

Excess Bereavement and Excess Mortality: Some Analysis

Contents

Independent Variables	1
Dependent Variable: Excess Bereavement	13
Preparing the Dataset for Regression Analysis	15
Regression Tables	15
Other Analysis: Probability of Losing Kin	37
Plots for Presentation	48

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)
source("~/covid_simulation/Kin_death/01_load_functions.R")
```

```
## [1] "library2: tidyverse loaded."
## [1] "library2: scales loaded."
## [1] "library2: patchwork loaded."
## [1] "library2: data.table loaded."
## [1] "library2: parallel loaded."
## [1] "library2: knitr loaded."
```

```
source('~/.covid_simulation/Kin_death/functions_bereavement.R')
```

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 ratio here (am I remembering this right?). Most quantities in this analysis will go from 0-100. Where number of deaths in a particular category are denoted by D ,

- Level of above-15 excess mortality:

$$EM = 100 \times \frac{D_{15+,Covid} - D_{15+,Counterfactual}}{D_{15+,Counterfactual}}$$

- Age ratio for above-15 excess mortality:

What share of excess deaths come from the age 65+ category:

$$Share_{65+} = 100 \times \frac{D_{65+,Covid} - D_{65+,Counterfactual}}{D_{15+,Covid} - D_{15+,Counterfactual}}$$

- Sex ratio for above-15 excess mortality: Here, we measure the ratio of excess deaths for men to excess deaths for women.

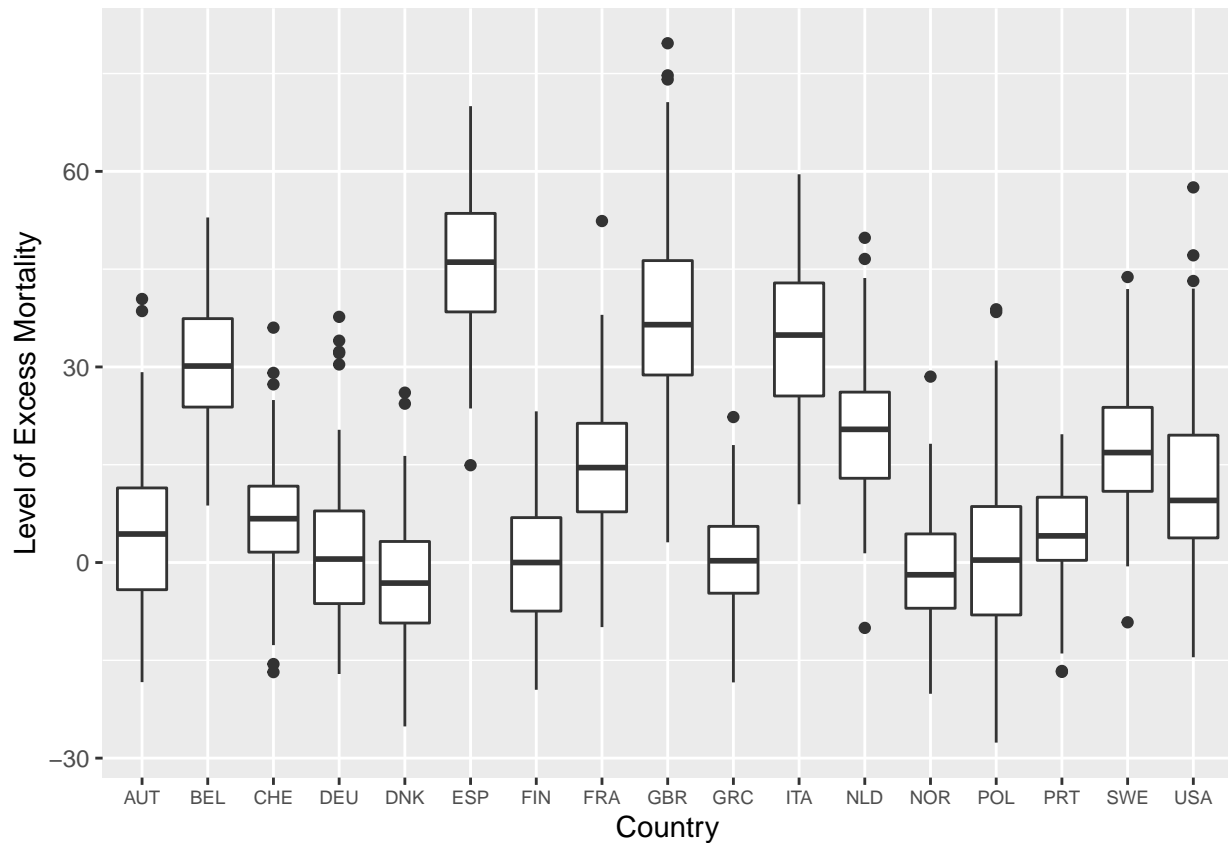
$$Share_{Male} = 100 \times \frac{D_{15+,Covid,Male} - D_{15+,Counterfactual,Male}}{D_{15+,Covid,Both} - D_{15+,Counterfactual,Both}}$$

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))
```



```
#Summarize it
em_level %>%
  ungroup() %>%
  group_by(country) %>%

  summarize(mean_em_level = mean(em_level, na.rm = T))
```

```
## # A tibble: 17 x 2
##   country                mean_em_level
##   <chr>                  <dbl>
## 1 Austria                 5.15
## 2 Belgium                31.4
## 3 Denmark                -2.65
## 4 Finland                 0.257
## 5 France                 15.1
## 6 Germany                 1.49
## 7 Greece                  0.681
## 8 Italy                   34.6
## 9 Netherlands            20.2
## 10 Norway                -1.08
## 11 Poland                 1.42
## 12 Portugal               4.85
## 13 Spain                  45.9
## 14 Sweden                 17.5
## 15 Switzerland            6.98
## 16 United_Kingdom         37.7
## 17 United_States_of_America 11.5
```

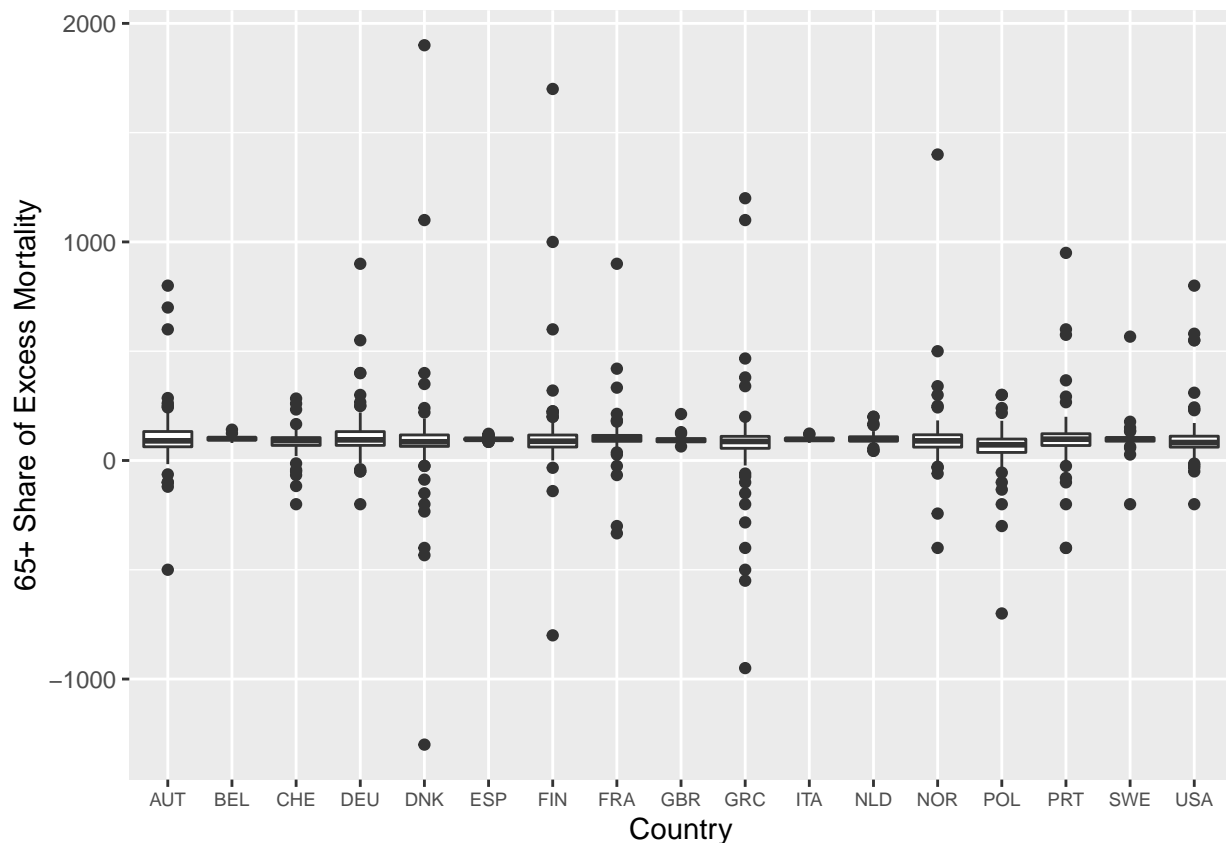
```

#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_covid_65plus),
         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 12 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),

```

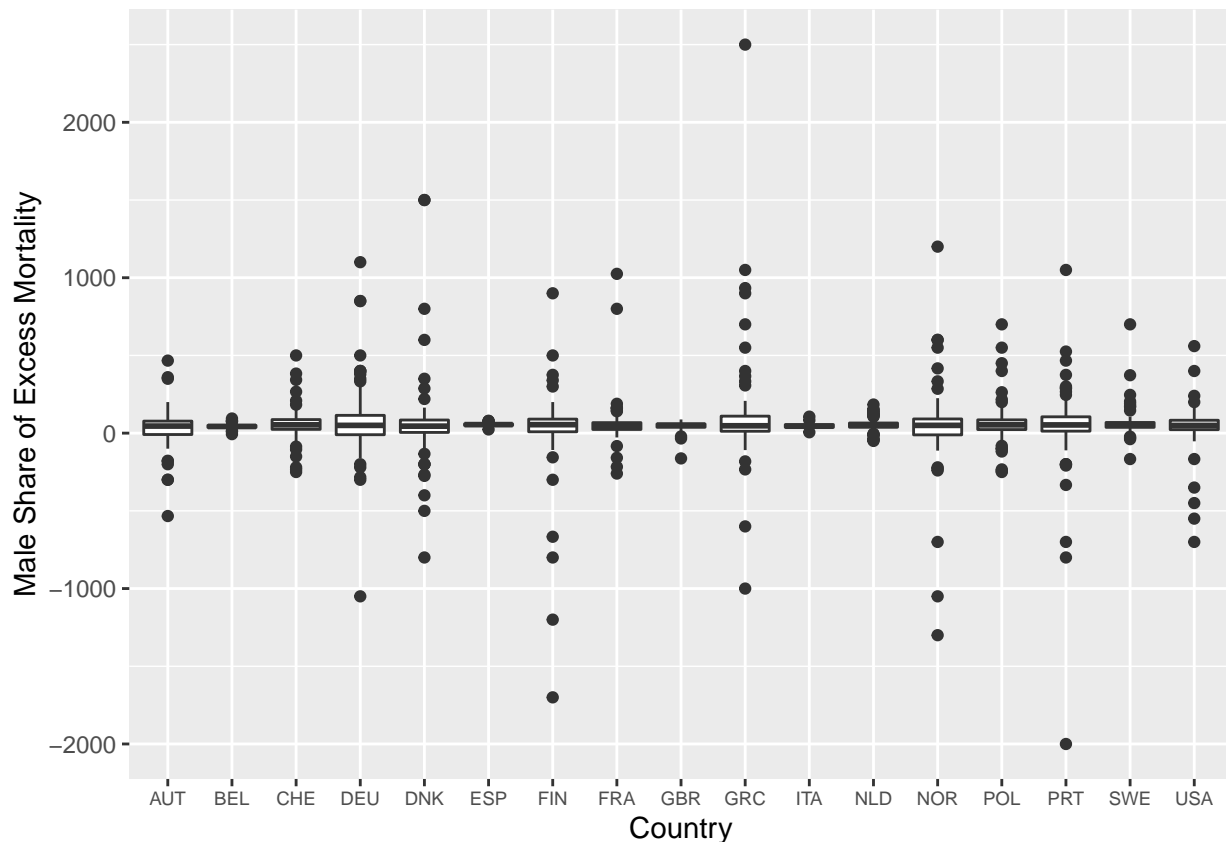
```

    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
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 12 rows containing non-finite values (stat_boxplot).
```



QUESTION: We see a lot of fluctuation in the age ratio and sex ratio measures for excess mortality, because the numbers are quite low and in some cases the number of excess deaths will be in fact negative. Do you have any thoughts on whether this is an issue? Should we be defining these quantities differently?

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. 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.

