercept_ICC_frame[,4])

#       ICC
free_move_control_ICC_frame[1, 4] / (free_move_control_ICC_frame[1, 4] +
                                       free_move_control_ICC_frame[2, 4])

#     Removing year-fixed effects
free_move_rig_control_no_yearfixed <-
  lmer(freedom_movement ~ (1|cowcode) + latent_personalism + lag_e_miinteco +
         lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
         lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)
coef_test(free_move_rig_control_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against personalism-only model
deviance_free_move_control <-
  deviance(lmer(freedom_movement ~ (1|cowcode) + as.factor(year) +
                  latent_personalism + lag_e_miinteco + lag_e_miinterc +
                  lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
                  lag_log10pop + gwf_monarch + gwf_military + gwf_party,
                data = datacomplete, REML = FALSE))
deviance_free_move_personalism - deviance_free_move_control
pchisq(deviance_free_move_personalism - deviance_free_move_control,
       df = 9, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_move_control_no_yearfixed <-
  deviance(lmer(freedom_movement ~ (1|cowcode) + latent_personalism +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_free_move_control_no_yearfixed - deviance_free_move_control
pchisq(deviance_free_move_control_no_yearfixed - deviance_free_move_control,
       df = 54, lower.tail = FALSE)

#   Previous model with rigour and impartiality of the public administration
free_move_rig_impart <-
  lmer(freedom_movement ~ (1|cowcode) + as.factor(year) + latent_personalism +
         v2clrspct + lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
         lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
         gwf_military + gwf_party, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(free_move_rig_impart, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(free_move_rig_impart, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
free_move_rig_impart_ICC_frame <- as.data.frame(VarCorr(free_move_rig_impart))
```

```
free_move_rig_impart_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(free_move_rig_impart_ICC_frame[,4])/
  sum(free_move_intercept_ICC_frame[,4])

#       ICC
free_move_rig_impart_ICC_frame[1, 4] / (free_move_rig_impart_ICC_frame[1, 4] +
                                          free_move_rig_impart_ICC_frame[2, 4])

#     Removing year-fixed effects
free_move_rig_impart_no_yearfixed <-
  lmer(freedom_movement ~ (1|cowcode) + latent_personalism + v2clrspct +
         lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
         lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)
coef_test(free_move_rig_impart_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against model without rigour and impartiality
#       of the public administration
deviance_free_move_rig_impart <-
  deviance(lmer(freedom_movement ~ (1|cowcode) + as.factor(year) +
                  latent_personalism + v2clrspct + lag_e_miinteco +
                  lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
                  lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military +
                  gwf_party, data = datacomplete, REML = FALSE))
deviance_free_move_control - deviance_free_move_rig_impart
pchisq(deviance_free_move_control - deviance_free_move_rig_impart,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_free_move_rig_impart_no_yearfixed <-
  deviance(lmer(freedom_movement ~ (1|cowcode) + latent_personalism +
                  v2clrspct + lag_e_miinteco + lag_e_miinterc +
                  lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
                  lag_log10pop + gwf_monarch + gwf_military + gwf_party,
                data = datacomplete, REML = FALSE))
deviance_free_move_rig_impart_no_yearfixed - deviance_free_move_rig_impart
pchisq(deviance_free_move_rig_impart_no_yearfixed -
         deviance_free_move_rig_impart, df = 54, lower.tail = FALSE)

#     Residual diagnostics

#       Add predicted values to dataset
datacomplete$free_move_predict <- predict(free_move_rig_impart)

#       Add residuals to dataset

#         Calculate level 1 residuals
datacomplete$free_move_resid_level_1 <- residuals(free_move_rig_impart)

#         Standardise level 1 residuals
for(c in unique(datacomplete$cowcode)) {
  datacomplete$free_move_country_resid_sds[datacomplete$cowcode == c] <-
    sd(datacomplete$free_move_resid_level_1[datacomplete$cowcode == c])
}
```

```
datacomplete$free_move_stand_resid_level_1 <-
  datacomplete$free_move_resid_level_1 /
  datacomplete$free_move_country_resid_sds

