Excess Bereavement and Excess Mortality: Some Analysis

In this document, I outline the process for analyzing the data derived from the analysis scripts and generating the quantities we need for the regression.

```
#Loading dependencies
require(tidyverse)
require(data.table)
require(countrycode) #For labeling graphs
require(stargazer)
```

First, load the data.

```
data <- get(load(file = "../Data/final_data_preprint.RData"))
remove(data_preprint)
```

Independent Variables

Excess Mortality

The first question is how we define excess: we run simulations for two scenarios, one without Covid-related excess mortality and one with it. We then randomly match each Covid simulation for a country with a non-Covid simulation. Most definitions of excess mortality focus on the differences between observed and expected deaths (see, for example, Our World in Data's page), and we calculate a rate here. Where number of deaths in a particular category are denoted by D and population exposure is the number of individuals at risk of dying (ie. alive at the start of the interval, and averaged across the Covid and counterfactual simulation),

- Level of above-15 excess mortality rate: $100 \times (\frac{D_{15+,Covid} - D_{15+,Counterfactual}}{Exposure})$
- Age ratio for above-15 excess mortality:

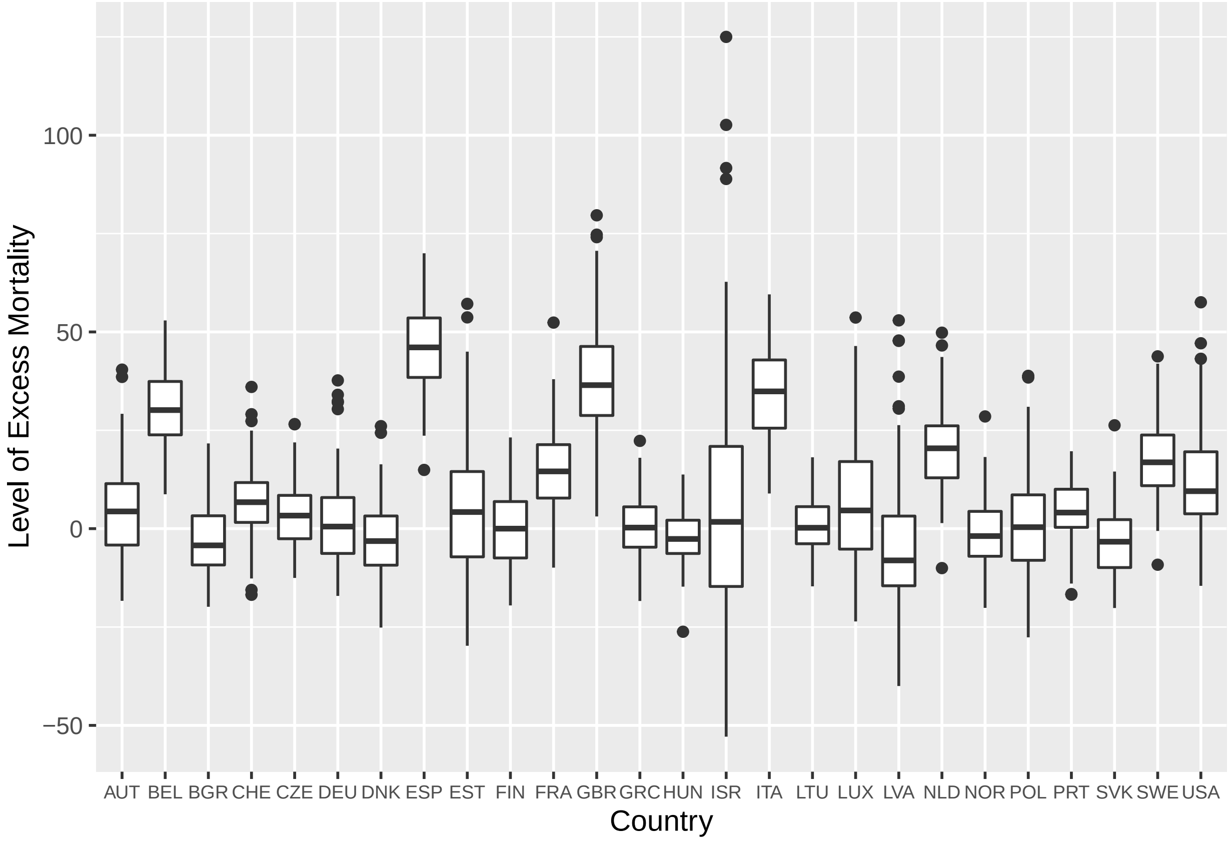
  What share of excess deaths come from the age 65+ category: $\frac{D_{65+,Covid} - D_{65+,Counterfactual}}{D_{15+,Covid} - D_{15+,Counterfactual}}$.
- Sex ratio for above-15 excess mortality: Here, we measure the ratio of excess deaths for men to excess deaths for women.

$$Share = \frac{(D_{15+,Covid,Male} - D_{15+,Counterfactual,Male})}{(D_{15+,Covid,Female} - D_{15+,Counterfactual,Female})}$$

We can calculate these quantities as follows:

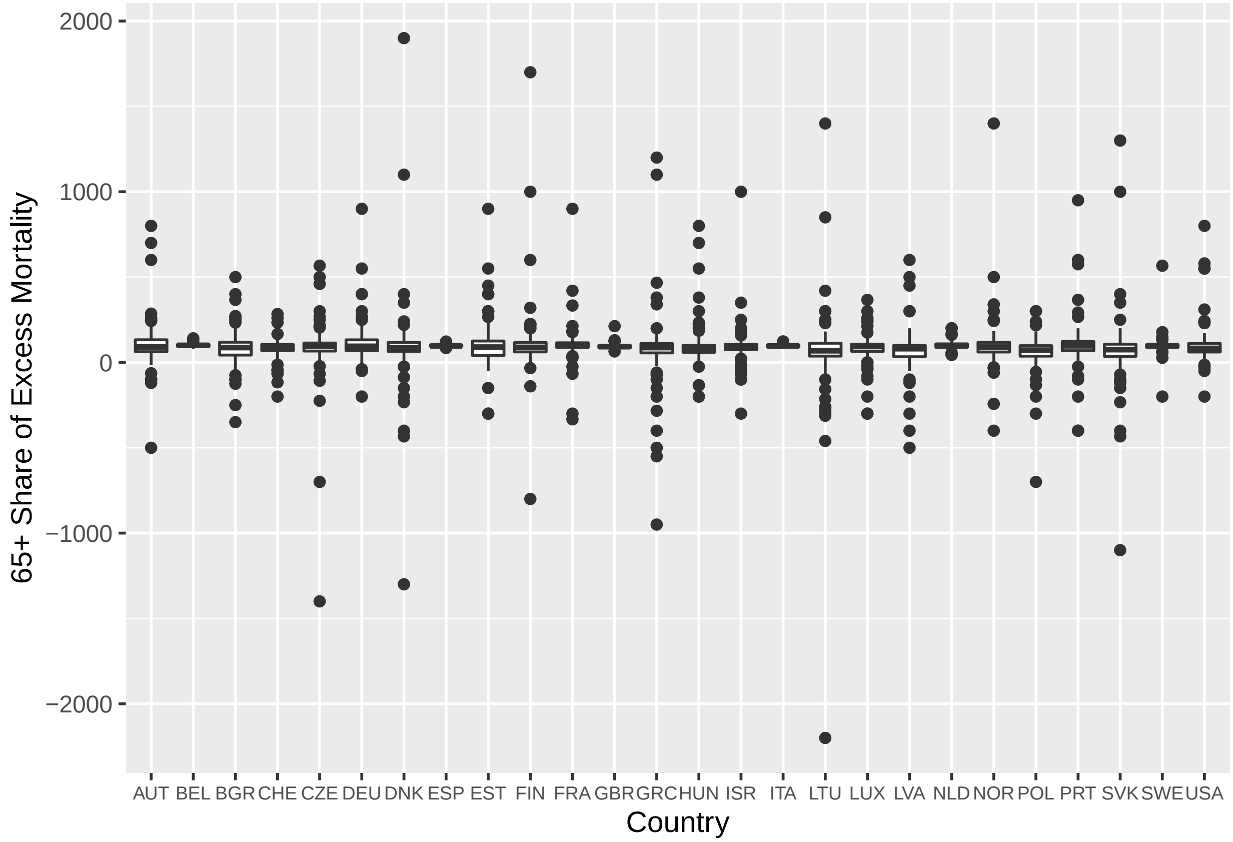
```
#Match Covid and counterfactual simulations, and clean data
death_rates <- data$death_rates %>%
  filter(!is.na(scenario)) %>%
  mutate(sex = ifelse(fem == 1, "F", "M")) %>%
  ungroup() %>%
  group_by(country, scenario, sex, age) %>%
  mutate(index = row_number(), fem = NULL) %>%
  filter(index <= 99) %>% #Temporary solution for addressing matching issue
  ungroup() %>%
  pivot_wider(id_cols = c("country", "sex", "age", "index"),
              names_from = "scenario",
              values_from = c("n_num", "n_den", "value", "sim.id"))
```

```
#Excess mortality level
em_level <- death_rates %>%
  filter(age %in% c("15-29", "30-44", "45-64", "65+")) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other) %>%
  summarize(deaths_covid = sum(n_num_covid, na.rm = T),
            deaths_other = sum(n_num_other, na.rm = T)) %>%
  mutate(em_level = 100*(deaths_covid - deaths_other)/(deaths_other),
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c'))

#Get the distribution of this by country and plot it
em_level %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = em_level)) +
  labs(x = "Country", y = "Level of Excess Mortality") +
  theme(axis.text.x = element_text(size = 7))
```



```
#Age ratio of excess mortality
em_age <- death_rates %>%
  filter(age %in% c("15-29", "30-44", "45-64", "65+")) %>%
  mutate(above65 = if_else(age %in% c("65+"), "65plus", "below65")) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other, above65) %>%
  summarize(deaths_covid = sum(n_num_covid, na.rm = T),
            deaths_other = sum(n_num_other, na.rm = T)) %>%
  pivot_wider(names_from = "above65", values_from = c("deaths_covid", "deaths_other")) %>%
  mutate(em_share_age = 100*(deaths_covid_65plus - deaths_other_65plus)/(deaths_covid_below65+deaths_co
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c'))

#Graph this
em_age %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = em_share_age)) +
  labs(x = "Country", y = "65+ Share of Excess Mortality") +
  theme(axis.text.x = element_text(size = 7))
```

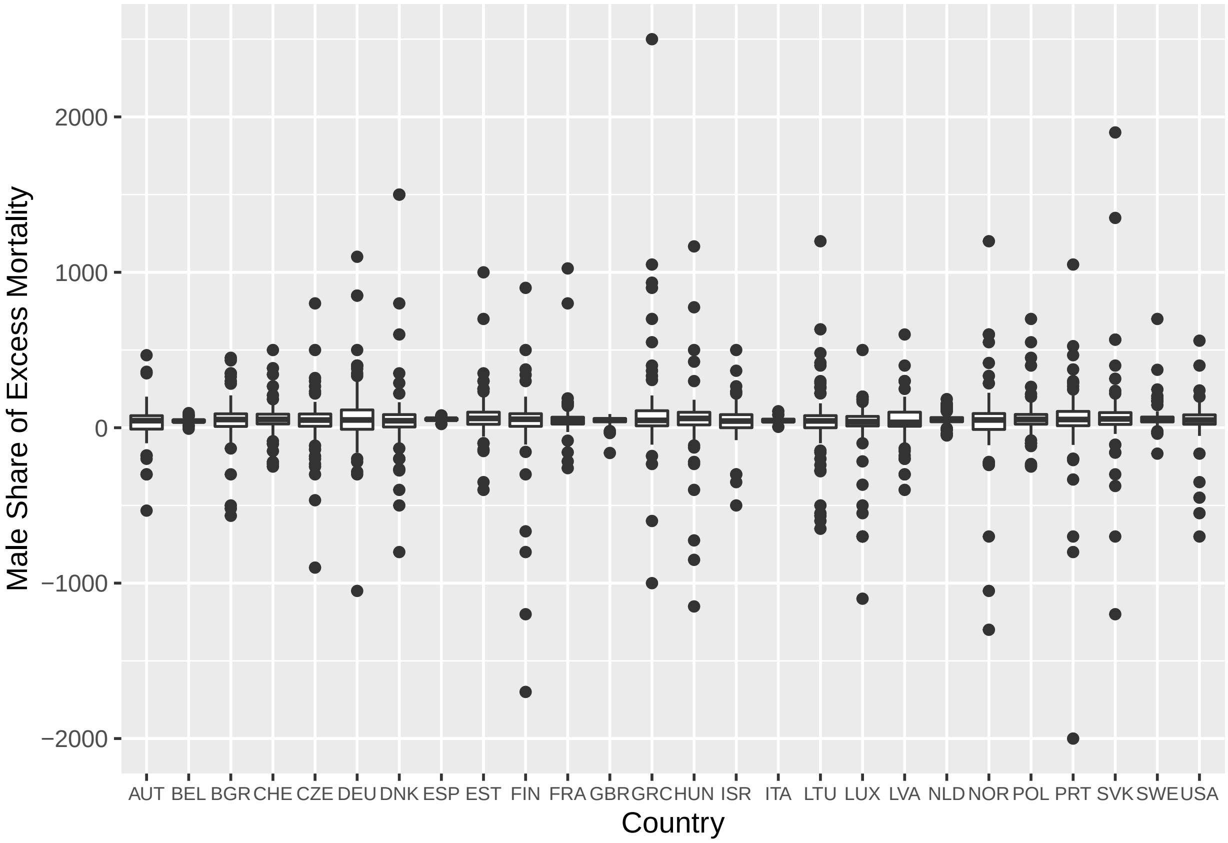
```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
```



```
#Sex ratio of excess mortality
em_sexratio <- death_rates %>%
  filter(age %in% c("15-29", "30-44", "45-64", "65+")) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other, sex) %>%
  summarize(deaths_covid = sum(n_num_covid, na.rm = T),
            deaths_other = sum(n_num_other, na.rm = T)) %>%
  pivot_wider(names_from = "sex", values_from = c("deaths_covid", "deaths_other")) %>%
  mutate(em_sexratio = 100*(deaths_covid_M - deaths_other_M)/(deaths_covid_F + deaths_covid_M - deaths_
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c'))

#Graph this
```

```r
em_sexratio %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = em_sexratio)) +
  labs(x = "Country", y = "Male Share of Excess Mortality") +
  theme(axis.text.x = element_text(size = 7))
```

```
## Warning: Removed 26 rows containing non-finite values (stat_boxplot).
```



Population Structure

We can calculate these quantities from the death rates object as well, using population exposures. One thing to note is that for each pair of simulations, we have two sets of exposures. Here, I simply average over the two cases, as this is pre-Covid-19 and unlikely to affect the results.