QUESTION: Should these ratios be shares as well? Or is that not necessary? I think it might be fine to leave them as is, as they are standard demographic quantities, but I'm trying to remember what we agreed on. For now I've calculated

- Share of population above 15 that is 65+:

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

$$Share_{65+} = 100 \times \frac{N_{65+}}{N_{15+}}$$

- Share of population above 15 that is female:

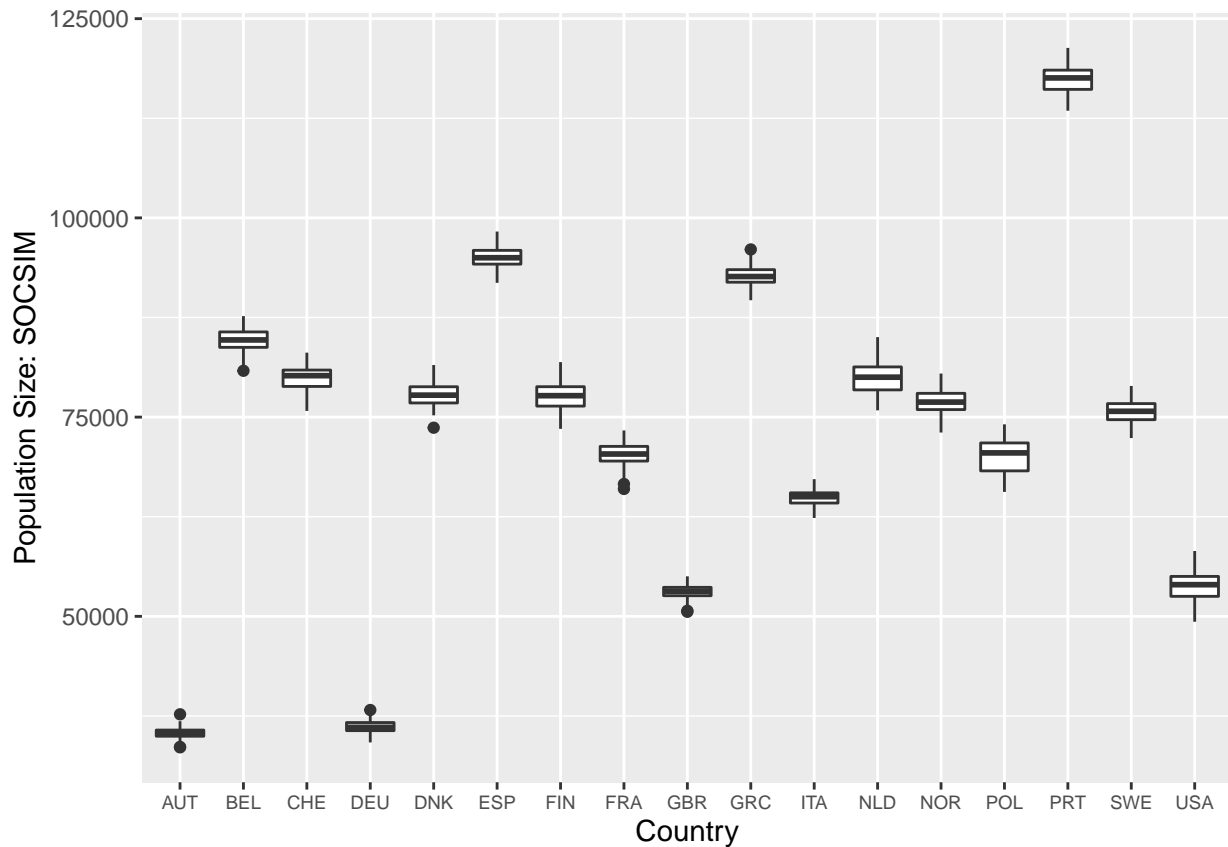
Similarly, this is a share,

$$Share_{Male} = 100 \times \frac{N_{Male}}{N_{Female} + N_{Male}}$$

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))
```



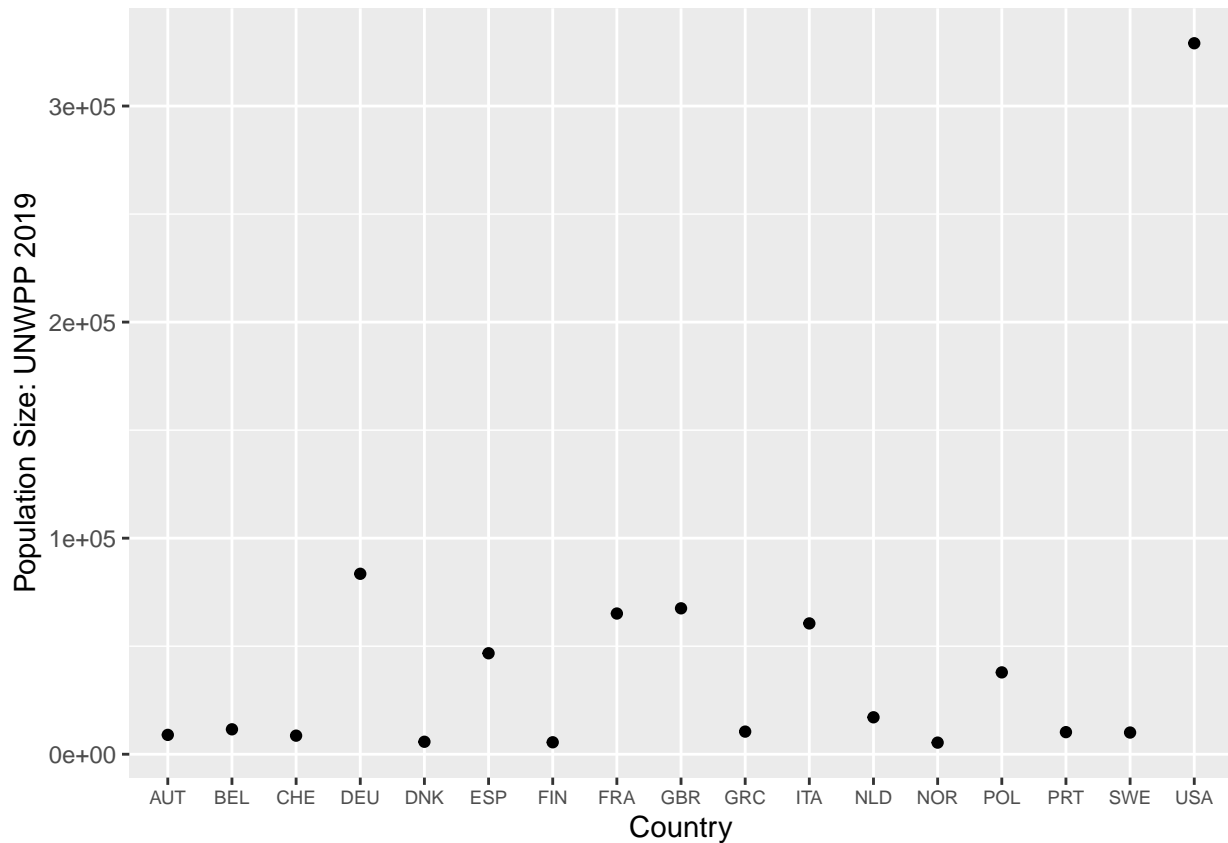
```
#Plot UNWPP populations by country
countrylist <- death_rates %>%
  mutate(ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c')) %>%
  pull(ccode)

countrylist <- unique(countrylist)

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

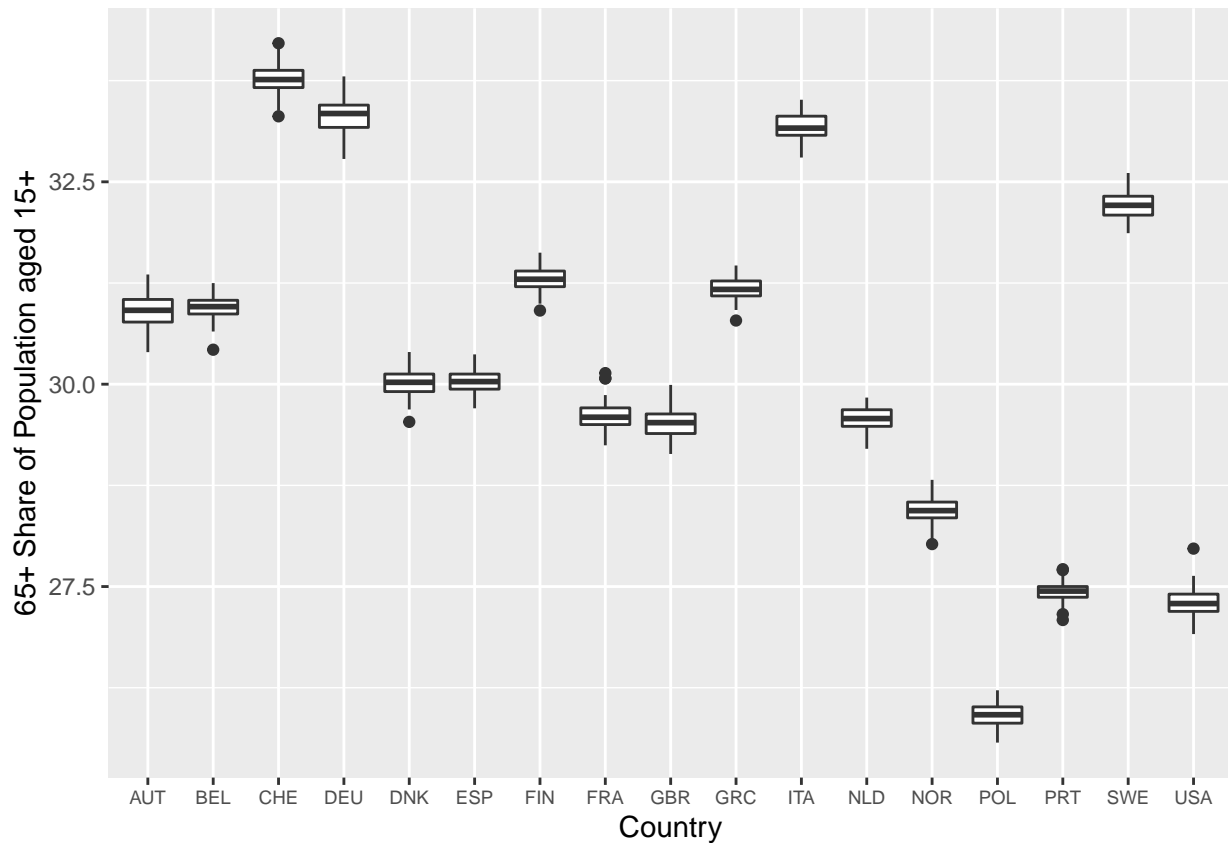
#Modify and filter data
unwpp_pop <- unwpp_data %>%
  filter(Time == 2019 & Variant == "Medium") %>%
  mutate(ccode = countrycode(Location, origin = 'country.name', destination = 'iso3c'),
         unwpp_pop = PopTotal) %>%
  filter(ccode %in% countrylist & Location %in% grep(pattern = "dependencies",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

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



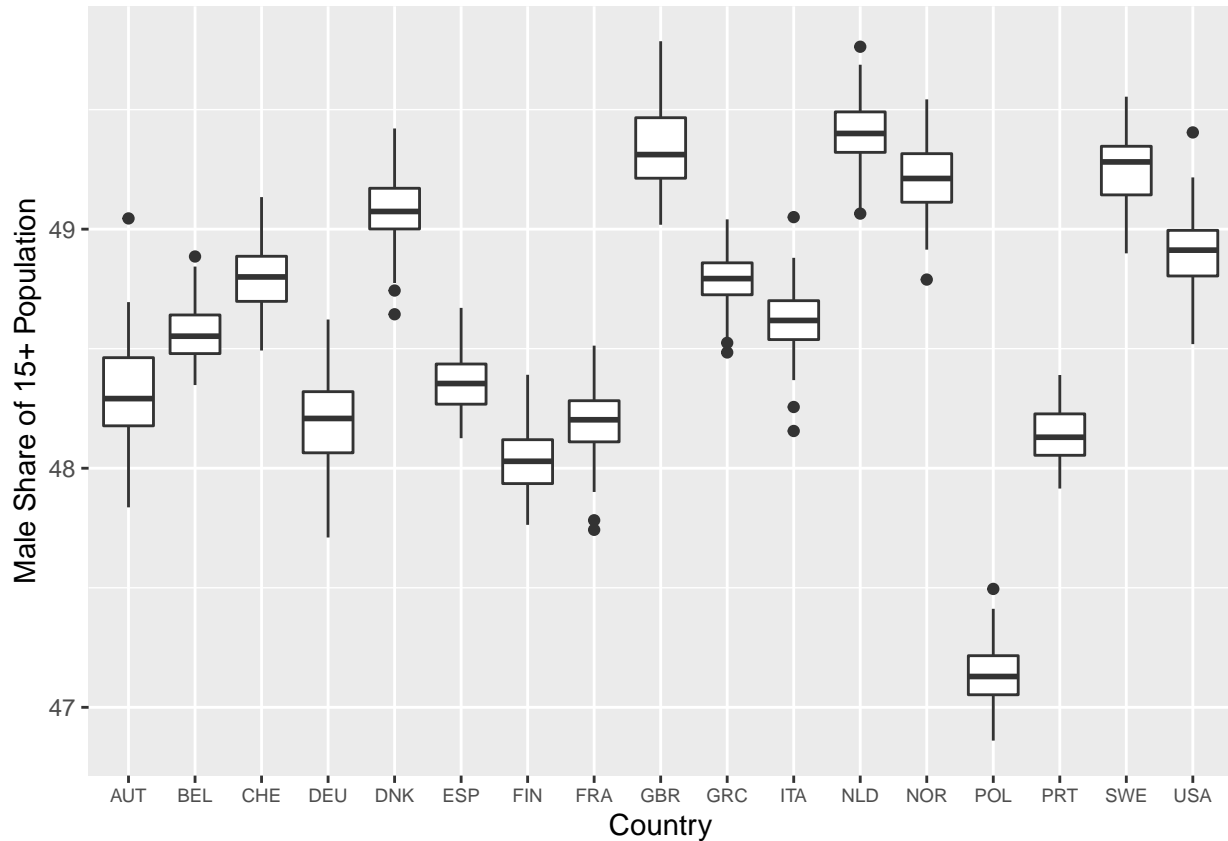
```
#Calculate above-15 age ratios
pop_ageratio <- death_rates %>%
  filter(age %in% c("15-29", "30-44", "45-64", "65+")) %>%
  mutate(above65 = if_else(age %in% c("65+"), "65plus", "below65")) %>%
  ungroup() %>%
  group_by(country, index, sim.id_covid, sim.id_other, above65) %>%
  summarize(pop_covid = 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 = "65+ Share of Population aged 15+") +
  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 nuclear family for individuals above 15 (due to data issues) for the average ego alive at the start of the period. Think about the next two quantities (not calculated for now):

- Age ratio of relatives This is the share of kin above 65+ to kin above 15 for the extended family.
- Sex ratio of relatives This is the share of male kin above age 15 of total kin above age 15 for the extended 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 == "nuclear") %>%
  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() %>%
```

