Getting and visualizing demographic data

```
# Loading packages needed
library(eurostat)
library(dplyr)
library(stringr)
library(haven)
library(ggplot2)
```

Retrieving demographic data using eurostat

To start let's retrieve NUTS3 data from Europe.

```
# Demographic data from Helsinki, Vilnius, Mannheim and Heidelberg
eu_population <- get_eurostat("demo_r_pjangrp3",
                              filters = list(
                                geo = c("LT011", "DE126", "DE125", "FI1B1"),
                                sex = c("F", "M"),
                                lastTimePeriod = 1),
                              type = "label") # Retrieve NUTS3 demographic data
kableExtra::kable(eu_population)
```

freq	sex	unit	age	geo	time	values
Annual	Males	Number	Total	Heidelberg, Stadtkreis	2023-01-01	77561
Annual	Males	Number	Total	Mannheim, Stadtkreis	2023-01-01	157130
Annual	Males	Number	Total	Vilniaus apskritis	2023-01-01	397128
Annual	Males	Number	Total	Helsinki-Uusimaa	2023-01-01	849733
Annual	Males	Number	Less than 5 years	Heidelberg, Stadtkreis	2023-01-01	3659
Annual	Males	Number	Less than 5 years	Mannheim, Stadtkreis	2023-01-01	7616

There is not NUTS3 level data available from Ukraine, so we will use data from whole of Ukraine. Also the data from Ukraine is from 2022 as there is no data available from 2023.

```
# Retrieving demographic data for Ukraine
ua_population <- get_eurostat("demo_pjan",
                              filters = list(
                                geo = "UA",
                                lastTimePeriod = 2),
                              type = "label")
kableExtra::kable(ua_population)
```

freq	unit	age	sex	geo	time	values
Annual	Number	Total	Total	Ukraine	2022-01-01	40997698
Annual	Number	Total	Total	Ukraine	2023-01-01	NA
Annual	Number	Total	Males	Ukraine	2022-01-01	19006979
Annual	Number	Total	Males	Ukraine	2023-01-01	NA
Annual	Number	Total	Females	Ukraine	2022-01-01	21990719
Annual	Number	Total	Females	Ukraine	2023-01-01	NA

```
# Filtering out Unkown and Total from age
eu_population <- eu_population %>%
  filter(!age %in% c("Unknown", "Total", "From 85 to 89 years",
                     "90 years or over")) %>%
  na.omit()
ua_population <- ua_population %>%
  filter(!age %in% c("Unknown", "Total", "Open-ended age class")) %>%
  na.omit()
```

We will now harmonize the variable values between the EU cities and Ukraine data. The mapping data is in the file 'input/other/codebook.xlsx'. The data will be read from the file and then used to map the variable values.

```
# Matching the factor levels
mapping_data_eu <- readxl::read_excel(
  "../../input/other/codebook.xlsx", sheet = 5) %>%
  filter(dataset == "europe") %>%
  select(-dataset)
# Printing example of the mapping dataset
kableExtra::kable(mapping_data_eu)
```

variable	original	mapped
age	Less than 5 years	0-19
age	From 5 to 9 years	0-19
age	From 10 to 14 years	0-19
age	From 15 to 19 years	0-19
age	From 20 to 24 years	20-24
age	From 25 to 29 years	25-29

```
# Remapping the variable values
for (i in unique(mapping_data_eu$variable)) {
  eu_population <- eu_population %>%
      left_join(mapping_data_eu %>%
                    filter(variable == i), by = setNames("original", i)) %>%
      mutate(!!sym(i) := coalesce(mapped, !!sym(i))) %>%
      select(-c(variable, mapped))
}
# Removing the extra columns
eu_population <- eu_population %>%
  select(-c(time, freq, unit)) %>%
  group_by(age, sex, geo) %>%
  summarise(values = sum(values)) %>%
  na.omit()

# Matching the factor levels for Ukraine data
mapping_data_ua <- readxl::read_excel(
  "../../input/other/codebook.xlsx", sheet = 5) %>%
  filter(dataset == "ua") %>%
  select(-dataset)
# Remapping the varaible values
for (i in unique(mapping_data_ua$variable)) {
  ua_population <- ua_population %>%
    left_join(mapping_data_ua %>%
                    filter(variable == i), by = setNames("original", i)) %>%
      mutate(!!sym(i) := coalesce(mapped, !!sym(i))) %>%
      select(-c(variable, mapped))
}
# Removing the extra columns
ua_population <- ua_population %>%
  select(-time) %>%
  group_by(age, sex, geo) %>%
  summarise(values = sum(values)) %>%
```

```
  na.omit() %>%
  filter(!sex == "Total")

kableExtra::kable(eu_population)
```