- Level of population

  This is calculated as the sum of all population exposures. Consistent with other quantities used, I think it may make make sense to have it be in terms of size of population aged 15 and over. For now, I calculate it based on all individuals. I also plot UNWPP populations for each of the countries (we will use the UNWPP populations as a covariate, not SOCSIM populations.) According to their documentation, UNWPP exclude dependencies.

- Age ratio above 15

  This is calculated as a ratio, where N is the number of individuals in a particular category

  $$Ratio = 100 \times \frac{N_{65+}}{N_{15-64}}$$

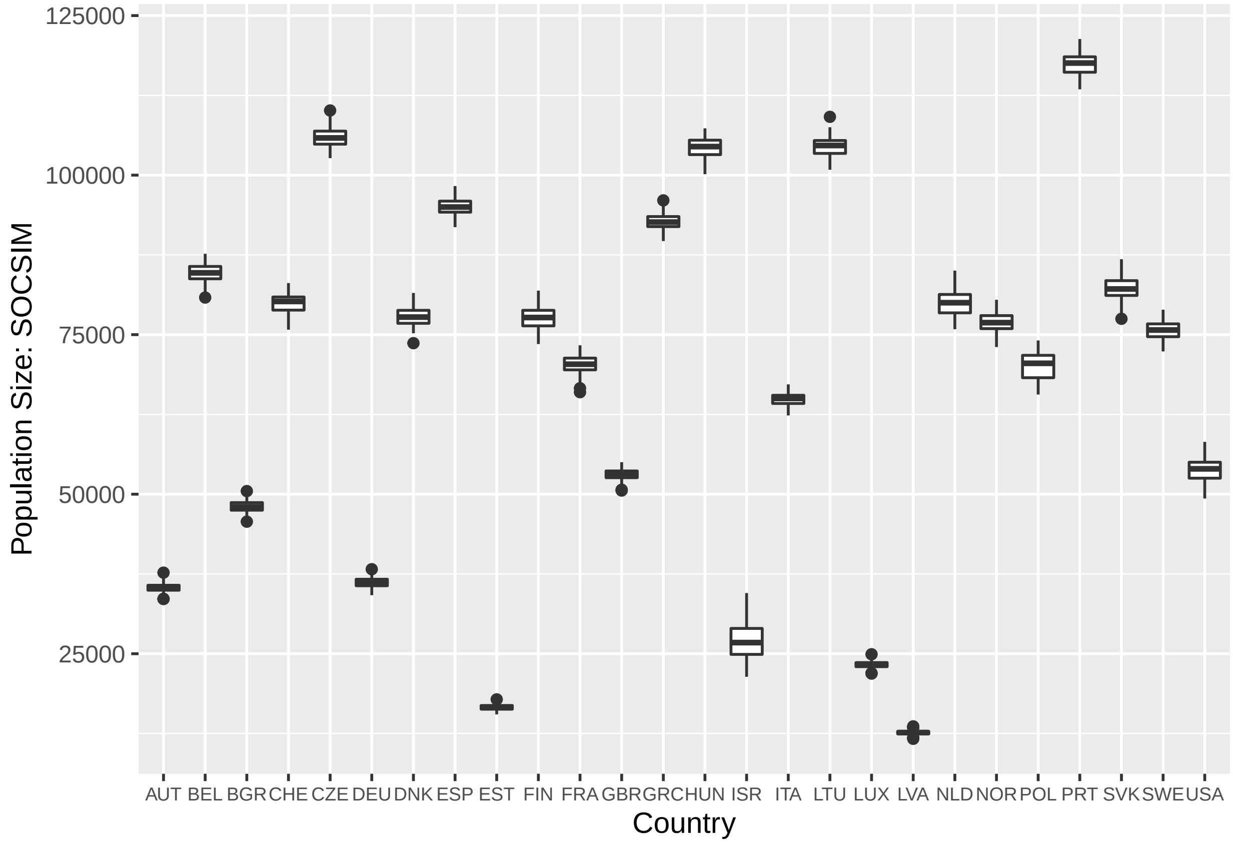
- Sex ratio above 15

Similarly, this is a ratio,

$$Ratio = 100 \times \frac{N_{Male}}{N_{Female}}$$

I use the same death_rates object from previously to calculate these quantities.

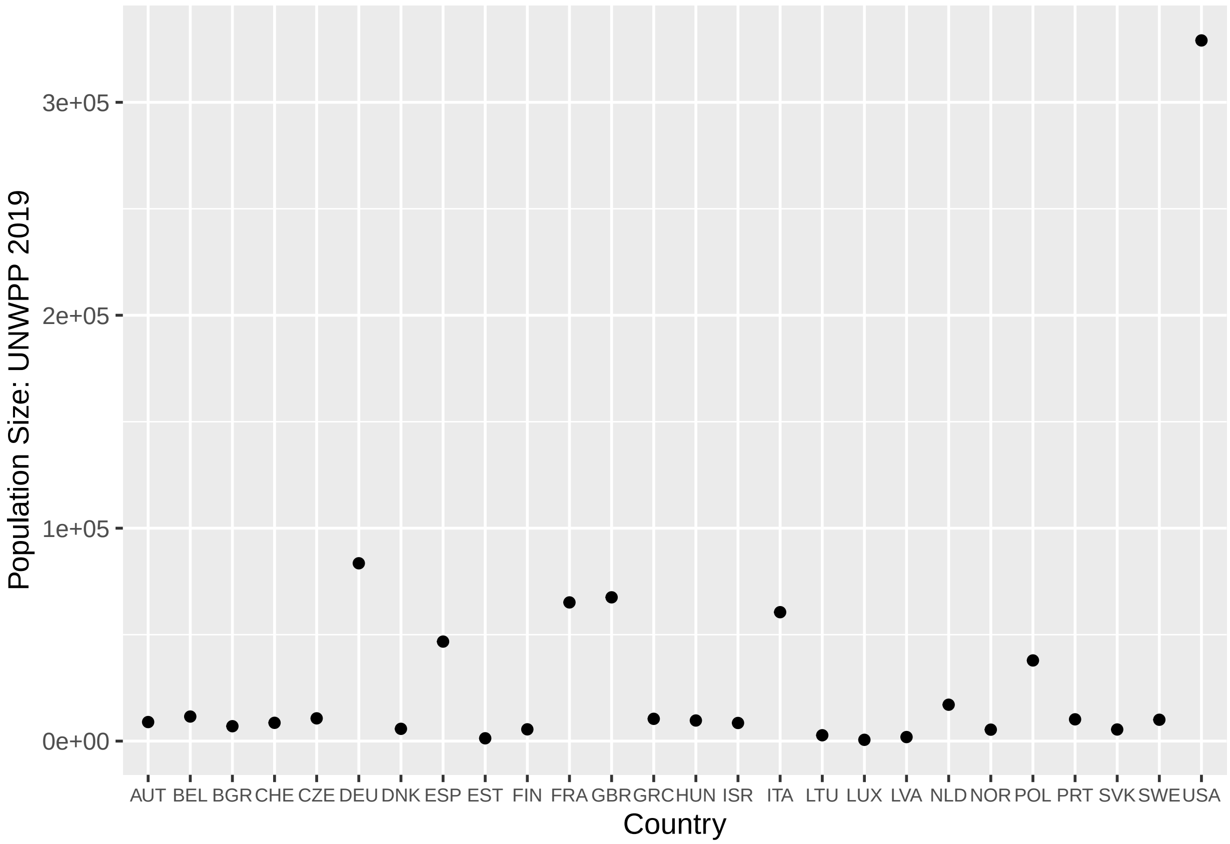
```
#Calculating size of population
pop_size <- death_rates %>%
  filter(age == "all_sum") %>% #This previously summed up population exposures
  mutate(pop_size = (n_den_covid+n_den_other)/2) %>%
  select(c(country, index, sim.id_covid, sim.id_other, pop_size))

#Graph these results
pop_size %>%
 mutate(ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c')) %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = pop_size)) +
  labs(x = "Country", y = "Population Size: SOCSIM") +
  theme(axis.text.x = element_text(size = 7))
```