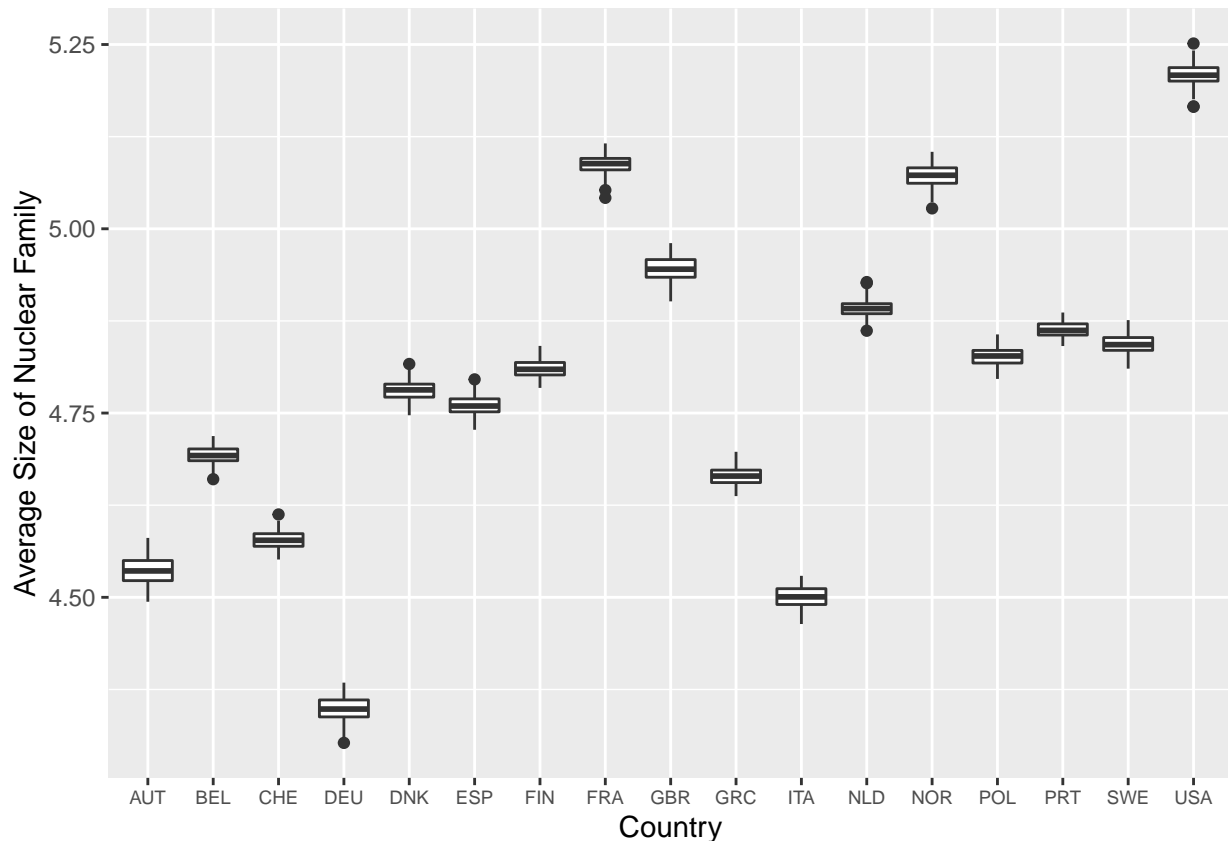
```

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 Nuclear 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.

```

#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))

#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 relative difference consistent with what we did for excess mortality. One definition that seems fairly consistent with excess mortality is to look 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. Here, I define the excess bereavement, where K represents the average number of kin surviving for an ego with at least one relative alive at the start of the period, as follows:

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

One thing we talked about a bit in some of the early work around the paper was the probability of losing a type of relative; I don't think that's what we're really measuring here, so we may need some alternative analysis to get at that. QUESTION: Am I remembering our discussion on this quantity correctly?

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). Later for the table, though, I think I will need to aggregate across age categories to address all the NA values.

```
#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_within",
                             "mean_post_with", "sd_post_with", "sim.id", "n_losekin", "n_total")) %>%
  mutate(bereavement_covid = (mean_pre_with_covid - mean_post_with_covid),
         bereavement_other = (mean_pre_with_other - mean_post_with_other),
         n_within = (n_within_covid + n_within_other)/2,
         ebr = 100*(bereavement_covid)/(bereavement_other),
         ccode = countrycode(gsub("_", " ", country),
                             origin = 'country.name', destination = 'iso3c'))
```

Here, I create an aggregated version of this dataset, aggregating across age and sex by kin type and simulation. This is useful because otherwise we have a large number of NA values (eg. no children for 0-14, etc.). This creates issues with the regression, and also prevents kin losses from being weighted appropriately by the number of egos actually in a category.

```
kin_ebr_kintype <- kin_ebr %>%
  ungroup %>%
  mutate(sum_kin_loss_covid = bereavement_covid*n_within,
         sum_kin_loss_other = bereavement_other*n_within) %>%
  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_within, 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)

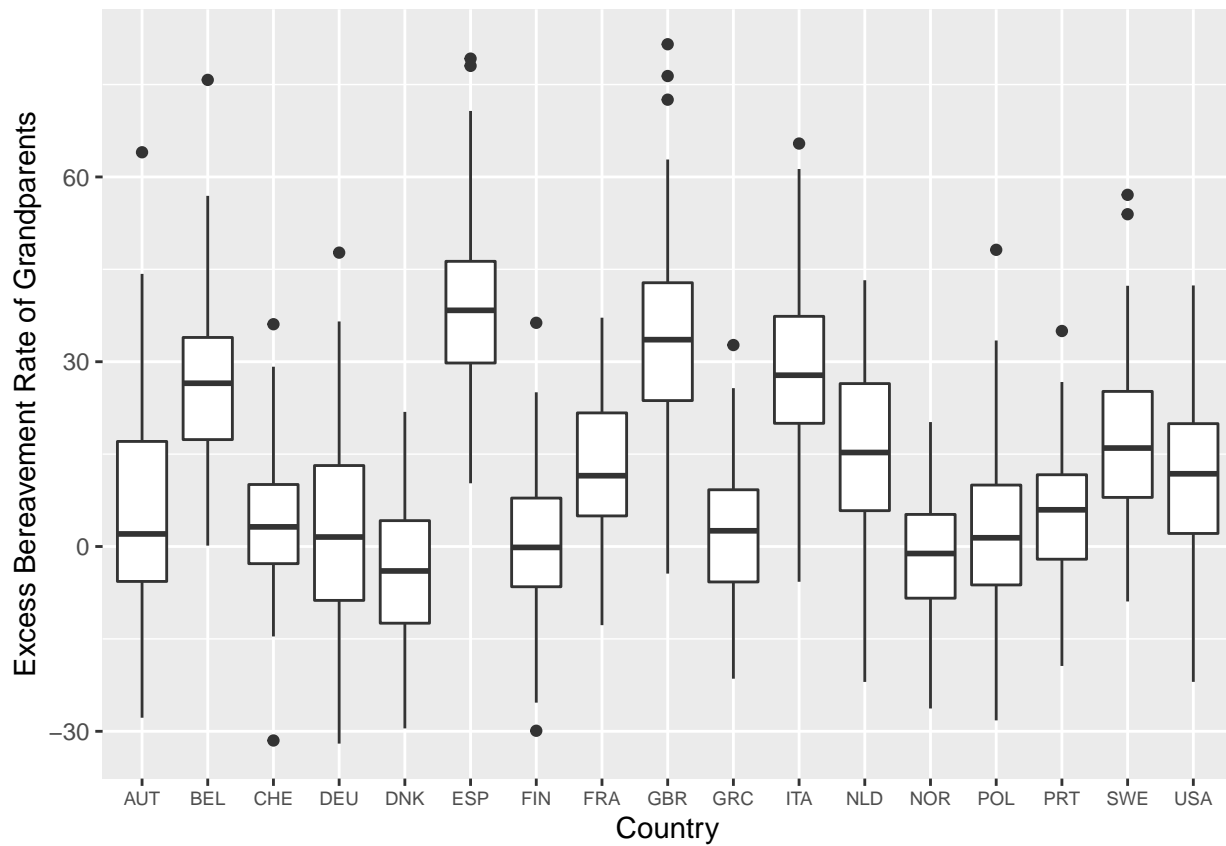
```

#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))

```



```

kin_ebr_kintype %>%
  filter(kintype == "gparents") %>%
  ungroup() %>%
  group_by(country) %>%
  summarize(mean_ebr = mean(ebr, na.rm = T))

```

```

## # A tibble: 17 x 2
##   country      mean_ebr
##   <chr>      <dbl>
## 1 Austria      5.47
## 2 Belgium     25.9
## 3 Denmark     -4.01
## 4 Finland      0.933
## 5 France      12.6

```

```
## 6 Germany                2.02
## 7 Greece                  2.14
## 8 Italy                   28.1
## 9 Netherlands            16.0
## 10 Norway                 -1.69
## 11 Poland                 2.85
## 12 Portugal               5.35
## 13 Spain                  39.4
## 14 Sweden                 17.4
## 15 Switzerland           4.03
## 16 United_Kingdom         34.0
## 17 United_States_of_America 11.4
```

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))
kin_data <- kin_size
```

```
#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

```
#A quick and clumsy way to get rid of the cases where we divide by 0--mostly an issue for great grandpa
tabledata_kintype <- tabledata_kintype %>%
  filter_all(all_vars(!is.infinite(.)))
```

Now, I fit a number of different regression models, by different types of kin.

```
nuclear_basic <- lm(ebr~em_level + country, subset = (kintype == "nuclear"), data = tabledata_kintype, na.rm=T)
nuclear_mort <- lm(ebr~em_level + country + em_share_age + em_sexratio, subset = (kintype == "nuclear"), data = tabledata_kintype, na.rm=T)
nuclear_pop <- lm(ebr~em_level + country + em_share_age + em_sexratio + unwpp_pop + pop_ageratio + pop_sexratio, subset = (kintype == "nuclear"), data = tabledata_kintype, na.rm=T)
nuclear_kinstructure <- lm(ebr~em_level + country + em_share_age + em_sexratio + unwpp_pop + pop_ageratio + pop_sexratio, subset = (kintype == "nuclear"), data = tabledata_kintype, na.rm=T)
stargazer(nuclear_basic, nuclear_mort, nuclear_pop, nuclear_kinstructure, column.labels = c("Basic", "Mort", "Pop", "Kinstructure"))
```

Nuclear Family

Dependent variable:

ebr

Basic

+Mortality
+Population
+Kinship Structure

(1)

(2)

(3)

(4)

em_level

0.651***

0.651***

0.651***

0.651***

(0.017)

(0.017)

(0.017)

(0.017)

countryBelgium

-2.341**

-2.338**

-3.125**

-8.322***

(1.168)

(1.168)

(1.212)

(2.983)

countryDenmark

-4.752***

-4.777***

-7.884***

-14.612***

(1.089)

(1.089)

(2.015)

(4.064)

countryFinland

-2.411**

-2.396**
 -1.222
 -10.636**
 (1.083)
 (1.084)
 (1.285)
 (5.103)
 countryFrance
 -2.662**
 -2.661**
 -3.138*
 -18.793**
 (1.094)
 (1.094)
 (1.875)
 (8.424)
 countryGermany
 -0.177
 -0.163
 1.810
 4.022
 (1.082)
 (1.083)
 (3.152)
 (3.356)
 countryGreece
 -3.327***
 -3.377***
 -4.733***
 -9.469***
 (1.080)
 (1.083)
 (1.250)
 (2.781)
 countryItaly
 -0.641

-0.641
 -0.104
 -2.769
 (1.190)
 (1.191)
 (2.916)
 (3.232)
 countryNetherlands
 -1.538
 -1.541
 -5.975**
 -15.672***
 (1.108)
 (1.108)
 (2.710)
 (5.763)
 countryNorway
 -3.181***
 -3.186***
 -7.738**
 -21.367***
 (1.088)
 (1.089)
 (3.677)
 (8.039)
 countryPoland
 -2.940***
 -3.016***
 -2.488
 -3.239
 (1.085)
 (1.088)
 (5.853)
 (5.861)
 countryPortugal
 -0.725

-0.725
 -2.480
 -7.283
 (1.080)
 (1.080)
 (4.258)
 (4.944)
 countrySpain
 0.311
 0.314
 -0.438
 -6.256*
 (1.285)
 (1.286)
 (1.674)
 (3.480)
 countrySweden
 2.296**
 2.291**
 0.086
 -12.337*
 (1.098)
 (1.099)
 (2.036)
 (6.827)
 countrySwitzerland
 -3.300***
 -3.330***
 -2.988
 -9.198*
 (1.081)
 (1.081)
 (3.510)
 (4.787)
 countryUnited_Kingdom
 3.276***

3.272***
 -1.012
 -12.343*
 (1.214)
 (1.214)
 (2.748)
 (6.548)
 countryUnited_States_of_America
 2.328**
 2.329**
 -1.980
 -17.997*
 (1.086)
 (1.086)
 (4.775)
 (9.663)
 em_share_age
 -0.002
 -0.002
 -0.001
 (0.001)
 (0.001)
 (0.001)
 em_sexratio
 0.0001
 0.0001
 0.0002
 (0.001)
 (0.001)
 (0.001)
 unwpp_pop
 pop_ageratio
 -0.668
 0.976
 (1.208)
 (1.483)