age	sex	geo	values
0-19	Female	Heidelberg	14060
0-19	Female	Helsinki	181207
0-19	Female	Mannheim	27669
0-19	Female	Vilnius	88331
0-19	Male	Heidelberg	14250
0-19	Male	Helsinki	189275

```
kableExtra::kable(ua_population)
```

age	sex	geo	values
0-19	Female	Lviv	3927530
0-19	Male	Lviv	4172653
20-24	Female	Lviv	926575
20-24	Male	Lviv	985767
25-29	Female	Lviv	1188474
25-29	Male	Lviv	1255392

Now we can join the Ukraine data into the rest of the data. After this we can calculate the proportion of different demographic categories.

```
# Combining Ukraine (Lviv) data with the other cities
eu_population <- rbind(eu_population, ua_population)
# Getting counts and proportions for the different demographic categories
eu_population <-  eu_population %>%
  group_by(geo) %>%
  na.omit() %>%
  mutate(prop = values/sum(values))
kableExtra::kable(eu_population, max_rows = 5)
```

age	sex	geo	values	prop
0-19	Female	Heidelberg	14060	0.0866441
0-19	Female	Helsinki	181207	0.1045606
0-19	Female	Mannheim	27669	0.0876839
0-19	Female	Vilnius	88331	0.1040751
0-19	Male	Heidelberg	14250	0.0878150
0-19	Male	Helsinki	189275	0.1092160

```
# Saving the demographic data
# saveRDS(eu_population, "eu_population.rds")
```

Comparing the census demographic breakdown to the population level one

Let's start by reading the census data.