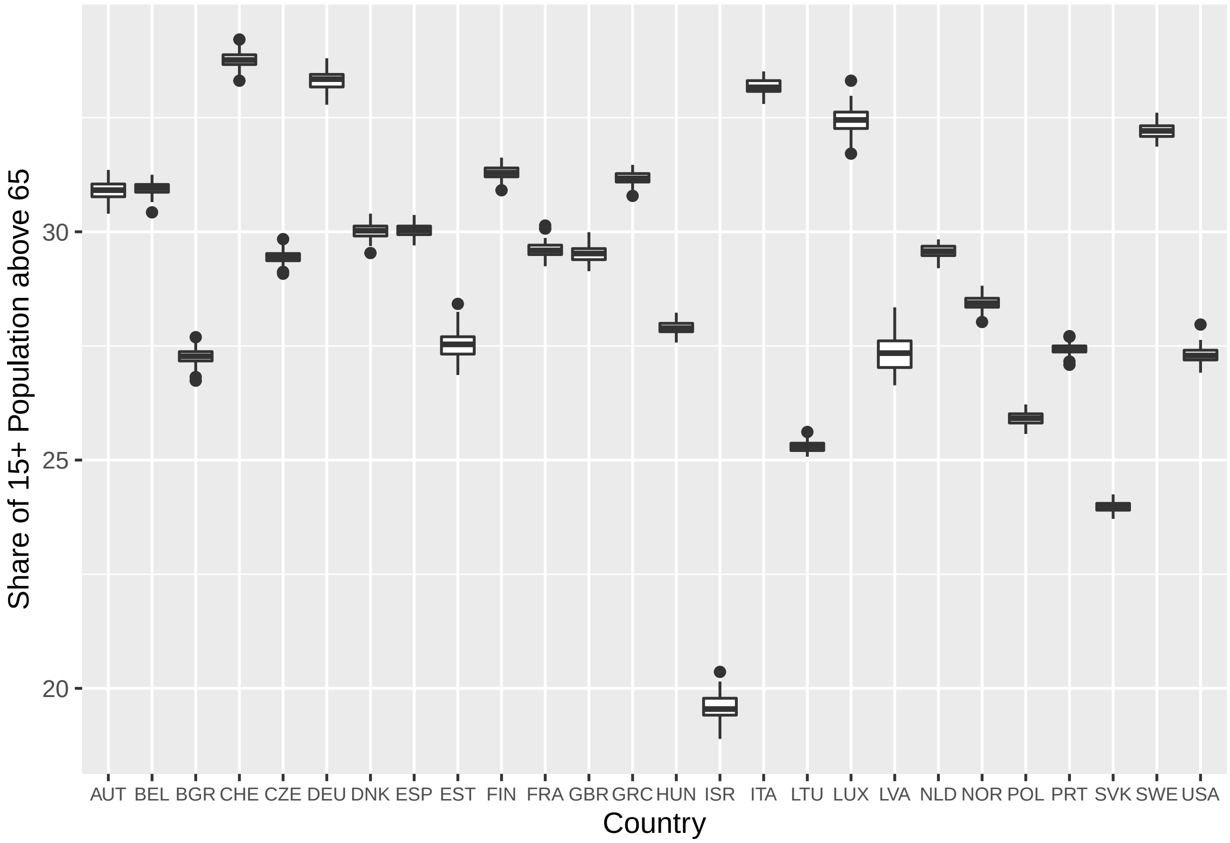
```
#Plot UNWPP populations by country
countrylist <- death_rates %>%
  mutate(ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c')) %>%
  pull(ccode)

countrylist <- unique(countrylist)
```

```
#Load UNWPP data
unwpp_data <- fread("https://population.un.org/wpp/Download/Files/1_Indicators%20(Standard)/CSV_FILES/WI
```

```
#Modify and filter data
unwpp_pop <- unwpp_data %>%
  filter(Time == 2019 & Variant == "Medium") %>%
  mutate(ccode = countrycode(Location, origin = 'country.name', destination = 'iso3c')) %>%
  filter(ccode %in% countrylist & Location %in% grep(pattern = "dependencies", x = Location, invert = T
  select(ccode, PopTotal)

unwpp_pop %>%
  ggplot() + geom_point(aes(x = ccode, y = PopTotal)) +
  labs(x = "Country", y = "Population Size: UNWPP 2019") +
  theme(axis.text.x = element_text(size = 7))
```