```

pop_sexratio
3.261**
3.620***
(1.352)
(1.364)
kin_size
32.382*
(16.988)
Constant
2.235***
2.394***
-134.541**
-349.570***
(0.769)
(0.783)
(64.966)
(130.153)
Observations
1,671
1,671
1,671
1,671
R2
0.750
0.751
0.752
0.752
Adjusted R2
0.748
0.748
0.748
0.749
Residual Std. Error
7.560 (df = 1653)
7.562 (df = 1651)
7.553 (df = 1649)

```

7.547 (df = 1648)

F Statistic

292.482*** (df = 17; 1653)

261.644*** (df = 19; 1651)

237.555*** (df = 21; 1649)

227.284*** (df = 22; 1648)

Note:

$p < 0.1$; $p < 0.05$; $p < 0.01$

```
gparents_basic <- lm(ebr~em_level + country, subset = (kintype == "gparents"), data = tabledata_kintype
gparents_mort <- lm(ebr~em_level + country + em_share_age + em_sexratio, subset = (kintype == "gparents
gparents_pop <- lm(ebr~em_level + country + em_share_age + em_sexratio + unwpp_pop + pop_ageratio + pop
gparents_kinstructure <- lm(ebr~em_level + country + em_share_age + em_sexratio + unwpp_pop + pop_agera
stargazer(gparents_basic, gparents_mort, gparents_pop, gparents_kinstructure, column.labels = c("Basic"
```

Grandparents

Dependent variable:

ebr

Basic

+Mortality

+Population

+Kinship Structure

(1)

(2)

(3)

(4)

em__level

0.639***

0.639***

0.640***

0.640***

(0.025)

(0.025)

(0.025)

(0.025)

countryBelgium

3.640**

3.630**
 3.168*
 7.907*
 (1.686)
 (1.685)
 (1.751)
 (4.311)
 countryDenmark
 -4.686***
 -4.616***
 -5.804**
 0.331
 (1.571)
 (1.571)
 (2.911)
 (5.872)
 countryFinland
 -1.279
 -1.322
 -0.910
 7.674
 (1.564)
 (1.563)
 (1.855)
 (7.374)
 countryFrance
 0.628
 0.621
 1.109
 15.385
 (1.579)
 (1.578)
 (2.707)
 (12.173)
 countryGermany
 -1.386

-1.430
 -1.747
 -3.765
 (1.562)
 (1.562)
 (4.551)
 (4.850)
 countryGreece
 -0.505
 -0.369
 -1.269
 3.049
 (1.559)
 (1.562)
 (1.806)
 (4.018)
 countryItaly
 3.774**
 3.774**
 2.749
 5.179
 (1.718)
 (1.717)
 (4.211)
 (4.670)
 countryNetherlands
 0.906
 0.911
 -0.746
 8.096
 (1.599)
 (1.598)
 (3.914)
 (8.328)
 countryNorway
 -3.334**

-3.322**
 -4.392
 8.037
 (1.571)
 (1.570)
 (5.311)
 (11.617)
 countryPoland
 -0.169
 0.046
 3.214
 3.899
 (1.566)
 (1.568)
 (8.452)
 (8.470)
 countryPortugal
 0.161
 0.160
 1.203
 5.583
 (1.559)
 (1.558)
 (6.148)
 (7.145)
 countrySpain
 7.825***
 7.813***
 7.905***
 13.210***
 (1.855)
 (1.854)
 (2.417)
 (5.029)
 countrySweden
 4.018**

4.028**
 2.066
 13.394
 (1.585)
 (1.584)
 (2.940)
 (9.866)
 countrySwitzerland
 -2.626*
 -2.541
 -4.016
 1.647
 (1.560)
 (1.559)
 (5.068)
 (6.917)
 countryUnited_Kingdom
 7.680***
 7.689***
 6.144
 16.478*
 (1.752)
 (1.751)
 (3.969)
 (9.463)
 countryUnited_States_of_America
 1.865
 1.861
 1.592
 16.197
 (1.567)
 (1.566)
 (6.896)
 (13.964)
 em_share_age
 0.004**

0.004**
0.004**
(0.002)
(0.002)
(0.002)
em_sexratio
-0.0002
-0.0002
-0.0002
(0.001)
(0.001)
(0.001)
unwpp_pop
pop_ageratio
0.213
-1.285
(1.744)
(2.143)
pop_sexratio
1.782
1.455
(1.953)
(1.971)
kin_size
-29.528
(24.549)
Constant
2.213**
1.756
-90.916
105.165
(1.110)
(1.129)
(93.817)
(188.078)
Observations

1,671

1,671

1,671

1,671

R2

0.638

0.640

0.640

0.640

Adjusted R2

0.635

0.635

0.635

0.635

Residual Std. Error

10.914 (df = 1653)

10.905 (df = 1651)

10.908 (df = 1649)

10.906 (df = 1648)

F Statistic

171.703*** (df = 17; 1653)

154.164*** (df = 19; 1651)

139.445*** (df = 21; 1649)

133.208*** (df = 22; 1648)

Note:

$p < 0.1$; $p < 0.05$; $p < 0.01$

```
children_basic <- lm(ebr~em_level + country, subset = (kintype == "children"), data = tabledata_kintype
```

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

```
children_pop <- lm(ebr~em_level + country + em_share_age + em_sexratio + unwpp_pop + pop_ageratio + pop
```

```
children_kinstructure <- lm(ebr~em_level + country + em_share_age + em_sexratio + unwpp_pop + pop_agera
```

```
stargazer(children_basic, children_mort, children_pop, children_kinstructure, column.labels = c("Basic"
```

Children

Dependent variable:

ebr

Basic

+Mortality
+Population
+Kinship Structure

(1)

(2)

(3)

(4)

em_level

0.705***

0.700***

0.701***

0.700***

(0.072)

(0.071)

(0.072)

(0.071)

countryBelgium

-40.192***

-40.153***

-41.116***

-58.837***

(4.863)

(4.849)

(5.038)

(12.402)

countryDenmark

-18.601***

-18.920***

-27.019***

-49.962***

(4.532)

(4.521)

(8.376)

(16.894)

countryFinland

-11.493**

-11.302**
 -7.992
 -40.094*
 (4.510)
 (4.498)
 (5.338)
 (21.213)
 countryFrance
 -21.912***
 -21.891***
 -27.665***
 -81.052**
 (4.553)
 (4.541)
 (7.791)
 (35.021)
 countryGermany
 -3.413
 -3.232
 9.154
 16.699
 (4.504)
 (4.495)
 (13.097)
 (13.953)
 countryGreece
 -20.906***
 -21.529***
 -22.485***
 -38.633***
 (4.497)
 (4.496)
 (5.196)
 (11.560)
 countryItaly
 -21.633***

-21.638***
 -11.914
 -21.001
 (4.955)
 (4.941)
 (12.118)
 (13.435)
 countryNetherlands
 -19.248***
 -19.277***
 -31.104***
 -64.169***
 (4.612)
 (4.599)
 (11.263)
 (23.958)
 countryNorway
 -14.201***
 -14.255***
 -30.662**
 -77.140**
 (4.531)
 (4.518)
 (15.282)
 (33.421)
 countryPoland
 -17.948***
 -18.908***
 -37.600
 -40.162*
 (4.516)
 (4.513)
 (24.323)
 (24.367)
 countryPortugal
 -17.839***

-17.838***
 -34.004*
 -50.385**
 (4.496)
 (4.483)
 (17.693)
 (20.555)
 countrySpain
 -33.112***
 -33.071***
 -37.599***
 -57.437***
 (5.351)
 (5.336)
 (6.957)
 (14.468)
 countrySweden
 -16.184***
 -16.241***
 -14.421*
 -56.783**
 (4.571)
 (4.560)
 (8.460)
 (28.382)
 countrySwitzerland
 -10.246**
 -10.629**
 1.132
 -20.048
 (4.498)
 (4.488)
 (14.584)
 (19.900)
 countryUnited_Kingdom
 -14.000***

-14.047***
 -25.849**
 -64.491**
 (5.054)
 (5.039)
 (11.420)
 (27.223)
 countryUnited_States_of_America
 -13.301***
 -13.283***
 -33.846*
 -88.464**
 (4.520)
 (4.507)
 (19.845)
 (40.171)
 em_share_age
 -0.020***
 -0.019***
 -0.019***
 (0.006)
 (0.006)
 (0.006)
 em_sexratio
 0.001
 0.001
 0.001
 (0.004)
 (0.004)
 (0.004)
 unwpp_pop
 pop_ageratio
 -4.910
 0.695
 (5.018)
 (6.165)

```

pop_sexratio
4.886
6.108
(5.619)
(5.671)
kin_size
110.425
(70.623)
Constant
21.236***
23.239***
-61.096
-794.371
(3.201)
(3.249)
(269.979)
(541.072)
Observations
1,671
1,671
1,671
1,671
R2
0.125
0.131
0.132
0.133
Adjusted R2
0.116
0.121
0.121
0.121
Residual Std. Error
31.475 (df = 1653)
31.383 (df = 1651)
31.390 (df = 1649)

```

31.376 (df = 1648)

F Statistic

13.882*** (df = 17; 1653)

13.106*** (df = 19; 1651)

11.916*** (df = 21; 1649)

11.495*** (df = 22; 1648)

Note:

$p < 0.1$; $p < 0.05$; $p < 0.01$

```
stargazer(nuclear_kinstructure, gparents_kinstructure, children_kinstructure, column.labels = c("Nuclear", "Grandparents", "Children"))
```

Different Types of Kin

Dependent variable:

ebr

Nuclear

Grandparents

Children

(1)

(2)

(3)

em_level

0.651***

0.640***

0.700***

(0.017)

(0.025)

(0.071)

em_share_age

-0.001

0.004**

-0.019***

(0.001)

(0.002)

(0.006)

em_sexratio

0.0002

-0.0002

0.001

(0.001)
(0.001)
(0.004)
unwpp_pop
pop_ageratio
0.976
-1.285
0.695
(1.483)
(2.143)
(6.165)
pop_sexratio
3.620***
1.455
6.108
(1.364)
(1.971)
(5.671)
kin_size
32.382*
-29.528
110.425
(16.988)
(24.549)
(70.623)
Constant
-349.570***
105.165
-794.371
(130.153)
(188.078)
(541.072)
Observations
1,671
1,671
1,671

```

R2
0.752
0.640
0.133
Adjusted R2
0.749
0.635
0.121
Residual Std. Error (df = 1648)
7.547
10.906
31.376
F Statistic (df = 22; 1648)
227.284***
133.208***
11.495***
Note:


$p < 0.1$ ;  $p < 0.05$ ;  $p < 0.01$


```

Other Analysis: Probability of Losing Kin

Another interesting concept to explore is changes in the probability of losing kin. A measure I use a lot here is absolute difference in the probability of losing kin across the two simulation scenarios (I have also considered relative, although it is a little problematic for a few noisy estimates for which absolute magnitudes are very low but relative difference is high.) The current measure represents a percentage point change, which seems fairly straightforward to me.

```

kin_probs <- kin_ebr %>%
  mutate(n_losekin = (n_losekin_covid+n_losekin_other)/2,
         n_total = (n_total_covid + n_total_other)/2,
         pc_within = (n_within/n_total)*100,
         pc_losekin = 100*(n_losekin/n_within))

```

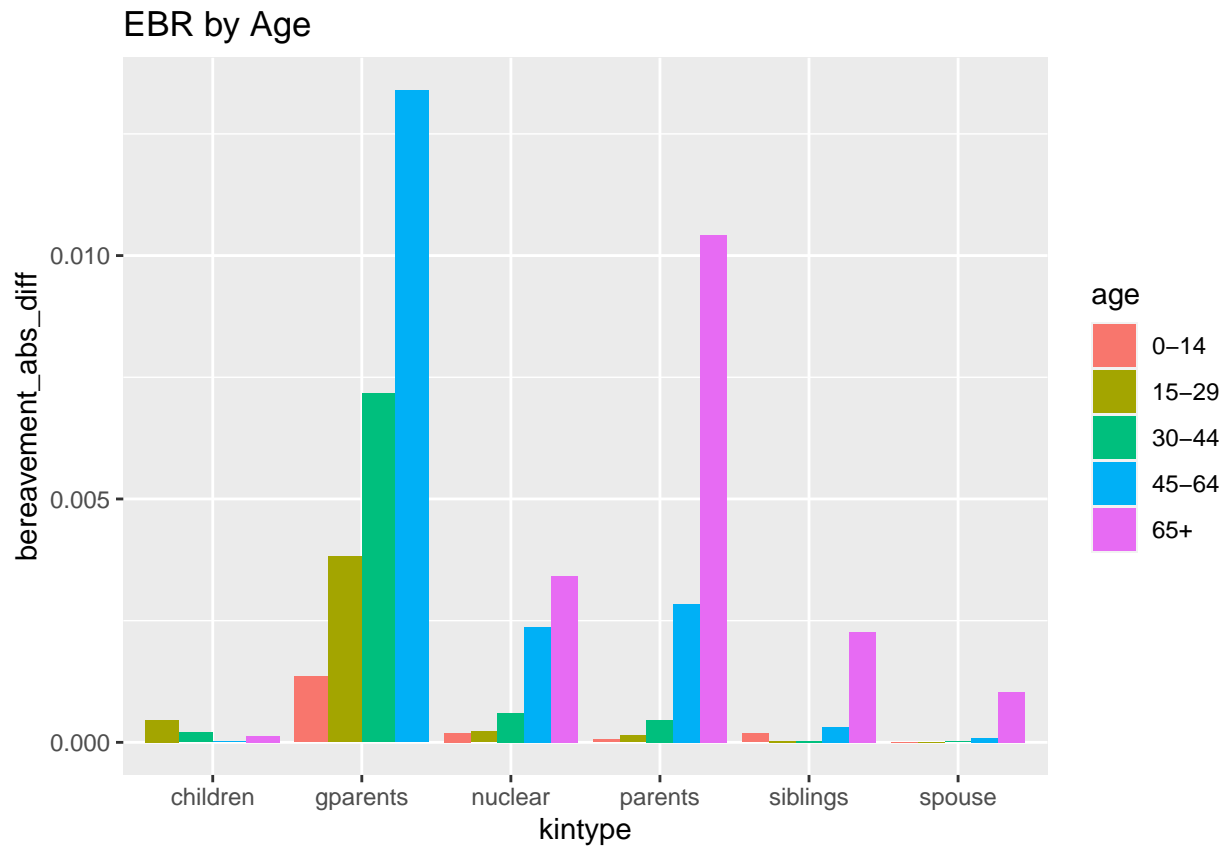
EBR