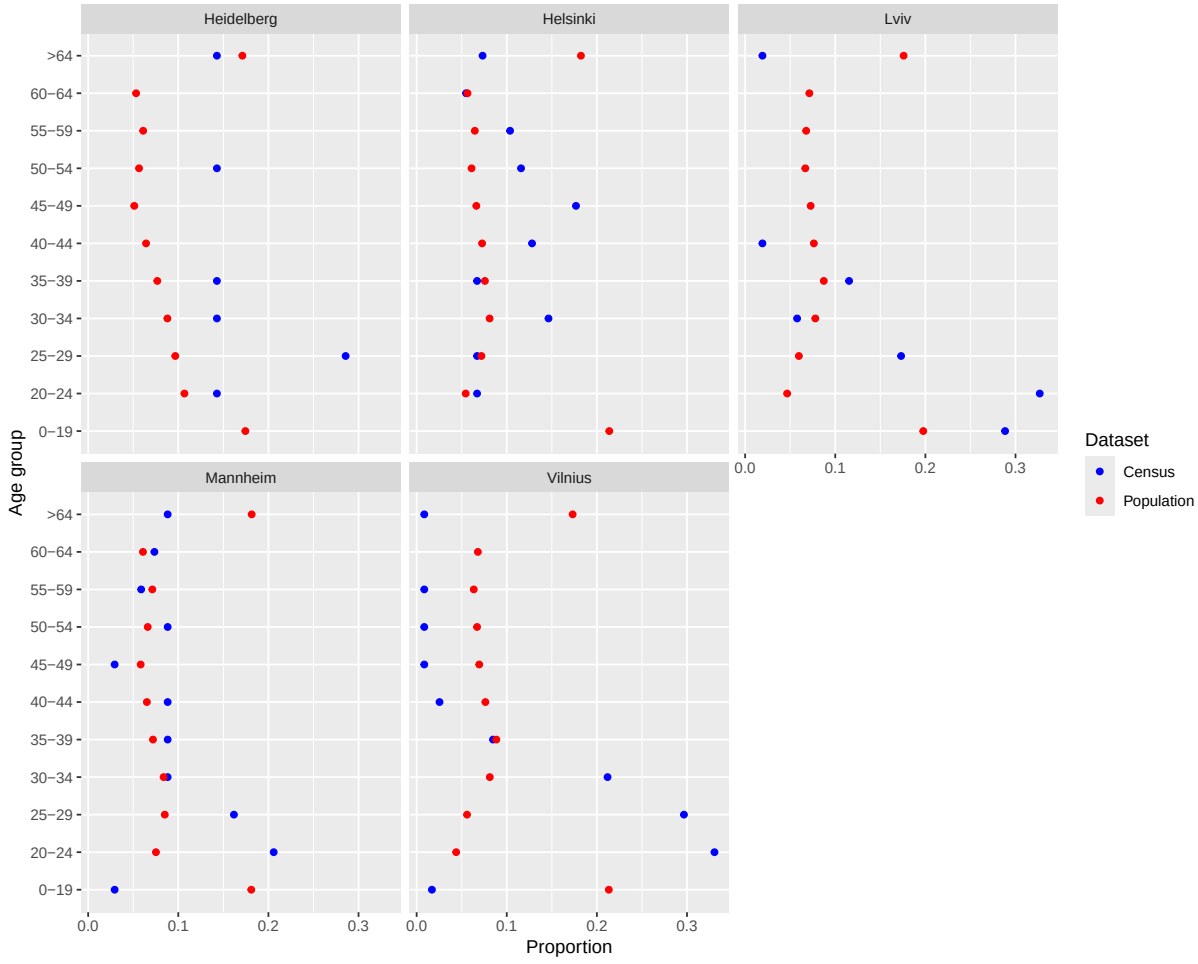
```
# Reading the census data
census_data_all <- read_sav("../../input/data_processed/combined_census_data.sav") %>%
  filter(exclude == 0)
```

```
# Renaming columns to match
eu_population <- eu_population %>%
  rename(City = geo, Age_group = age, Q2 = sex)
```

Now we can plot comparisons of the demographic distributions in the census vs. the population.