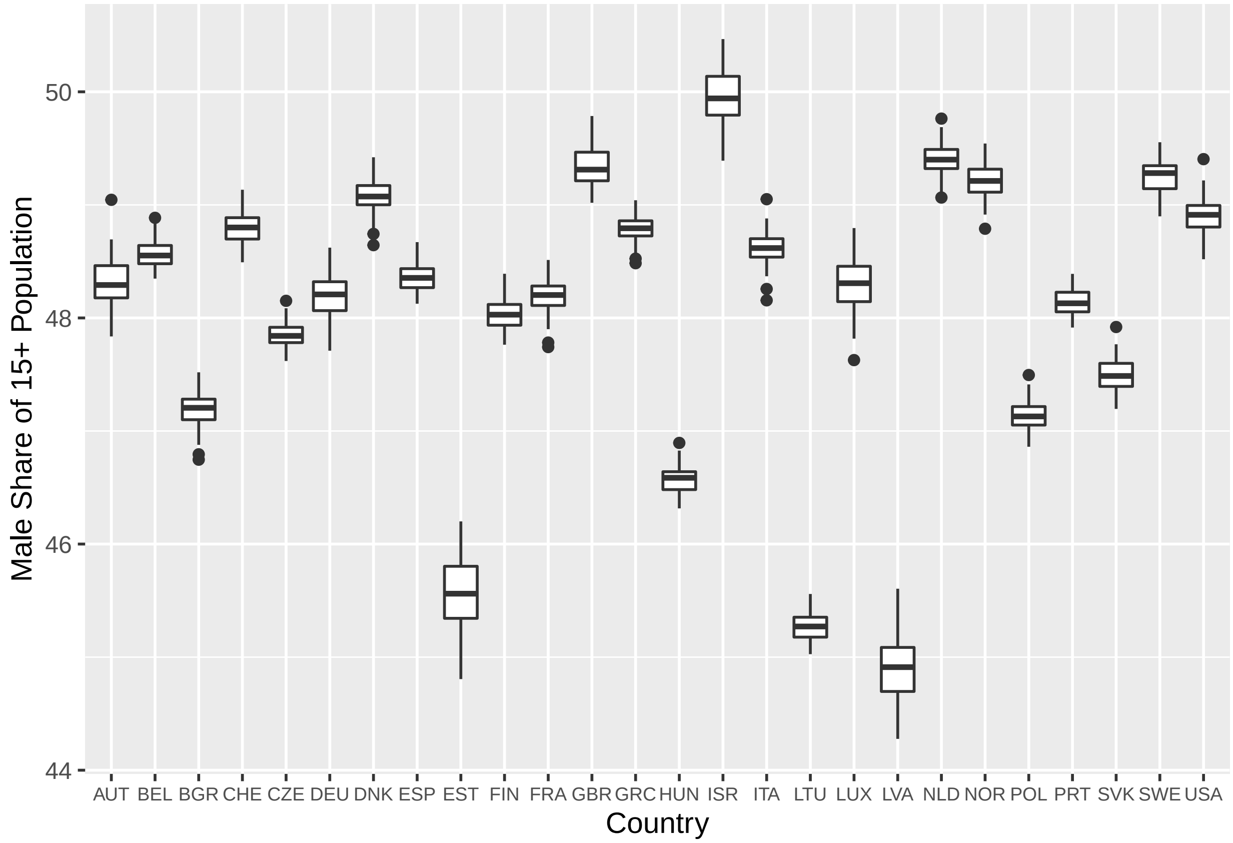


```
#Calculate above-15 age ratios
pop_ageratio <- death_rates %>%
  filter(age %in% c("15-29", "30-44", "45-64", "65+")) %>%
  mutate(above65 = if_else(age %in% c("65+"), "65plus", "below65")) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other, above65) %>%
  summarize(pop_covid = sum(n_den_covid, na.rm = T),
            pop_other = sum(n_den_other, na.rm = T)) %>%
  mutate(pop_total = (pop_covid + pop_other)/2) %>%
  select(-c(pop_covid, pop_other)) %>%
  pivot_wider(names_from = "above65", names_prefix = "pop", values_from = "pop_total") %>%
  mutate(pop_ageratio = 100*(pop65plus/(pop65plus + popbelow65))) %>%
  select(-c(pop65plus, popbelow65))


#Graph these by country
pop_ageratio %>%
 mutate(ccode = countrycode(gsub("_", " ", country),
                            origin = 'country.name', destination = 'iso3c')) %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = pop_ageratio)) +
  labs(x = "Country", y = "Share of 15+ Population above 65") +
  theme(axis.text.x = element_text(size = 7))
```



```
#Calculate above-15 sex ratio
pop_sexratio <- death_rates %>%
  filter(age %in% c("15-29", "30-44", "45-64", "65+")) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other, sex) %>%
  summarize(pop_covid = sum(n_den_covid, na.rm = T),
            pop_other = sum(n_den_other, na.rm = T)) %>%
  mutate(pop_total = (pop_covid + pop_other)/2) %>%
  select(-c(pop_covid, pop_other)) %>%
  pivot_wider(names_from = "sex", names_prefix = "pop", values_from = "pop_total") %>%
  mutate(pop_sexratio = 100*(popM/(popF+popM)),
         ccode = countrycode(gsub("_", " ", country),
                            origin = 'country.name', destination = 'iso3c')) %>%
  select(-c(popM, popF))
```

```
#Graph these by country
pop_sexratio %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = pop_sexratio)) +
  labs(x = "Country", y = "Male Share of 15+ Population") +
  theme(axis.text.x = element_text(size = 7))
```



Kinship Structure

- Size of family

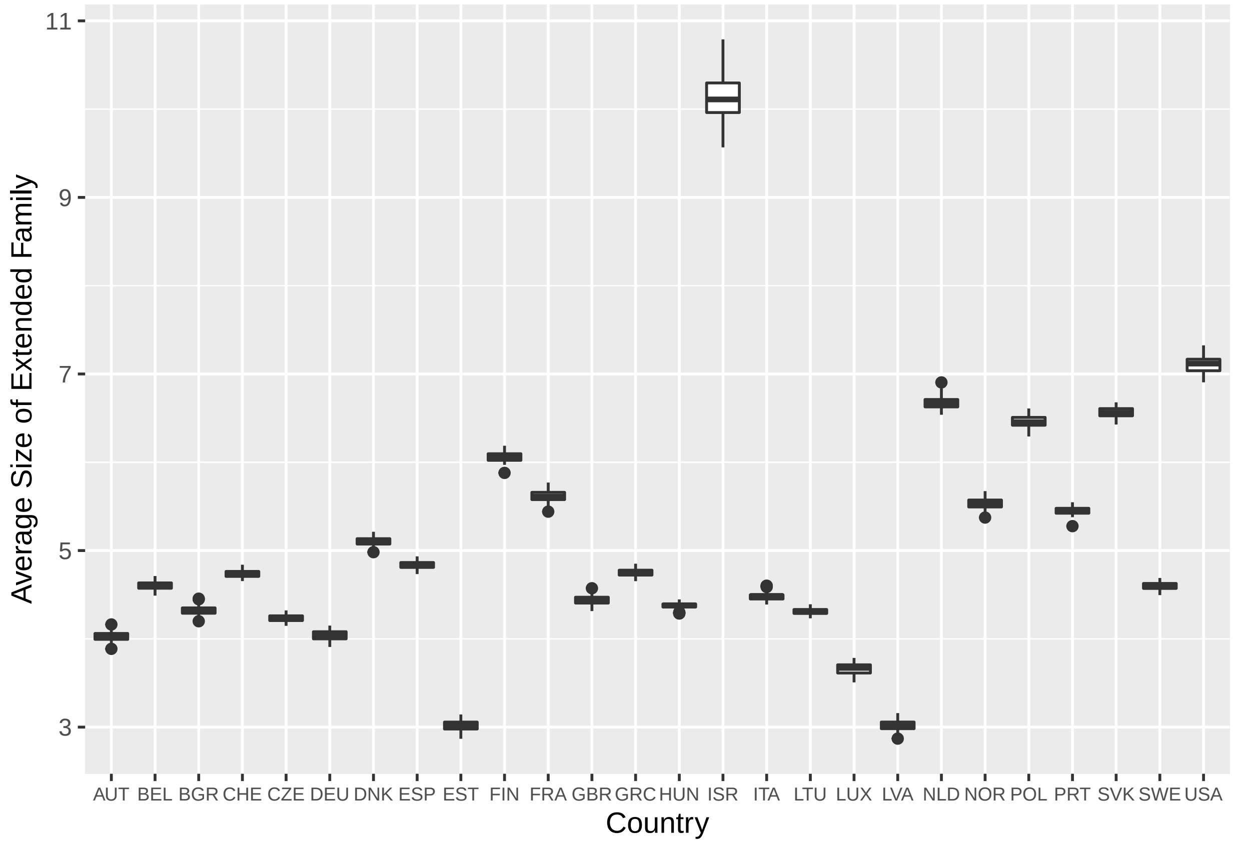
  I measure this as the average size of the extended family for individuals above 15 (due to data issues) for the average ego alive at the start of the period.

- Age ratio of relatives This is the ratio of kin above 65+ to kin between 15-64 for the extended family.
- Sex ratio of relatives This is the ratio of average number of male kin above age 15 to average number of female kin above age 15 for the nuclear family.

We calculate this using the kin_ratio object.