```

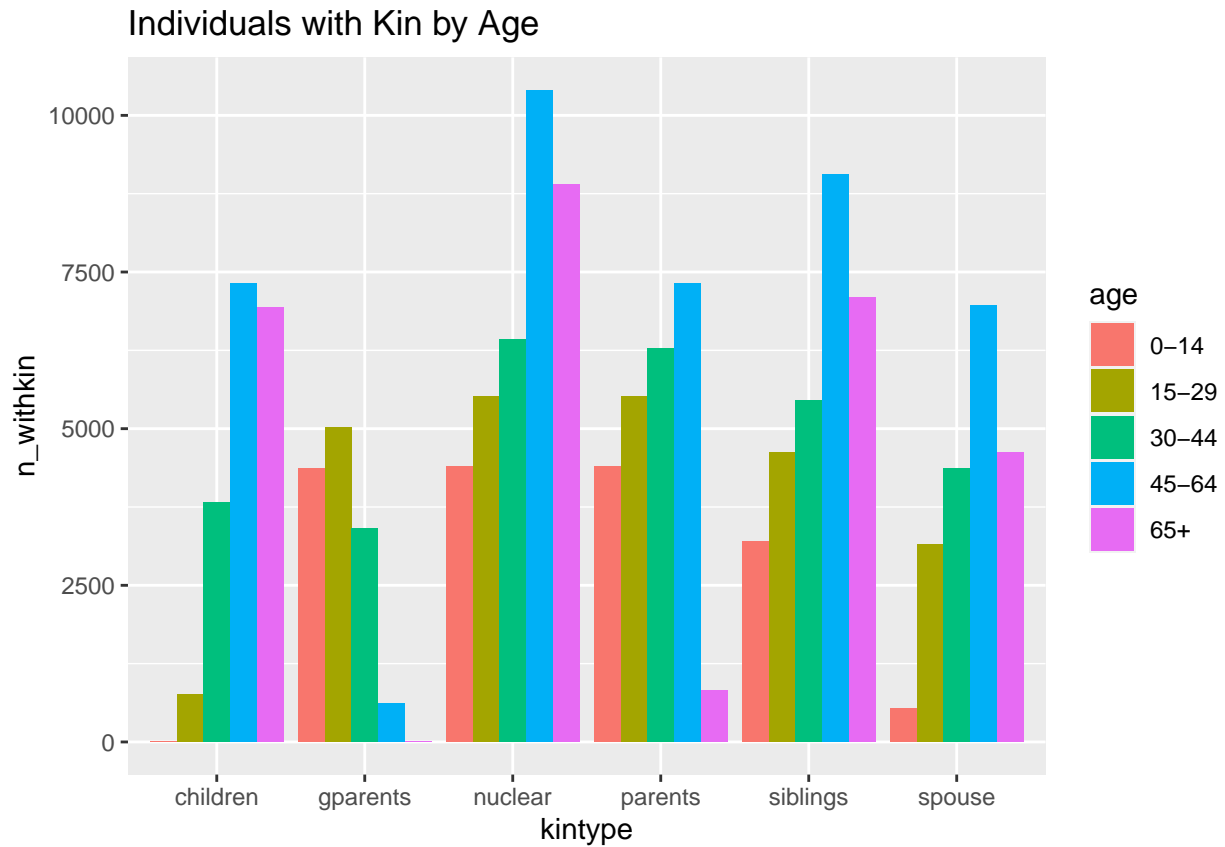
#Visualize the distribution of kin
kin_probs %>%
  ungroup() %>%
  group_by(kintype, age) %>%
  summarize(n_within = mean(n_within, na.rm = T),
            n_losekin = mean(n_losekin, na.rm = T),
            pc_within = mean(pc_within, na.rm = T),
            n_total = mean(n_total, na.rm = T),
            bereavement_covid = mean(bereavement_covid, na.rm = T),
            bereavement_other = mean(bereavement_other, na.rm = T),
            bereavement_abs_diff = bereavement_covid - bereavement_other) %>%

```

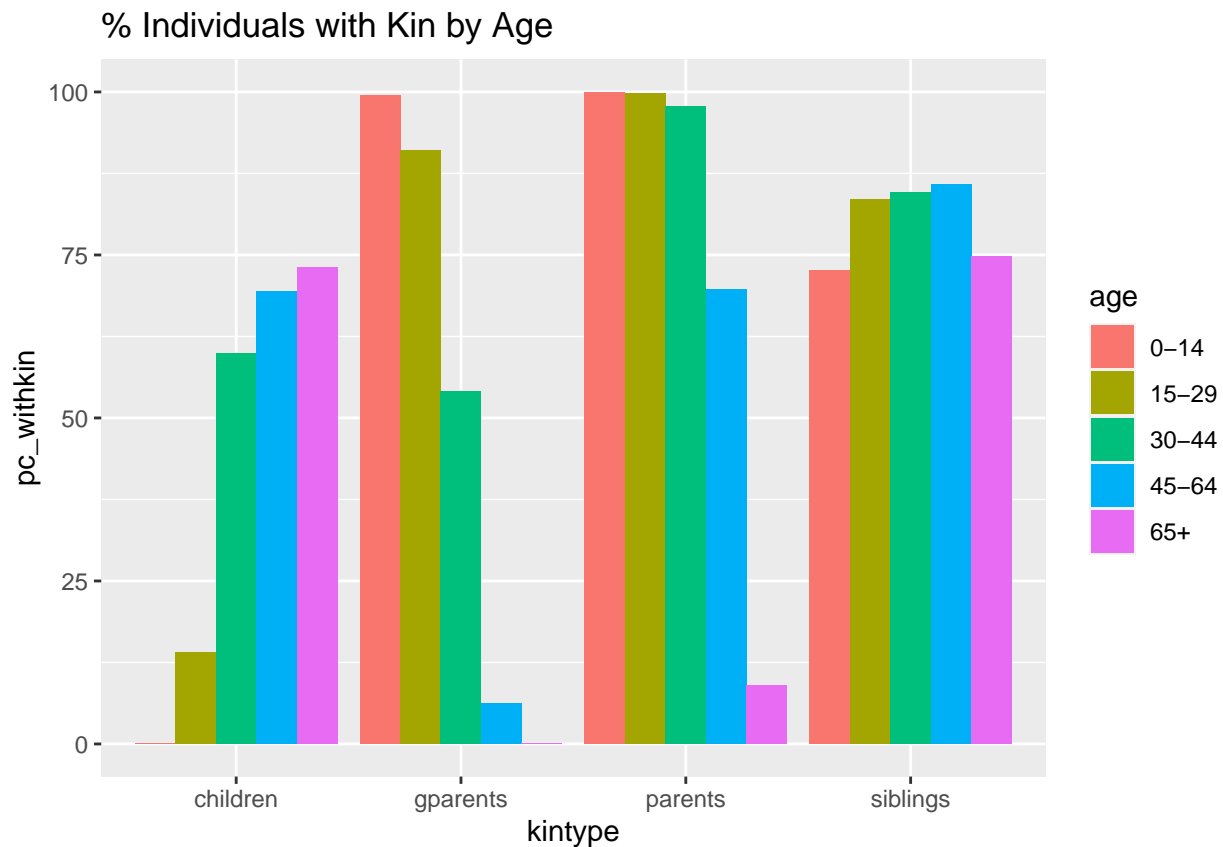
```
filter(pc_within > 1) %>%
ggplot() + geom_bar(aes(x = kintype, y = bereavement_abs_diff, fill = age), position = "dodge", stat = "identity")
```



```
#Visualize the distribution of kin
kin_probs %>%
  ungroup() %>%
  group_by(kintype, age) %>%
  summarize(n_within = mean(n_within, na.rm = T),
            n_losekin = mean(n_losekin, na.rm = T),
            pc_within = mean(pc_within, na.rm = T),
            n_total = mean(n_total, na.rm = T)) %>%
  ggplot() + geom_bar(aes(x = kintype, y = n_within, fill = age), position = "dodge", stat = "identity")
```

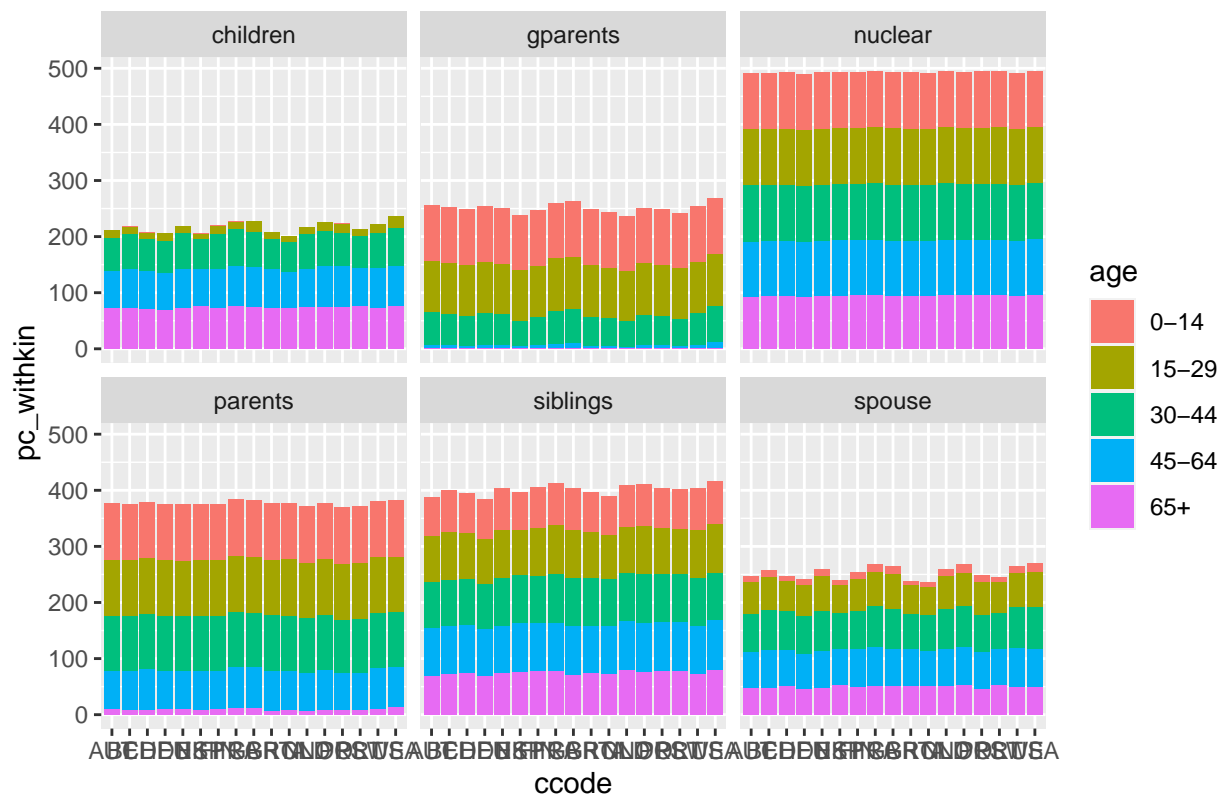


```
#Visualize the distribution of kin
kin_probs %>%
  ungroup() %>%
  group_by(kintype, age) %>%
  summarize(n_within = mean(n_within, na.rm = T),
            n_losekin = mean(n_losekin, na.rm = T),
            pc_within = mean(pc_within, na.rm = T),
            n_total = mean(n_total, na.rm = T)) %>%
  filter(kintype %in% c("nuclear", "spouse")) %>%
  ggplot() + geom_bar(aes(x = kintype, y = pc_within, fill = age), position = "dodge", stat = "identity")
```