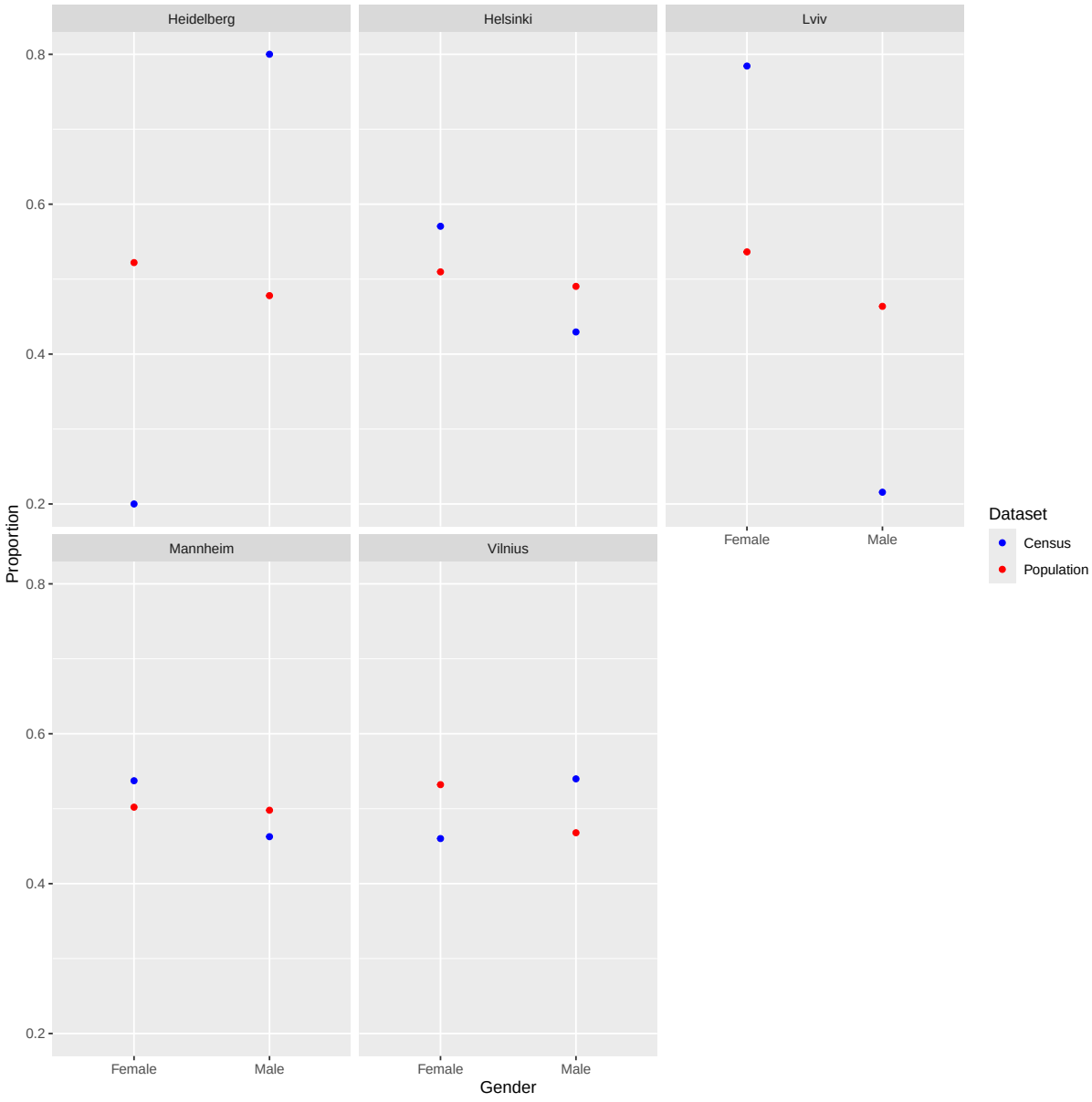
```
# Comparing age distributions
census_data_all %>%
  select(City, Age_group) %>%
  haven::as_factor() %>%
  group_by(City, Age_group) %>%
  count() %>%
  ungroup() %>%
  group_by(City) %>%
  na.omit() %>%
  mutate(prop = n/sum(n)) %>%
  ggplot(aes(x = Age_group, y = prop)) +
  geom_point(aes(color = "Census")) + facet_wrap(vars(City)) +
  geom_point(data = eu_population %>%
               group_by(City, Age_group) %>%
```

```
            summarise(prop = sum(prop)),
          aes(x = Age_group, y = prop, color = "Population")) +
  scale_color_manual(name = "Dataset",
                     breaks = c("Census", "Population"),
                     values = c("Census" = "blue", "Population" = "red")) +
  labs(x = "Age group", y = "Proportion") + coord_flip()
```




```
# Comparing gender distributions
census_data_all %>%
  group_by(City, Q2) %>%
  haven::as_factor() %>%
  filter(Q2 %in% c("Female", "Male")) %>%
  count() %>%
  ungroup() %>%
```

```
  group_by(City) %>%
  na.omit() %>%
  mutate(prop = n/sum(n)) %>%
  ggplot(aes(x = Q2, y = prop)) +
  geom_point(aes(color = "Census")) + facet_wrap(vars(City)) +
  geom_point(data = eu_population %>%
               group_by(City, Q2) %>%
               summarise(prop = sum(prop)),
             aes(x = Q2, y = prop, color = "Population")) +
  scale_color_manual(name = "Dataset",
                     breaks = c("Census", "Population"),
                     values = c("Census" = "blue", "Population" = "red")) +
  labs(x = "Gender", y = "Proportion")
```


Heidelberg
Helsinki
Lviv
Mannheim
Vilnius
Proportion
0.8
0.6
0.4
0.2
Female
Male
Gender
Dataset
Census
Population