#       Calculate level 2 residuals
free_move_resid_level_2 <- as.data.frame(ranef(free_move_rig_impart))
free_move_resid_level_2$cowcode <-
  as.numeric(levels(free_move_resid_level_2$grp))[free_move_resid_level_2$grp]

#       Standardise level 2 residuals
free_move_resid_level_2$free_move_stand_resid_level_2 <-
  free_move_resid_level_2$condval / free_move_rig_impart_ICC_frame$sdcor[1]

#       Add level 2 residuals to dataset
datacomplete <-
  left_join(datacomplete,
            free_move_resid_level_2[, c("cowcode", "condval",
                                        "free_move_stand_resid_level_2")],
            by = c("cowcode"))
datacomplete <- rename(datacomplete, free_move_resid_level_2 = condval)

#       Autocorrelation

#         Create dataframe with just the residuals
free_move_resid_pdata <-
  pdata.frame(datacomplete[, c("year", "cowcode", "free_move_resid_level_1")],
              index = c("cowcode", "year"))
#         Create lagged residuals
for (j in 1:20) {
  # Create the name of the lagged variable
  var <- paste("free_move_lag", j ,"_resid_level_1", sep = "")
  # Generate the lagged variable via plm's implementation of "lag"
  free_move_resid_pdata[, var] <-
    plm::lag(free_move_resid_pdata$free_move_resid_level_1, k= j)
}

#         Calculate correlations between present values and lags
autocors_free_move_resid_level_1 <-
  cor(free_move_resid_pdata$free_move_resid_level_1,
      free_move_resid_pdata[, 4:ncol(free_move_resid_pdata)],
      use = "complete.obs")

#         Reschape the correlations into a more workable format
autocors_free_move_resid_level_1 <-
  tibble(Lag = 1:20, Correlation = t(autocors_free_move_resid_level_1))

#         Create an autocorrelation plot
ggplot(data = autocors_free_move_resid_level_1, aes(x = Lag, y = Correlation)) +
  geom_col(fill = "lightblue") + blue_light

#       Linearity

#         Plot predicted values against level 1 residuals
ggplot(datacomplete, aes(x = free_move_predict,
                         y = free_move_stand_resid_level_1)) +
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Predicted value", y = "Level 1 residual (standardised)")
```

```
#       Plot fixed part predictions against level 2 residuals
ggplot(datacomplete, aes(x = free_move_predict - free_move_resid_level_2,
                         y = free_move_stand_resid_level_2)) +
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Fixed part predictions", y = "Level 2 residual (standardised)")

#       Normality

#         Assign residual levels to the residuals for combination
rbind_resids_level_2 <-
  rename(free_move_resid_level_2[, c("cowcode",
                                     "free_move_stand_resid_level_2")],
         resid = free_move_stand_resid_level_2)
rbind_resids_level_2$level <- "Level 2"
rbind_resids_level_1 <-
  rename(datacomplete[, c("cowcode", "free_move_stand_resid_level_1")],
         resid = free_move_stand_resid_level_1)
rbind_resids_level_1$level <- "Level 1"

#         Create long dataset of all residuals
free_move_resids <- rbind(rbind_resids_level_1, rbind_resids_level_2)
rm(list = c("rbind_resids_level_1", "rbind_resids_level_2"))

#         Plot residuals in dual QQ-plot
withr::with_options(
  list(ggplot2.discrete.colour = c("#E69F00", "#56B4E9", "#009E73", "#F0E442",
                                   "#0072B2", "#D55E00", "#CC79A7")),
  print(ggplot(data = free_move_resids, aes(sample = resid,
                                            colour = as.factor(level))) +
          geom_qq() + geom_qq_line() + blue_light +
          scale_colour_discrete(name = "Residual level") +
          labs(y = "Standardised residuals",
               x = "Reference normal distribution")) )

#         Plot residuals in single QQ-plots
#             Level 1
ggplot(data = free_move_resids[free_move_resids$level == "Level 1", ],
       aes(sample = resid)) + geom_qq(colour = "lightblue") + geom_qq_line() +
  blue_light + labs(y = "Standardised residuals",
                    x = "Reference normal distribution")