```
#Visualize the distribution of kin by country
kin_probs %>%
  ungroup() %>%
  group_by(ccode, kintype, age) %>%
  summarize(n_within = mean(n_within, na.rm = T),
            n_losekin = mean(n_losekin, na.rm = T),
            pc_within = mean(pc_within, na.rm = T),
            n_total = mean(n_total, na.rm = T)) %>%
  #filter(kintype == "gparents") %>%
  ggplot() + geom_bar(aes(x = ccode, y = pc_within, fill = age), position = "stack", stat = "identity")
```

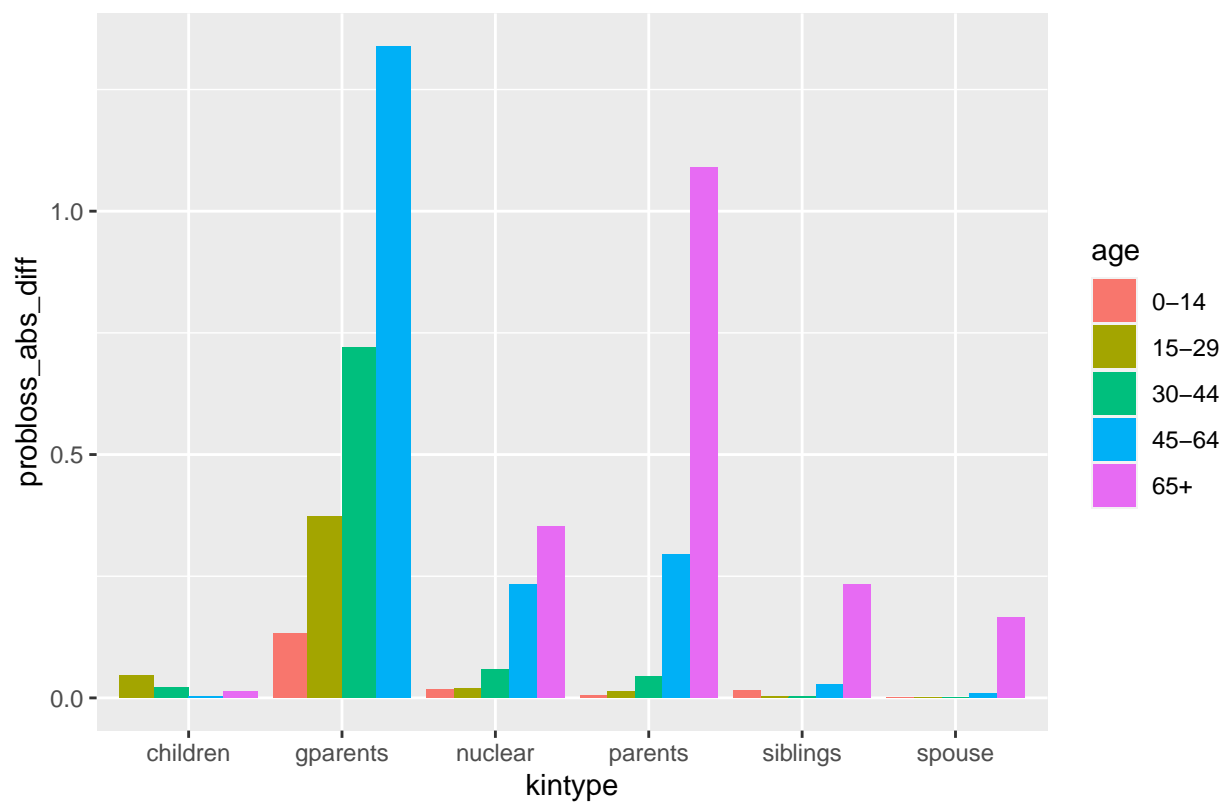

% Individuals with Kin by Age and Country



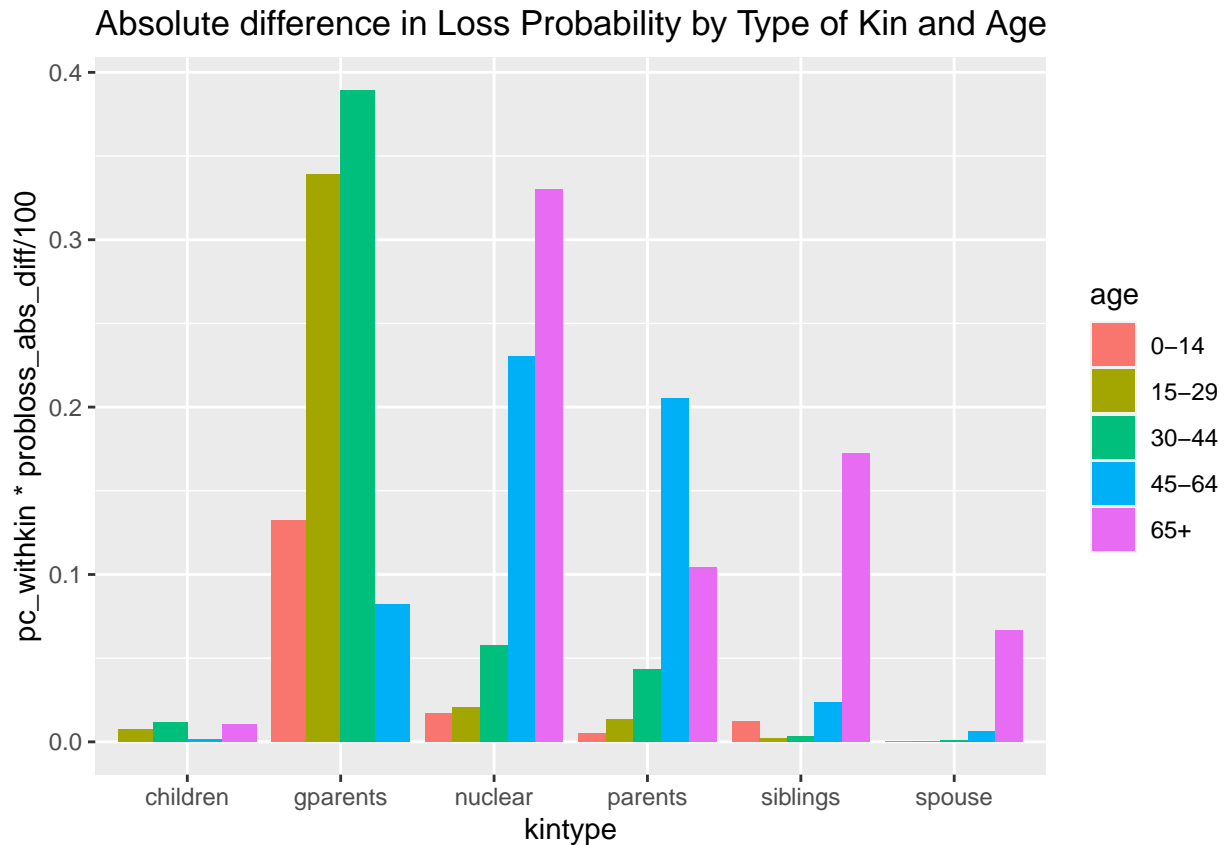
```
prob_loss_age <- kin_probs %>%
  ungroup() %>%
  filter(pc_within > 1) %>%
  group_by(kintype, sex, age) %>%
  summarize(n_within_covid = mean(n_within_covid, na.rm = T),
            n_within_other = mean(n_within_other, na.rm = T),
            n_losekin_covid = mean(n_losekin_covid, na.rm = T),
            n_losekin_other = mean(n_losekin_other, na.rm = T),
            pc_within = mean(pc_within, na.rm = T)) %>%
  mutate(probloss_covid = 100*(n_losekin_covid/n_within_covid),
         probloss_other = 100*(n_losekin_other/n_within_other),
         probloss_rel_diff = 100*(probloss_covid - probloss_other)/probloss_other,
         probloss_abs_diff = probloss_covid - probloss_other)
```

```
prob_loss_age %>%
  ggplot() + geom_bar(aes(x = kintype, y = probloss_abs_diff, fill = age), position = "dodge", stat = ".")
```

Absolute difference in Loss Probability by Type of Kin and Age



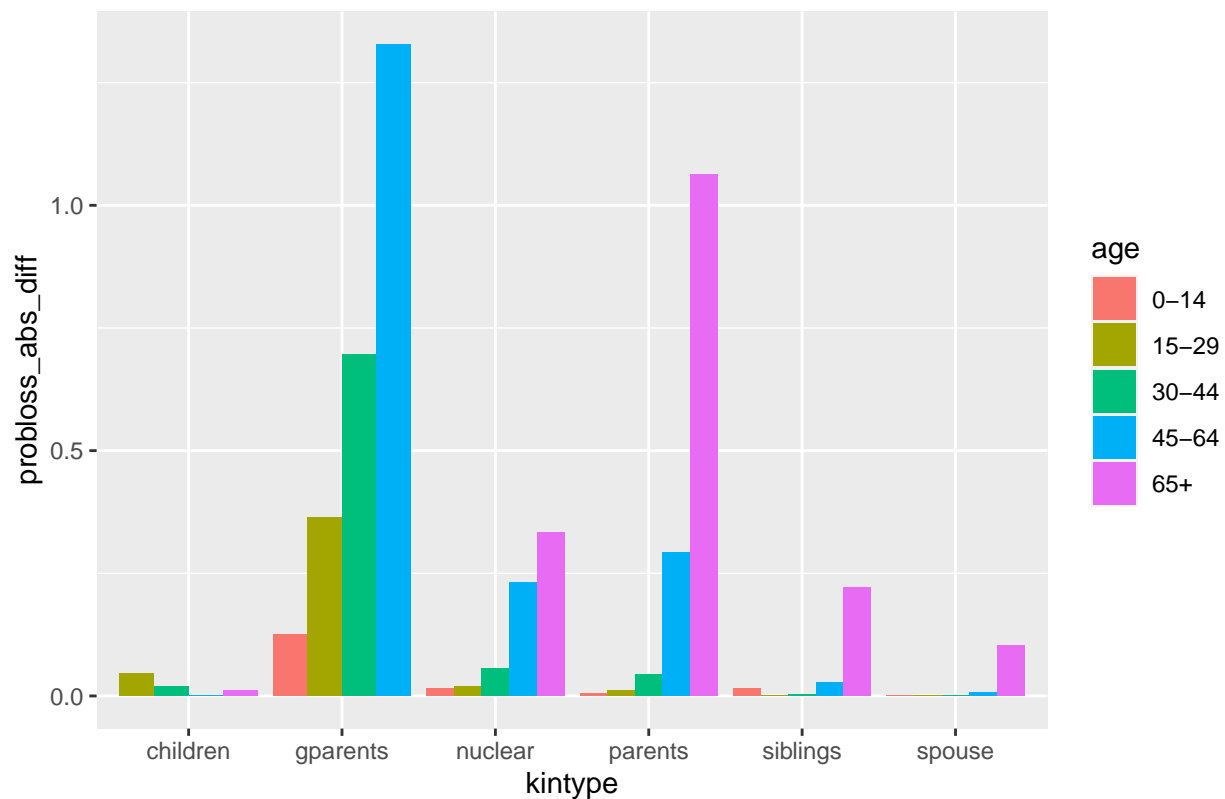
```
prob_loss_age %>%
  ggplot() + geom_bar(aes(x = kintype, y = pc_within*probloss_abs_diff/100, fill = age), position = "dodge")
```



```
prob_loss_age <- kin_probs %>%
  ungroup() %>%
  filter(pc_within > 1) %>%
  group_by(kintype, age) %>%
  summarize(n_within_covid = mean(n_within_covid, na.rm = T),
            n_within_other = mean(n_within_other, na.rm = T),
            n_losekin_covid = mean(n_losekin_covid, na.rm = T),
            n_losekin_other = mean(n_losekin_other, na.rm = T),
            pc_within = mean(pc_within, na.rm = T)) %>%
  mutate(probloss_covid = 100*(n_losekin_covid/n_within_covid),
         probloss_other = 100*(n_losekin_other/n_within_other),
         probloss_rel_diff = 100*(probloss_covid - probloss_other)/probloss_other,
         probloss_abs_diff = probloss_covid - probloss_other)
```