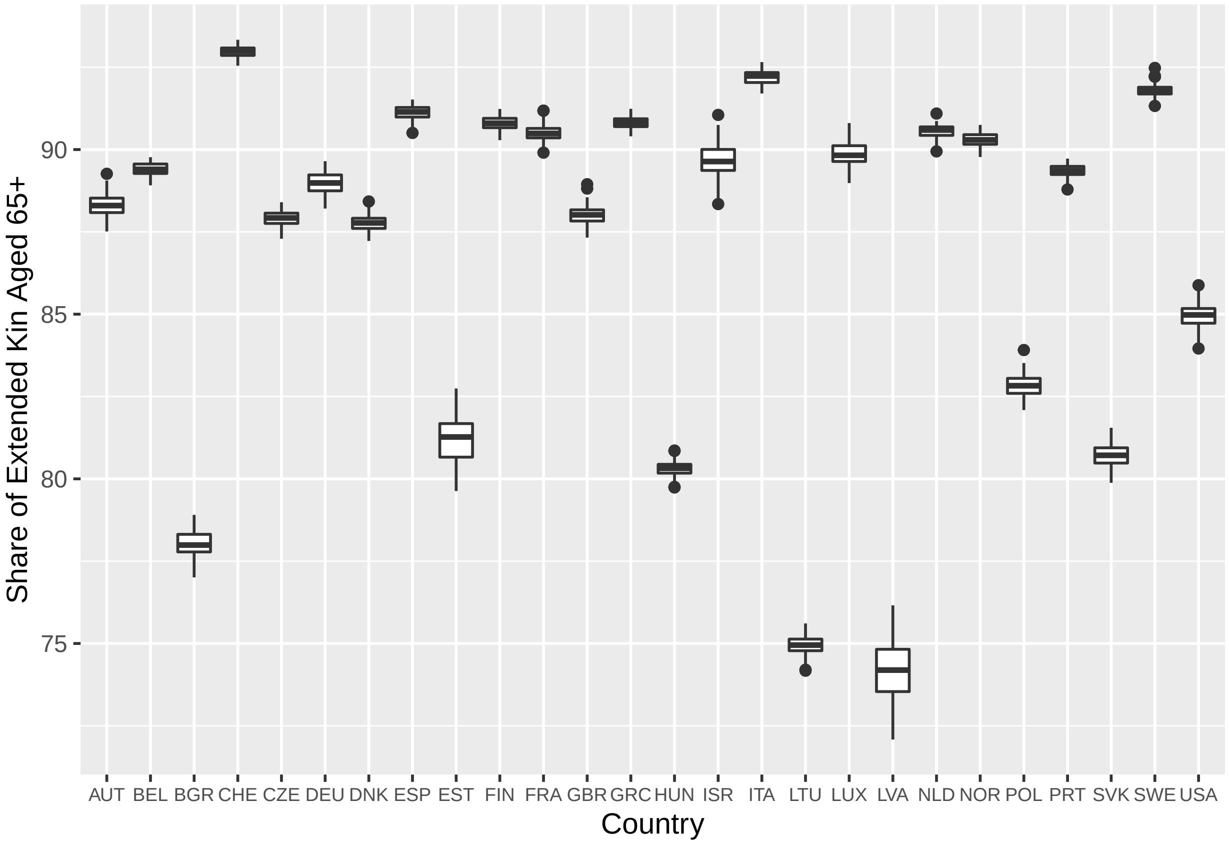
```
#Match Covid and counterfactual simulations, and clean data
kin_ratio <- data$kin_ratio %>%
  filter(!is.na(scenario)) %>%
  mutate(sex = ifelse(grepl("f1", category), "F", "M"),
  age = gsub("f[0-1]{1}", "", category),
  category = NULL) %>%
  ungroup() %>%
  filter(kintype == "extended") %>%
  complete(age, nesting(country, scenario, sim.id, sex, kintype)) %>%
  group_by(country, scenario, sex, age) %>%
  mutate(index = row_number()) %>%
  filter(index <= 99) %>% #Temporary solution for addressing matching issue
  ungroup() %>%
  pivot_wider(id_cols = c("country", "sex", "age", "index"),
              names_from = "scenario",
              values_from = c("count_all", "count_female", "count_male", "count_65plus", "count_below65
```

```
#Size of family
kin_size <- kin_ratio %>%
  mutate(count_all_ages_covid = count_all_covid*n_egos_covid,
         count_all_ages_other = count_all_other*n_egos_other) %>% group_by(country, index, sim.id_covid
  summarize(n_egos_covid = sum(n_egos_covid, na.rm = T),
            n_egos_other = sum(n_egos_other, na.rm = T),
            n_kin_covid = sum(count_all_ages_covid, na.rm=T)/n_egos_covid,
            n_kin_other = sum(count_all_ages_other, na.rm =T)/n_egos_other) %>%
  mutate(kin_size = (n_kin_covid + n_kin_other)/2,
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c')) %>%
  select(country, ccode, index, sim.id_covid, sim.id_other, kin_size)

#Graph this
kin_size %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = kin_size)) +
  labs(x = "Country", y = "Average Size of Extended Family") +
  theme(axis.text.x = element_text(size = 7))
```