#             Level 2
ggplot(data = free_move_resids[free_move_resids$level == "Level 2", ],
       aes(sample = resid)) + geom_qq(colour = "lightblue") + geom_qq_line() +
  blue_light + labs(y = "Standardised residuals",
                    x = "Reference normal distribution")

############# Rigour and impartiality of the public administration ############

#   Random intercept-only model
rig_impart_intercept <- lmer(v2clrspct ~ (1|cowcode), data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(rig_impart_intercept, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
```

```
conf_int(rig_impart_intercept, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
rig_impart_intercept_ICC_frame <- as.data.frame(VarCorr(rig_impart_intercept))
rig_impart_intercept_ICC_frame

#     ICC
rig_impart_intercept_ICC_frame[1, 4] / (rig_impart_intercept_ICC_frame[1, 4] +
                                          rig_impart_intercept_ICC_frame[2, 4])

#     Calculate deviance
deviance_rig_impart_intercept <- deviance(lmer(v2clrspct ~ (1|cowcode),
                                               data = datacomplete,
                                               REML = FALSE))

#   Model with time-fixed effects
rig_impart_yearfixed <- lmer(v2clrspct ~ (1|cowcode) + as.factor(year),
                             data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(rig_impart_yearfixed, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(rig_impart_yearfixed, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
rig_impart_yearfixed_ICC_frame <- as.data.frame(VarCorr(rig_impart_yearfixed))
rig_impart_yearfixed_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(rig_impart_yearfixed_ICC_frame[,4])/
  sum(rig_impart_intercept_ICC_frame[,4])

#     ICC
rig_impart_yearfixed_ICC_frame[1, 4] / (rig_impart_yearfixed_ICC_frame[1, 4] +
                                          rig_impart_yearfixed_ICC_frame[2, 4])

#     LR-test against intercept-only model
deviance_rig_impart_yearfixed <-
  deviance(lmer(v2clrspct ~ (1|cowcode) + as.factor(year), data = datacomplete,
                REML = FALSE))
deviance_rig_impart_intercept - deviance_rig_impart_yearfixed
pchisq(deviance_rig_impart_intercept - deviance_rig_impart_yearfixed,
       df = 54, lower.tail = FALSE)

#   Model with personalism
rig_impart_personalism <- lmer(v2clrspct ~ (1|cowcode) + as.factor(year) +
                                 latent_personalism, data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(rig_impart_personalism, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(rig_impart_personalism, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
```

```
rig_impart_personalism_ICC_frame <-
  as.data.frame(VarCorr(rig_impart_personalism))
rig_impart_personalism_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(rig_impart_personalism_ICC_frame[,4])/
  sum(rig_impart_intercept_ICC_frame[,4])

#       ICC
rig_impart_personalism_ICC_frame[1, 4] /
  (rig_impart_personalism_ICC_frame[1, 4] +
     rig_impart_personalism_ICC_frame[2, 4])

#     Removing year-fixed effects
rig_impart_personalism_no_yearfixed <-
  lmer(v2clrspct ~ (1|cowcode) + latent_personalism, data = datacomplete)
coef_test(rig_impart_personalism_no_yearfixed, vcov = "CR2",
          test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against year-fixed effects model
deviance_rig_impart_personalism <-
  deviance(lmer(v2clrspct ~ (1|cowcode) + as.factor(year) + latent_personalism,
                data = datacomplete, REML = FALSE))
deviance_rig_impart_yearfixed - deviance_rig_impart_personalism
pchisq(deviance_rig_impart_intercept - deviance_rig_impart_personalism,
       df = 1, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_rig_impart_personalism_no_yearfixed <-
  deviance(lmer(v2clrspct ~ (1|cowcode) + latent_personalism,
                data = datacomplete, REML = FALSE))
deviance_rig_impart_personalism_no_yearfixed - deviance_rig_impart_personalism
pchisq(deviance_rig_impart_personalism_no_yearfixed -
         deviance_rig_impart_personalism,
       df = 54, lower.tail = FALSE)