```
prob_loss_age %>%
  ggplot() + geom_bar(aes(x = kintype, y = probloss_abs_diff, fill = age), position = "dodge", stat = "sum")
```

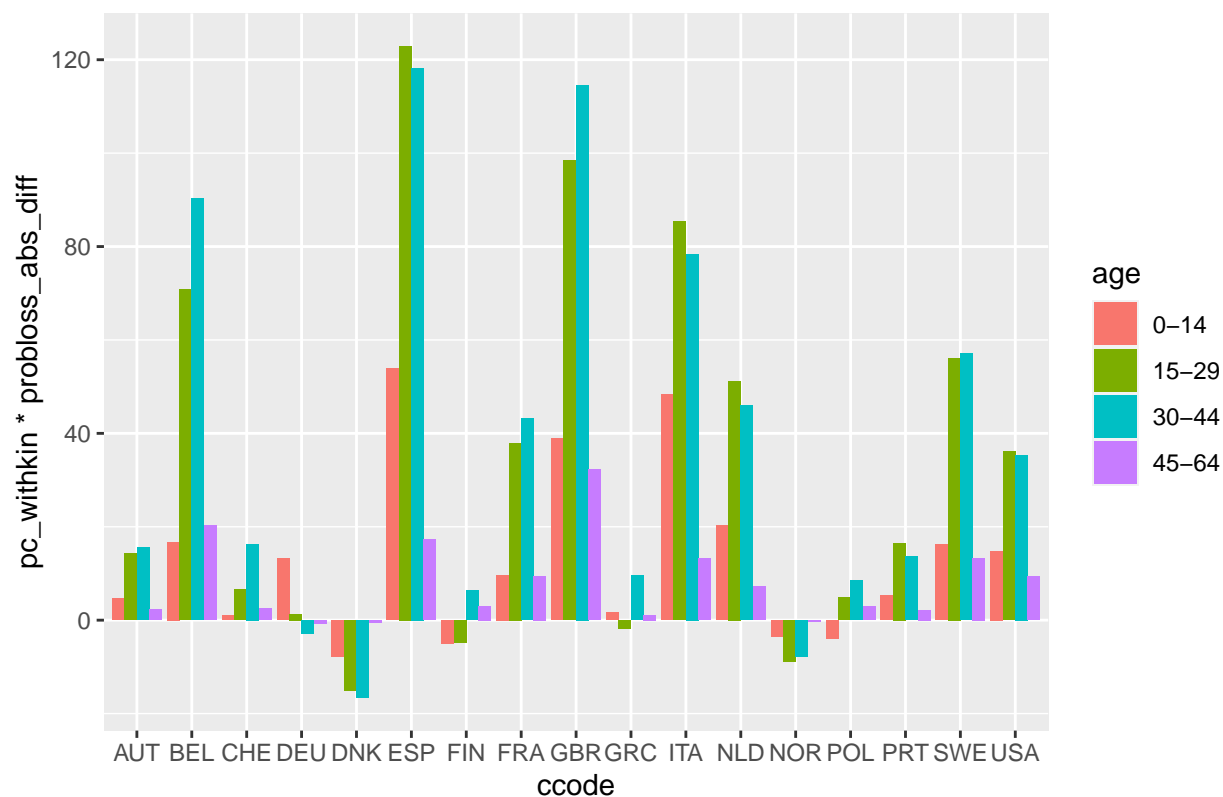
Absolute difference in Loss Probability by Type of Kin and Age



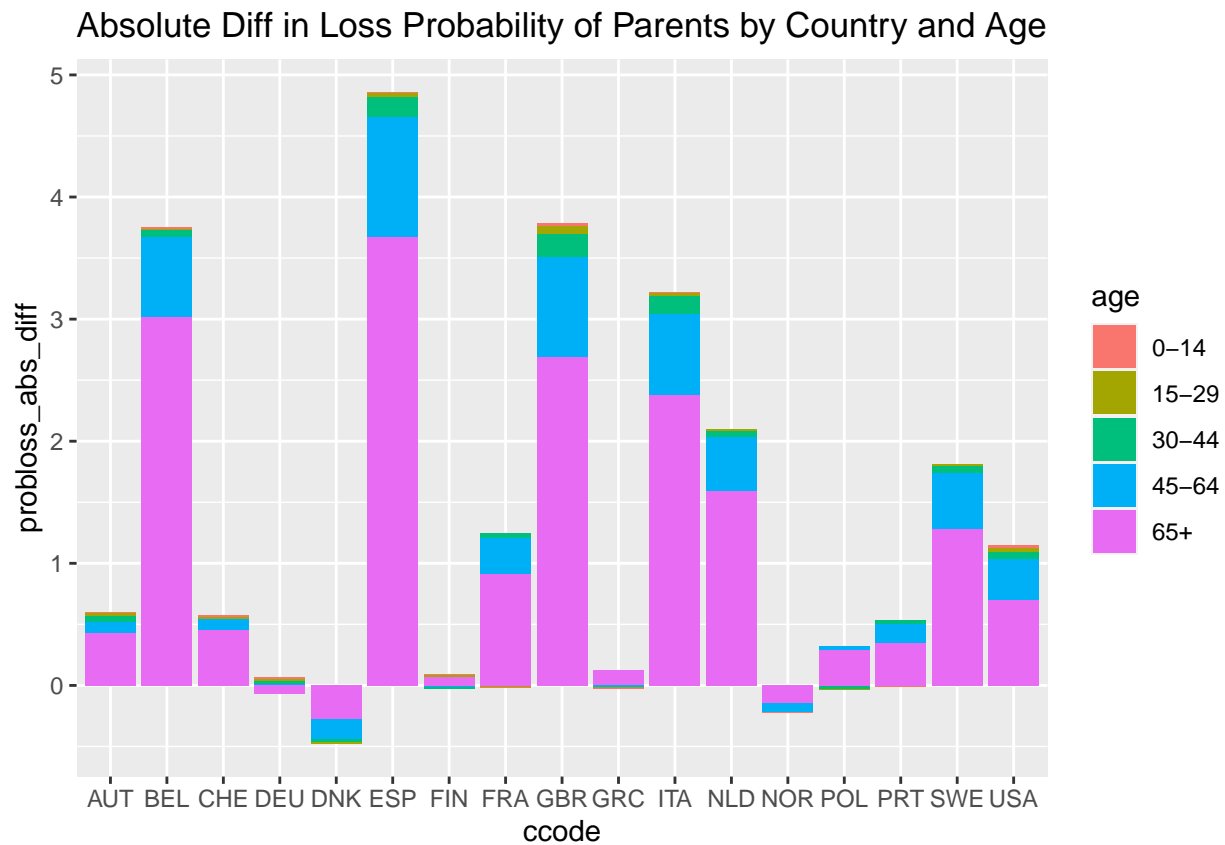
```
#By country
prob_loss_country <- kin_probs %>%
  ungroup() %>%
  filter(pc_within > 1) %>%
  group_by(ccode, country, kintype, age) %>%
  summarize(n_within_covid = mean(n_within_covid, na.rm = T),
            n_within_other = mean(n_within_other, na.rm = T),
            n_losekin_covid = mean(n_losekin_covid, na.rm = T),
            n_losekin_other = mean(n_losekin_other, na.rm = T),
            pc_within = mean(pc_within, na.rm = T)) %>%
  mutate(probloss_covid = 100*(n_losekin_covid/n_within_covid),
         probloss_other = 100*(n_losekin_other/n_within_other),
         probloss_rel_diff = 100*(probloss_covid - probloss_other)/probloss_other,
         probloss_abs_diff = probloss_covid - probloss_other)

prob_loss_country %>%
  filter(kintype == "gparents") %>%
  ggplot() + geom_bar(aes(x = ccode, y = pc_within*probloss_abs_diff, fill = age), position = "dodge",
```

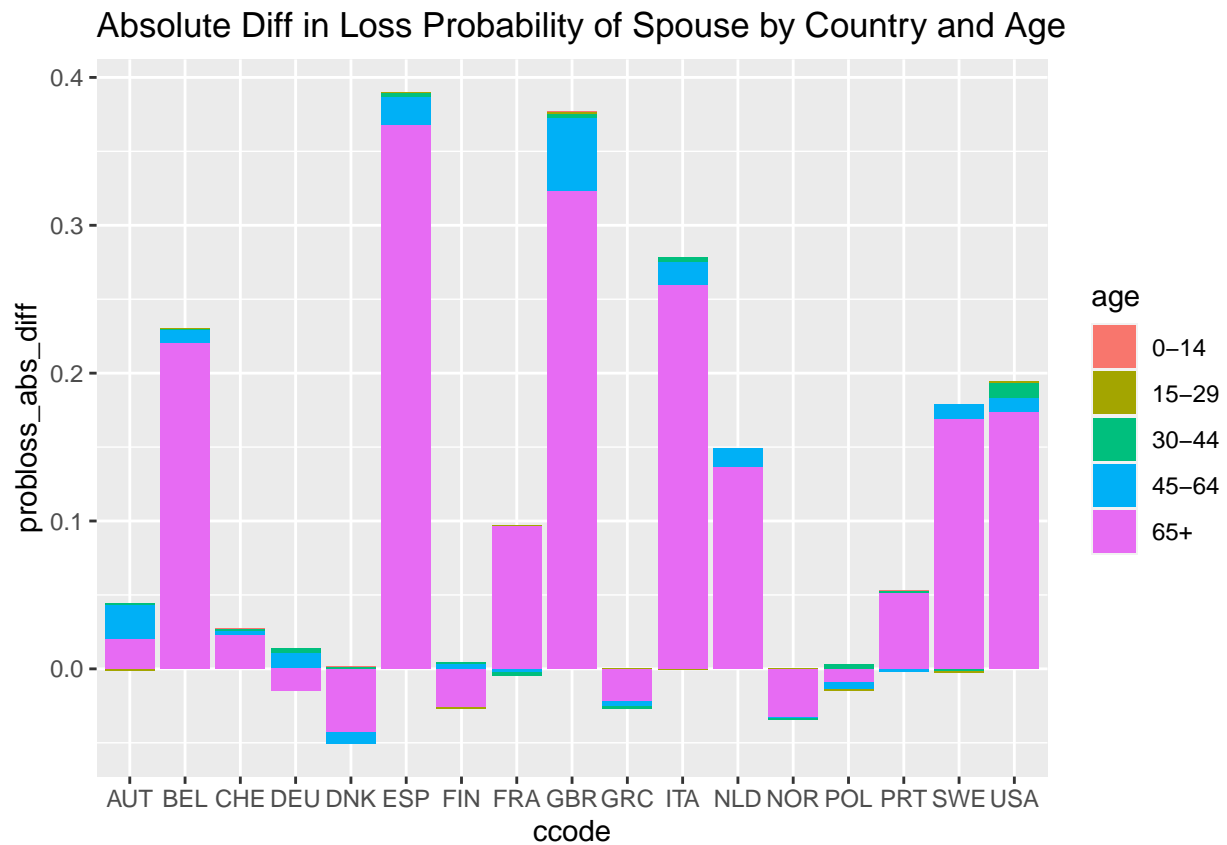
Absolute Diff in Loss Probability of Grandparents by Country and Age



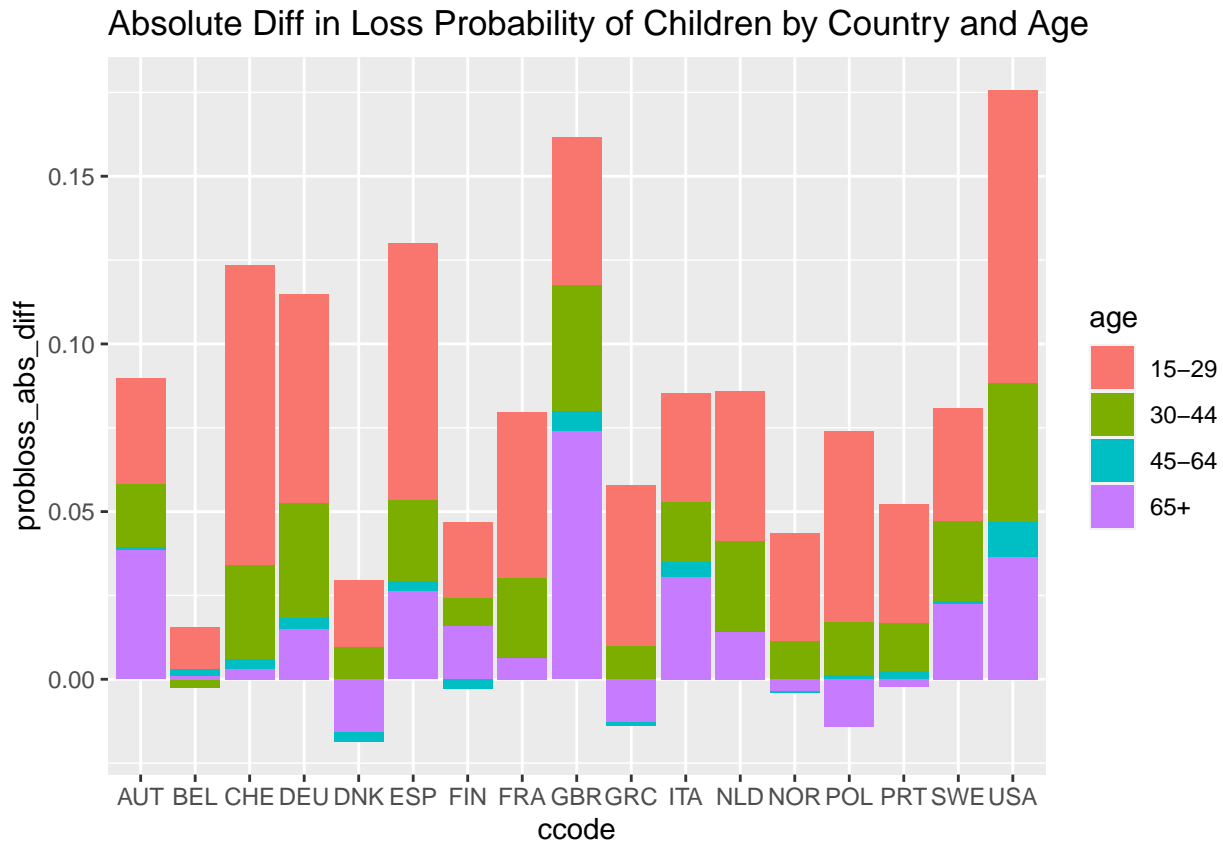
```
prob_loss_country %>%
  filter(kintype == "parents") %>%
  ggplot() + geom_bar(aes(x = ccode, y = probloss_abs_diff, fill = age), position = "stack", stat = "id")
```



```
prob_loss_country %>%
  filter(kintype == "spouse") %>%
  ggplot() + geom_bar(aes(x = ccode, y = probloss_abs_diff, fill = age), position = "stack", stat = "identity")
```