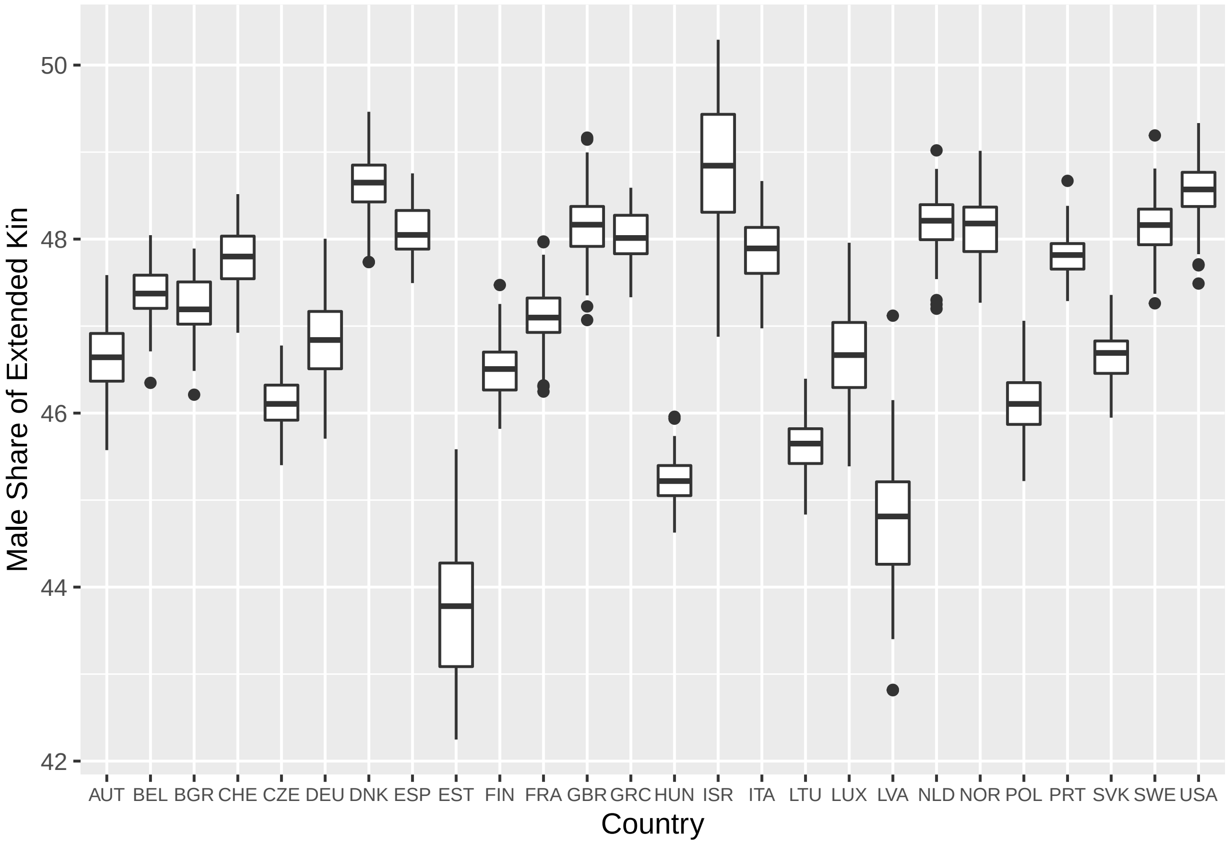


Note: I think the age ratio quantity will be heavily weighted towards older ages, which it seems to be here, based on which kin we include in our calculations.

```r
#Age ratio for kin
kin_ageratio <- kin_ratio %>%
  mutate(count_65plus_covid = count_65plus_covid*n_egos_covid,
         count_65plus_other = count_65plus_other*n_egos_other,
         count_below65_covid = count_below65_covid*n_egos_covid,
         count_below65_other= count_below65_other*n_egos_other) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other) %>%
  summarize(n_egos_covid = sum(n_egos_covid, na.rm = T),
            n_egos_other = sum(n_egos_other, na.rm = T),
            n_kin_65plus_covid = sum(count_65plus_covid, na.rm=T)/n_egos_covid,
            n_kin_65plus_other = sum(count_65plus_other, na.rm=T)/n_egos_other,
            n_kin_below65_covid = sum(count_below65_covid, na.rm=T)/n_egos_covid,
            n_kin_below65_other = sum(count_below65_other, na.rm=T)/n_egos_other) %>%
  mutate(kin_65plus = (n_kin_65plus_covid + n_kin_65plus_other)/2,
         kin_below65 = (n_kin_below65_covid + n_kin_below65_other)/2,
         kin_ageratio = 100*(kin_65plus/(kin_below65 + kin_65plus)),
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c')) %>%
  select(country, ccode, index, sim.id_covid, sim.id_other, kin_ageratio)

#Graph this
kin_ageratio %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = kin_ageratio)) +
  labs(x = "Country", y = "Share of Extended Kin Aged 65+") +
  theme(axis.text.x = element_text(size = 7))
```



```r
#Sex ratio for kin
kin_sexratio <- kin_ratio %>%
  mutate(count_female_covid = count_female_covid*n_egos_covid,
         count_female_other = count_female_other*n_egos_other,
         count_male_covid = count_male_covid*n_egos_covid,
         count_male_other= count_male_other*n_egos_other) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other) %>%
  summarize(n_egos_covid = sum(n_egos_covid, na.rm = T),
            n_egos_other = sum(n_egos_other, na.rm = T),
            n_kin_female_covid = sum(count_female_covid, na.rm=T)/n_egos_covid,
            n_kin_female_other = sum(count_female_other, na.rm=T)/n_egos_other,
            n_kin_male_covid = sum(count_male_covid, na.rm=T)/n_egos_covid,
            n_kin_male_other = sum(count_male_other, na.rm=T)/n_egos_other) %>%
  mutate(kin_female = (n_kin_female_covid + n_kin_female_other)/2,
         kin_male = (n_kin_male_covid + n_kin_male_other)/2,
         kin_sexratio = 100*(kin_male/(kin_female + kin_male)),
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c')) %>%
  select(country, ccode, index, sim.id_covid, sim.id_other, kin_sexratio)

#Graph this
kin_sexratio %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = kin_sexratio)) +
  labs(x = "Country", y = "Male Share of Extended Kin") +
  theme(axis.text.x = element_text(size = 7))
```



Dependent Variable: Excess Bereavement

There are different ways of calculating excess bereavement. Here, we use a rate consistent with what we did for excess mortality. One definition that seems fairly consistent with excess mortality is to look in the numerator at bereavement across the two different scenarios, where bereavement is measured as the difference in mean kin counts for individuals with at least one kin member of a certain type at the start of the period; in the denominator we have the population with at least one kin member alive in 2020. Here, I define the excess bereavement rate as follows, where K represents the average number of kin surviving for an ego with at least one relative alive at the start of the period, and exposure is the number of individuals with at least one relative at the start of the period who survive the Covid period, averaged across Covid and counterfactual scenarios:

$$EBR = \frac{(K_{pre,Counterfactual} - K_{post,Counterfactual}) - (K_{pre,Covid} - K_{post,Covid})}{Exposure}$$

$$EBR_possible = \frac{(K_{pre,Covid} - K_{post,Covid})}{(K_{pre,Counterfactual} - K_{post,Counterfactual})}$$