#   Model with personalism and controls
rig_impart_control <-
  lmer(v2clrspct ~ (1|cowcode) + as.factor(year) + latent_personalism +
         lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
         lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)

#     Calculate clustered standard errors, t-values, p-values and confidence
#     intervals
coef_test(rig_impart_control, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)
conf_int(rig_impart_control, vcov = "CR2", level = 0.9875,
         test = "Satterthwaite", coefs = "All", p_values = TRUE)

#     Retrieve variance decomposition
rig_impart_control_ICC_frame <- as.data.frame(VarCorr(rig_impart_control))
rig_impart_control_ICC_frame

#       Calculate R-squared compared to intercept-only model
1 - sum(rig_impart_control_ICC_frame[,4])/
  sum(rig_impart_intercept_ICC_frame[,4])
```

```
#       ICC
rig_impart_control_ICC_frame[1, 4] /
  (rig_impart_control_ICC_frame[1, 4] +
     rig_impart_control_ICC_frame[2, 4])

#     Removing year-fixed effects
rig_impart_control_no_yearfixed <-
  lmer(v2clrspct ~ (1|cowcode) + latent_personalism + lag_e_miinteco +
         lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro + lag_v2caviol +
         lag_log10pop + gwf_monarch + gwf_military + gwf_party,
       data = datacomplete)
coef_test(rig_impart_control_no_yearfixed, vcov = "CR2", test = "Satterthwaite",
          coefs = "All", p_values = TRUE)

#     LR-test

#       LR-test against model without regime type controls
deviance_rig_impart_control <-
  deviance(lmer(v2clrspct ~ (1|cowcode) + as.factor(year) + latent_personalism +
                  lag_e_miinteco + lag_e_miinterc + lag_e_migdppcln +
                  lag_e_migdpgro + lag_v2caviol + lag_log10pop + gwf_monarch +
                  gwf_military + gwf_party, data = datacomplete, REML = FALSE))
deviance_rig_impart_personalism - deviance_rig_impart_control
pchisq(deviance_rig_impart_personalism - deviance_rig_impart_control,
       df = 9, lower.tail = FALSE)

#       LR-test against model without year-fixed effects
deviance_rig_impart_control_no_yearfixed <-
  deviance(lmer(v2clrspct ~ (1|cowcode) + latent_personalism + lag_e_miinteco +
                  lag_e_miinterc + lag_e_migdppcln + lag_e_migdpgro +
                  lag_v2caviol + lag_log10pop + gwf_monarch + gwf_military +
                  gwf_party, data = datacomplete, REML = FALSE))
deviance_rig_impart_control_no_yearfixed - deviance_rig_impart_control
pchisq(deviance_rig_impart_control_no_yearfixed - deviance_rig_impart_control,
       df = 54, lower.tail = FALSE)

#     Residual diagnostics

#       Add predicted values to dataset
datacomplete$rig_impart_predict <- predict(rig_impart_control)

#       Add residuals to dataset

#         Calculate level 1 residuals
datacomplete$rig_impart_resid_level_1 <- residuals(rig_impart_control)

#         Standardise level 1 residuals
for(c in unique(datacomplete$cowcode)) {
  datacomplete$rig_impart_country_resid_sds[datacomplete$cowcode == c] <-
  sd(datacomplete$rig_impart_resid_level_1[datacomplete$cowcode == c])
  }
datacomplete$rig_impart_stand_resid_level_1 <-
  datacomplete$rig_impart_resid_level_1 /
  datacomplete$rig_impart_country_resid_sds

#       Calculate level 2 residuals
rig_impart_resid_level_2 <- as.data.frame(ranef(rig_impart_control))
rig_impart_resid_level_2$cowcode <-
```

```
  as.numeric(levels(rig_impart_resid_level_2$grp))[rig_impart_resid_level_2$grp]

#       Standardise level 2 residuals
rig_impart_resid_level_2$rig_impart_stand_resid_level_2 <-
  rig_impart_resid_level_2$condval / rig_impart_control_ICC_frame$sdcor[1]