```
prob_loss_country %>%
  filter(kintype == "children") %>%
  ggplot() + geom_bar(aes(x = ccode, y = probloss_abs_diff, fill = age), position = "stack", stat = "identity")
```

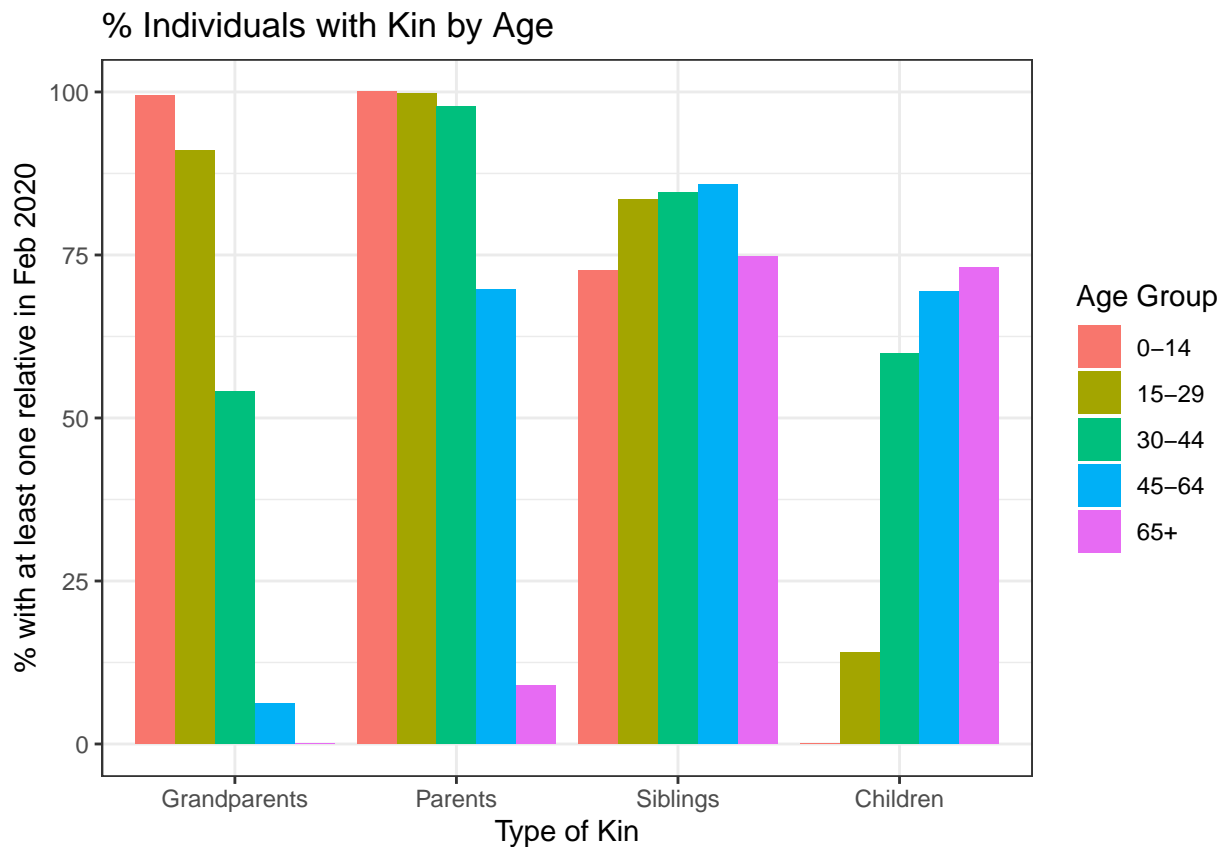


Plots for Presentation

Distribution of kin

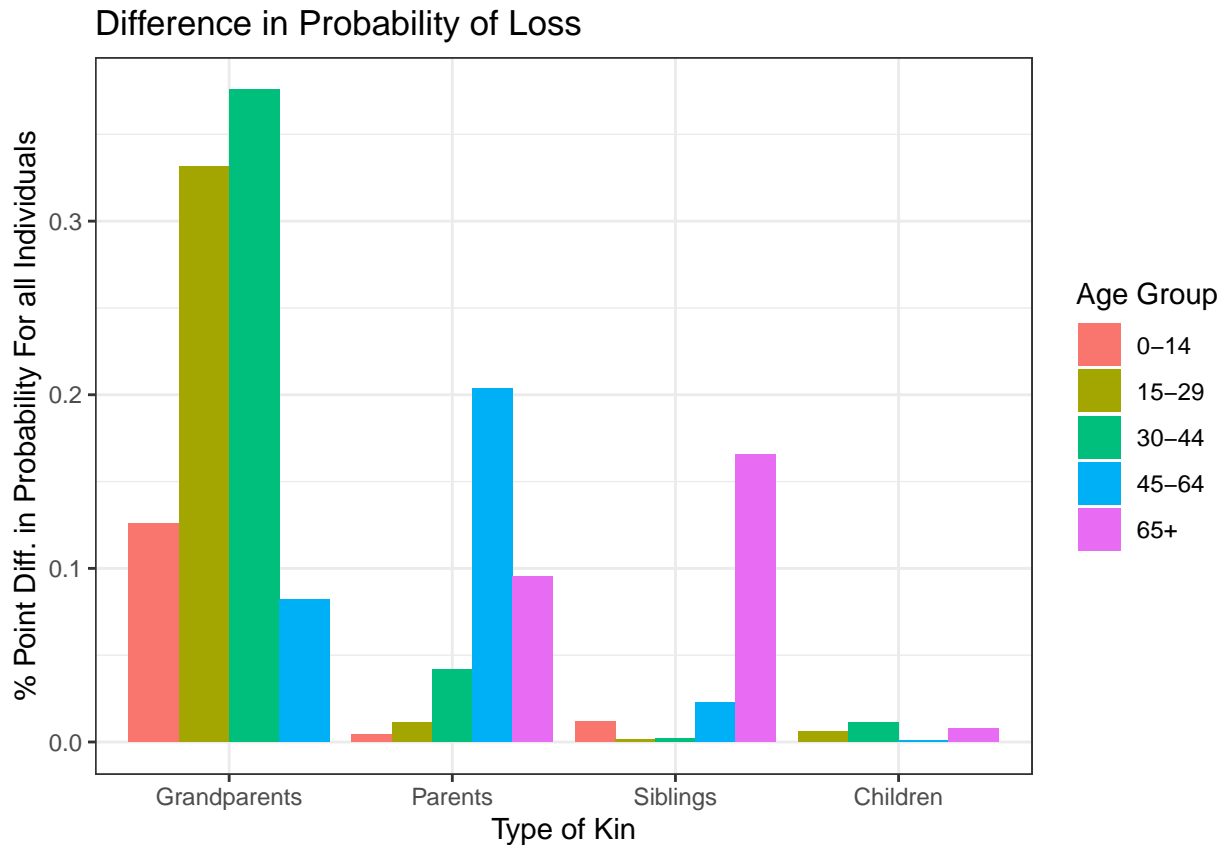
#Visualize the distribution of kin

```
kin_probs %>%
  ungroup() %>%
  group_by(kintype, age) %>%
  summarize(n_within = mean(n_within, na.rm = T),
            n_losekin = mean(n_losekin, na.rm = T),
            pc_within = mean(pc_within, na.rm = T),
            n_total = mean(n_total, na.rm = T)) %>%
  filter(kintype %in% c("nuclear", "spouse")) %>%
  ungroup() %>%
  mutate(kintype = str_to_title(if_else(kintype == "gparents", "grandparents", kintype)),
         kintype = ordered(kintype, levels = c("Grandparents", "Parents", "Siblings", "Children"))) %>%
  ggplot() + geom_bar(aes(x = kintype, y = pc_within, fill = age), position = "dodge", stat = "identity") +
  theme_bw()
```

Probability of Loss

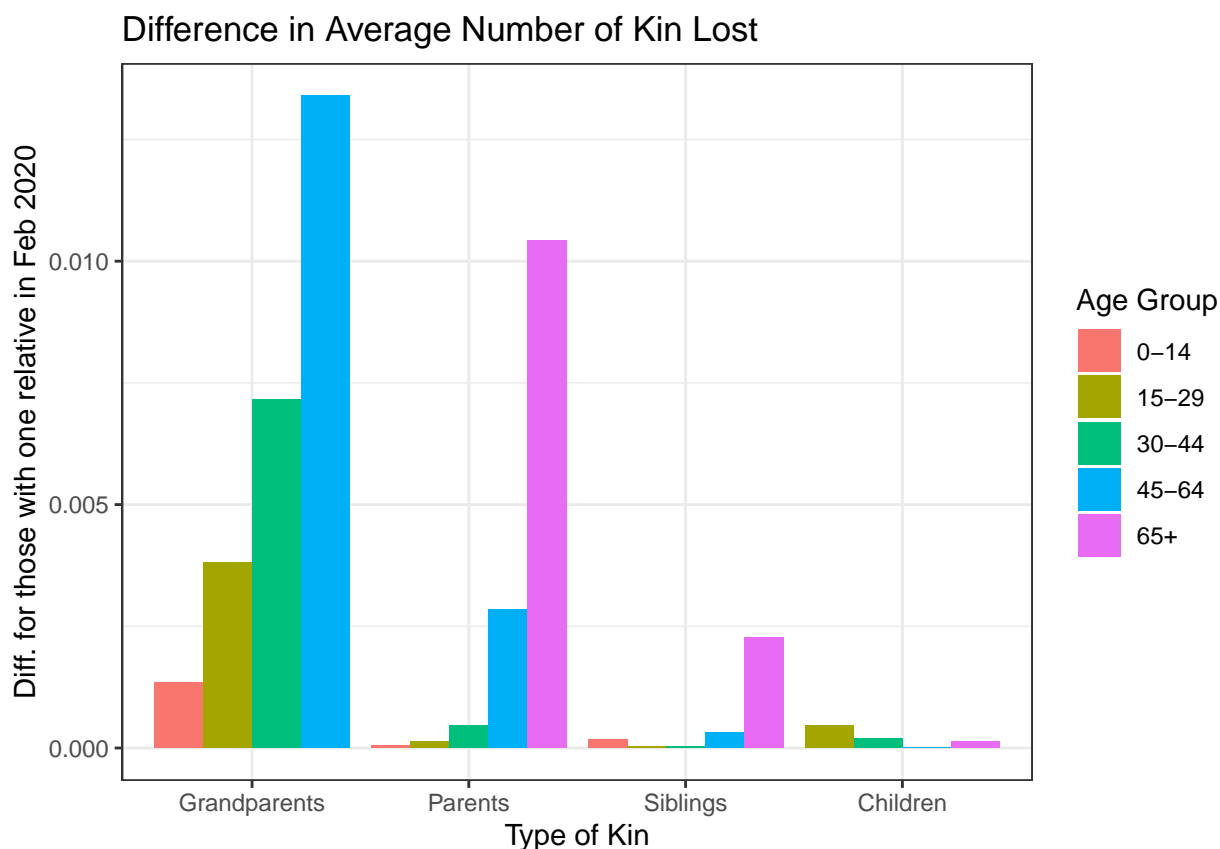
```
prob_loss_age %>%
  filter(kintype %ni% c("nuclear", "spouse")) %>%
  ungroup() %>%
  mutate(kintype = str_to_title(if_else(kintype == "gparents", "grandparents", kintype)),
         kintype = ordered(kintype, levels = c("Grandparents", "Parents", "Siblings", "Children"))) %>%
  ggplot() + geom_bar(aes(x = kintype, y = pc_within*probloss_abs_diff/100, fill = age), position = "dodge")
```



Expected Excess Bereavement

#Visualize the difference in expected kin

```
kin_probs %>%
  ungroup() %>%
  group_by(kintype, age) %>%
  summarize(n_within = mean(n_within, na.rm = T),
            n_losekin = mean(n_losekin, na.rm = T),
            pc_within = mean(pc_within, na.rm = T),
            n_total = mean(n_total, na.rm = T),
            bereavement_covid = mean(bereavement_covid, na.rm = T),
            bereavement_other = mean(bereavement_other, na.rm = T),
            bereavement_abs_diff = bereavement_covid - bereavement_other) %>%
  filter(kintype %in% c("nuclear", "spouse")) %>%
  ungroup() %>%
  mutate(kintype = str_to_title(if_else(kintype == "gparents", "grandparents", kintype)),
         kintype = ordered(kintype, levels = c("Grandparents", "Parents", "Siblings", "Children"))) %>%
  filter(pc_within > 1) %>%
  ggplot() + geom_bar(aes(x = kintype, y = bereavement_abs_diff, fill = age), position = "dodge", stat = "sum") +
  theme_bw()
```



Country Level

```
em_country <- em_level %>%
  group_by(country, ccode) %>%
  summarize(em_level = mean(em_level, na.rm = T))

prob_loss_country_sum <- kin_probs %>%
  ungroup() %>%
  filter(pc_within > 1) %>%
  group_by(ccode, country, kintype) %>%
  summarize(n_within_covid = mean(n_within_covid, na.rm = T),
            n_within_other = mean(n_within_other, na.rm = T),
            n_losekin_covid = mean(n_losekin_covid, na.rm = T),
            n_losekin_other = mean(n_losekin_other, na.rm = T),
            pc_within = mean(pc_within, na.rm = T),
            bereavement_covid = mean(bereavement_covid, na.rm = T),
            bereavement_other = mean(bereavement_other, na.rm = T),
            bereavement_abs_diff = bereavement_covid - bereavement_other) %>%
  mutate(probloss_covid = 100*(n_losekin_covid/n_within_covid),
         probloss_other = 100*(n_losekin_other/n_within_other),
         probloss_rel_diff = 100*(probloss_covid - probloss_other)/probloss_other,
         probloss_abs_diff = probloss_covid - probloss_other)

pop_ageratio_country <- pop_ageratio %>%
  ungroup() %>%
  group_by(country) %>%
  summarize(pop_ageratio = mean(pop_ageratio, na.rm = T))
```

```
figuredata <- merge(pop_ageratio_country, merge(em_country, prob_loss_country_sum))
```

```
figuredata %>%
  filter(kintype %in% c("nuclear", "spouse")) %>%
  ungroup() %>%
  mutate(kintype = str_to_title(if_else(kintype == "gparents", "grandparents", kintype)),
         kintype = ordered(kintype, levels = c("Grandparents", "Parents", "Siblings", "Children"))) %>%
  ggplot(aes(x = em_level, y = probloss_abs_diff)) + geom_point() + geom_smooth(method = "lm", se = F) +
  labs(title = "Excess Mortality and Excess Kin Loss by Country", y = "Abs. Diff in Probability of Lost
```