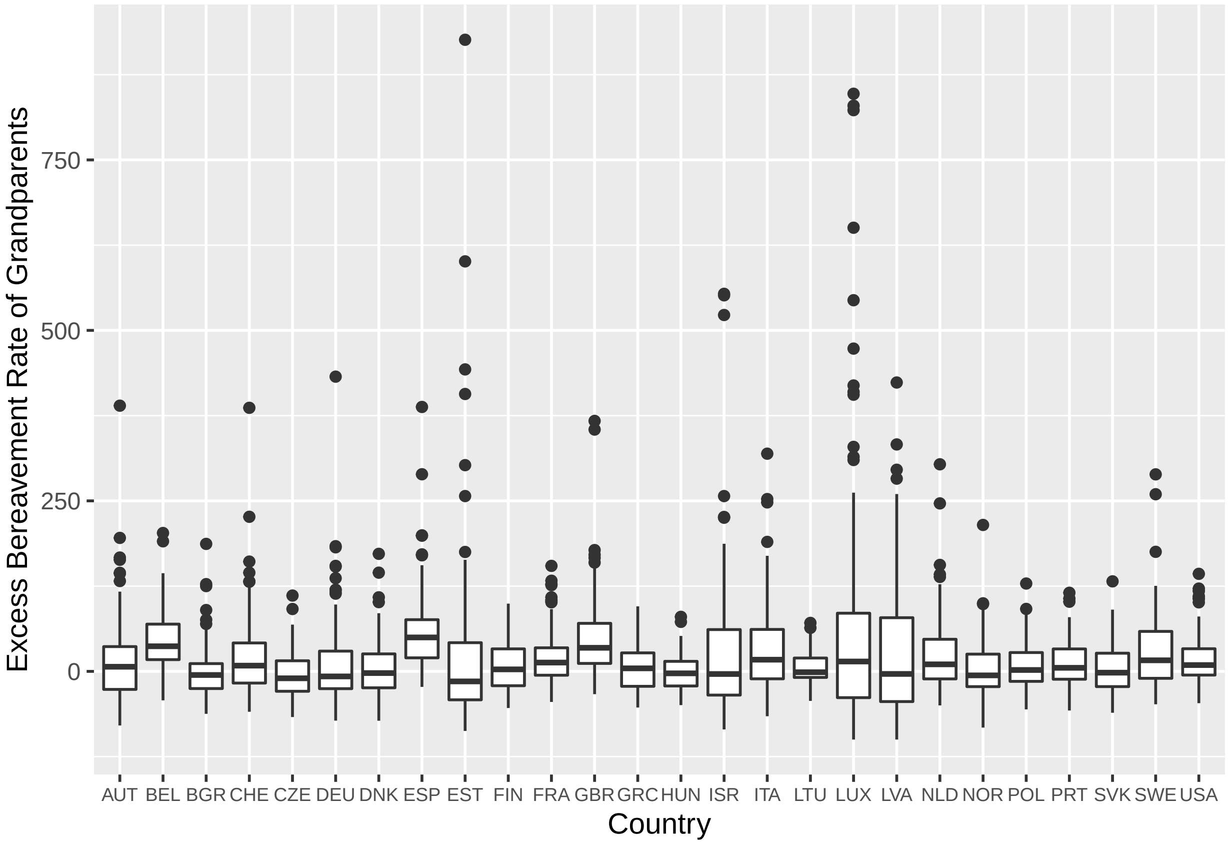
I load the kin_bb object, which is used for calculating excess bereavement. One big question is what level we want this data to be at. For now I have these numbers calculated individually for each age-sex category and kin type for each simulation (which will facilitate regression analysis later by either ego characteristics/type of kin). I only calculate this for above age 15 categories, though, due to data issues with 0-14.

```
#Calculate excess bereavement rates
kin_ebr <-data$kin_bb %>%
  filter(!is.na(scenario)) %>%
  mutate(sex = ifelse(grepl("f1", category), "F", "M"),
    age = gsub("f[0-1]{1}", "", category),
    category = NULL) %>%
  ungroup() %>%
  complete(age, nesting(country, scenario, sim.id, sex, kintype)) %>%
  group_by(country, scenario, sex, age, kintype) %>%
  mutate(index = row_number(), category = NULL) %>%
  filter(index <= 99) %>% #Temporary solution for addressing matching issue
  ungroup() %>%
  pivot_wider(id_cols = c("country", "sex", "age", "kintype", "index"),
              names_from = "scenario",
              values_from = c("mean_pre_with", "sd_pre_with", "n_withkin",
                              "mean_post_with", "sd_post_with", "sim.id")) %>%
  mutate(bereavement_covid = (mean_pre_with_covid - mean_post_with_covid),
         bereavement_other = (mean_pre_with_other - mean_post_with_other),
         n_withkin = (n_withkin_covid + n_withkin_other)/2,
         ebr = 100*(bereavement_covid)/(bereavement_other),
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c'))
```

```
kin_ebr_kintype <- kin_ebr %>%
  ungroup %>%
  mutate(sum_kin_loss_covid = bereavement_covid*n_withkin,
         sum_kin_loss_other = bereavement_other*n_withkin) %>%
  group_by(country, ccode, sim.id_covid, sim.id_other, kintype) %>%
  summarize(sum_bereavement_covid = sum(sum_kin_loss_covid, na.rm = T),
            sum_bereavement_other = sum(sum_kin_loss_other, na.rm = T),
            n_total = sum(n_withkin, na.rm = T),
            mean_bereavement_covid = sum_bereavement_covid/n_total,
            mean_bereavement_other = sum_bereavement_other/n_total) %>%
  mutate(ebr = 100*(mean_bereavement_covid - mean_bereavement_other)/mean_bereavement_other)
```

```
#Graph an example
kin_ebr_kintype %>%
  filter(kintype == "gparents") %>%
  ggplot() + geom_boxplot(aes(x = ccode, y = ebr)) +
  labs(x = "Country", y = "Excess Bereavement Rate of Grandparents") +
  theme(axis.text.x = element_text(size = 7))
```

```
## Warning: Removed 4 rows containing non-finite values (stat_boxplot).
```



Preparing the Dataset for Regression Analysis

Now, we can combine the various objects generated previously to create a dataset used in our regression analysis.

```
#Merge data objects by category
em_data <- merge(em_sexratio, merge(em_level, em_age))
pop_data <- merge(unwpp_pop, merge(pop_sexratio, pop_ageratio))
kin_data <- merge(kin_sexratio, merge(kin_size, kin_ageratio))
```

```
#And get a tibble of covariates
covariates <- merge(kin_data, merge(pop_data, em_data))
```

```
#Now merge with the dependent variable
tabledata_kintype <- merge(covariates, kin_ebr_kintype)
```

Regression Tables

```
tabledata_kintype <- tabledata_kintype %>%
  filter_all(all_vars(!is.infinite(.)))
```

```
extended_basic <- lm(ebr~em_level, subset = (kintype == "extended"), data = tabledata_kintype, na.actio

extended_mort <- lm(ebr~em_level + em_share_age + em_sexratio, subset = (kintype == "extended"), data =
extended_pop <- extended_mort <- lm(ebr~em_level + em_share_age + em_sexratio + PopTotal + pop_ageratio

extended_kinstructure <- extended_pop <- extended_mort <- lm(ebr~em_level + em_share_age + em_sexratio
```

```
stargazer(extended_basic, extended_mort, extended_pop, extended_kinstructure, colnames = c("Basic", "+M
```

```
##
## % Error: Argument 'colnames' must be NULL, or of type 'logical' (TRUE/FALSE)
## % Error: Argument 'colnames' must be of length 1.'
```

```
nuclear_basic <- lm(ebr~em_level, subset = (kintype == "nuclear"), data = tabledata_kintype, na.action

nuclear_mort <- lm(ebr~em_level + em_share_age + em_sexratio, subset = (kintype == "nuclear"), data = ta

nuclear_pop <- lm(ebr~em_level + em_share_age + em_sexratio + PopTotal + pop_ageratio + pop_sexratio, su

nuclear_kinstructure <- lm(ebr~em_level + em_share_age + em_sexratio + PopTotal + pop_ageratio + pop_se
```

```
stargazer(nuclear_basic, nuclear_mort, nuclear_pop, nuclear_kinstructure, colnames = c("Basic", "+Mortal
```

```
##
## % Error: Argument 'colnames' must be NULL, or of type 'logical' (TRUE/FALSE)
## % Error: Argument 'colnames' must be of length 1.'
```

```
gparents_basic <- lm(ebr~em_level, subset = (kintype == "gparents"), data = tabledata_kintype, na.actio

gparents_mort <- lm(ebr~em_level + em_share_age + em_sexratio, subset = (kintype == "gparents"), data =

gparents_pop <- lm(ebr~em_level + em_share_age + em_sexratio + PopTotal + pop_ageratio + pop_sexratio,

gparents_kinstructure <- lm(ebr~em_level + em_share_age + em_sexratio + PopTotal + pop_ageratio + pop_s
```

```
stargazer(gparents_basic, gparents_mort, gparents_pop, gparents_kinstructure, colnames = c("Basic", "+M
```

```
##
## % Error: Argument 'colnames' must be NULL, or of type 'logical' (TRUE/FALSE)
## % Error: Argument 'colnames' must be of length 1.'
```

```
children_basic <- lm(ebr~em_level, subset = (kintype == "children"), data = tabledata_kintype, na.actio

children_mort <- lm(ebr~em_level + em_share_age + em_sexratio, subset = (kintype == "children"), data =

children_pop <- lm(ebr~em_level + em_share_age + em_sexratio + PopTotal + pop_ageratio + pop_sexratio,

children_kinstructure <- lm(ebr~em_level + em_share_age + em_sexratio + PopTotal + pop_ageratio + pop_s
```

```
stargazer(children_basic, children_mort, children_pop, children_kinstructure, colnames = c("Basic", "+M
```

```
##
## % Error: Argument 'colnames' must be NULL, or of type 'logical' (TRUE/FALSE)
## % Error: Argument 'colnames' must be of length 1.'
```

```
stargazer(extended_kinstructure, nuclear_kinstructure, gparents_kinstructure, children_kinstructure, co
```

```
##
## % Error: Argument 'colnames' must be NULL, or of type 'logical' (TRUE/FALSE)
## % Error: Argument 'colnames' must be of length 1.'
```