#       Add level 2 residuals to dataset
datacomplete <-
  left_join(datacomplete,
            rig_impart_resid_level_2[, c("cowcode", "condval",
                                         "rig_impart_stand_resid_level_2")],
            by = c("cowcode"))
datacomplete <- rename(datacomplete, rig_impart_resid_level_2 = condval)

#       Autocorrelation

#         Create dataframe with just the residuals
rig_impart_resid_pdata <-
  pdata.frame(datacomplete[, c("year", "cowcode", "rig_impart_resid_level_1")],
              index = c("cowcode", "year"))
#         Create lagged residuals
for (j in 1:20) {
  # Create the name of the lagged variable
  var <- paste("rig_impart_lag", j ,"_resid_level_1", sep = "")
  # Generate the lagged variable via plm's implementation of "lag"
  rig_impart_resid_pdata[, var] <-
    plm::lag(rig_impart_resid_pdata$rig_impart_resid_level_1, k= j)
}

#         Calculate correlations between present values and lags
autocors_rig_impart_resid_level_1 <-
  cor(rig_impart_resid_pdata$rig_impart_resid_level_1,
      rig_impart_resid_pdata[, 4:ncol(rig_impart_resid_pdata)],
      use = "complete.obs")

#         Reschape the correlations into a more workable format
autocors_rig_impart_resid_level_1 <-
  tibble(Lag = 1:20, Correlation = t(autocors_rig_impart_resid_level_1))

#         Create an autocorrelation plot
ggplot(data = autocors_rig_impart_resid_level_1, aes(x = Lag,
                                                     y = Correlation)) +
  geom_col(fill = "lightblue") + blue_light

#       Linearity

#         Plot predicted values against level 1 residuals
ggplot(datacomplete, aes(x = rig_impart_predict,
                         y = rig_impart_stand_resid_level_1)) +
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Predicted value", y = "Level 1 residual (standardised)")

#       Plot fixed part predictions against level 2 residuals
ggplot(datacomplete, aes(x = rig_impart_predict - rig_impart_resid_level_2,
                         y = rig_impart_stand_resid_level_2)) +
  geom_point(colour = "lightblue") +
  geom_smooth(method = lm, formula = y ~ x, colour = "black",
```

```
              linetype = "dashed", se = FALSE) + blue_light +
  labs(x = "Fixed part predictions", y = "Level 2 residual (standardised)")

#       Normality

#         Assign residual levels to the residuals for combination
rbind_resids_level_2 <-
  rename(rig_impart_resid_level_2[, c("cowcode",
                                      "rig_impart_stand_resid_level_2")],
         resid = rig_impart_stand_resid_level_2)
rbind_resids_level_2$level <- "Level 2"
rbind_resids_level_1 <-
  rename(datacomplete[, c("cowcode", "rig_impart_stand_resid_level_1")],
         resid = rig_impart_stand_resid_level_1)
rbind_resids_level_1$level <- "Level 1"

#         Create long dataset of all residuals
rig_impart_resids <- rbind(rbind_resids_level_1, rbind_resids_level_2)
rm(list = c("rbind_resids_level_1", "rbind_resids_level_2"))

#         Plot residuals in dual QQ-plot
withr::with_options(
  list(ggplot2.discrete.colour = c("#E69F00", "#56B4E9", "#009E73", "#F0E442",
                                   "#0072B2", "#D55E00", "#CC79A7")),
  print(ggplot(data = rig_impart_resids, aes(sample = resid,
                                             colour = as.factor(level))) +
          geom_qq() + geom_qq_line() + blue_light +
          scale_colour_discrete(name = "Residual level") +
          labs(y = "Standardised residuals",
               x = "Reference normal distribution")) )

#         Plot residuals in single QQ-plots

#             Level 1
ggplot(data = rig_impart_resids[rig_impart_resids$level == "Level 1", ],
               aes(sample = resid)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + blue_light + labs(y = "Standardised residuals",
                                     x = "Reference normal distribution")

#             Level 2
ggplot(data = rig_impart_resids[rig_impart_resids$level == "Level 2", ],
               aes(sample = resid)) + geom_qq(colour = "lightblue") +
  geom_qq_line() + blue_light + labs(y = "Standardised residuals",
                                     x = "Reference normal distribution")
```