# Day 1 Exercises - Data Visualisation Fundamentals

2023-09-19

Welcome to the first exercise session for the course *Data Visualisation and Storytelling*!

This exercise sheet covers material from the first day of lectures. We will be using two datasets, one from Statistics Denmark (DST) exploring age groups and populations of residents in Denmark, and the European Quality of Life Survey (EQLS).

Some of the questions have specific tasks for you to do, others are more open-ended, and you can have a go at them in any order you like.

## Preliminaries

First, set your working directory.

Then, load the libraries that you will be using - these should include dplyr and ggplot2! Don't forget to install these packages first, using install.packages() if they are not already installed.

```
# Include your directory here!
library(dplyr)
library(ggplot2)
library(gridExtra)
library(Hmisc)

# DK Postcode population (2023)
dk_pop <- read.csv("../data/dk_pop_2023.csv")

# European Quality of Life Survey
eqls_data <- read.csv("../data/eqls_2007and2011.csv")
```

## Exercise 1: Univariate Plots

For this exercise we will use the "dk_pop" dataset extracted from DST.

The dataset contains the population of different age groups + the total population in all postcodes in Denmark.

We will ignore the different age groups here and just focus on the total population living in each postcode.

1. Start by plotting the most basic histogram (fig1).

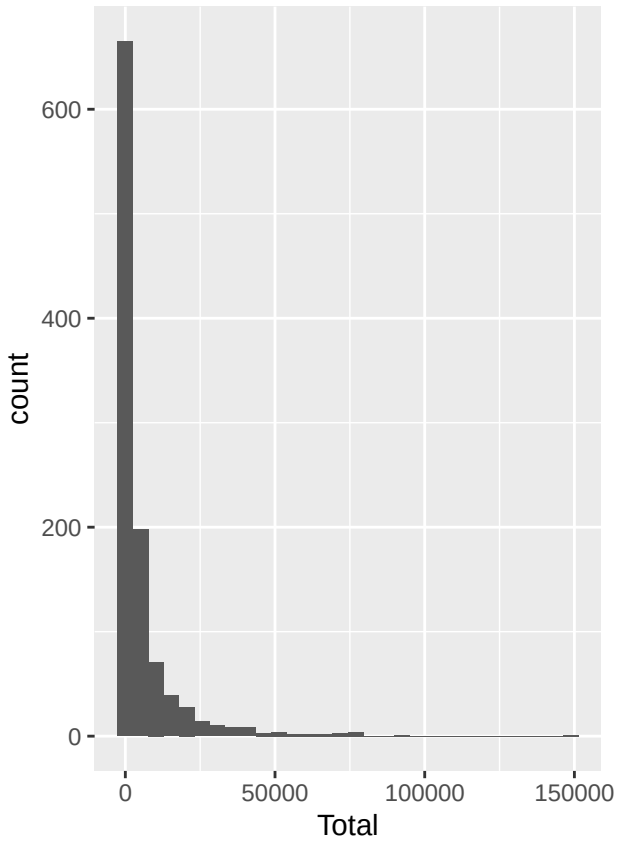
What do you notice about the distribution?

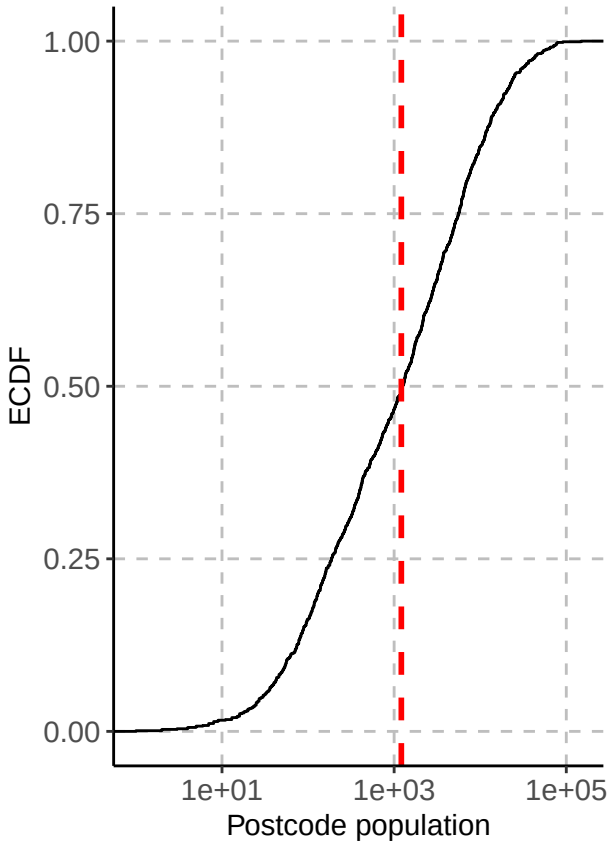
2. Propose another plot (fig2) that incorporates the following elements:

- A better way to visualise the median population size in Danish towns
- A red, dashed, vertical line that corresponds to this median
- Better data-to-ink-ratio: opt for a white background, dashed grid lines, and remove the top and right borders
- Increase the font size of the x and y ticks to 11

Plot the two figures (fig1 and fig2) side by side.

```
# Get the median total population wrt postcodes
median_town_pop <- median(dk_pop$Total)

# Fig 1: the simplest histogram possible
fig1 <- ggplot(dk_pop, aes(x = Total)) +
  geom_histogram()

# Fig 2: a more complex way to show data
fig2 <- ggplot(dk_pop, aes(x = Total)) +
  # cumulative plot
  stat_ecdf(geom = "step") +
  # log-scale the x-axis
  scale_x_log10() +
  labs(
    # rename the x label
    x = "Postcode population",
    # rename the y label
    y = "ECDF"
  ) +
  # choose a minimal theme (no grey background)
  theme_classic() +
  theme(
    # set grid lines
    panel.grid.major = element_line(color = "gray", linetype = "dashed"),
    # set font size of x axis
    axis.text.x = element_text(size = 11L),
    # set font size of y axis
    axis.text.y = element_text(size = 11L)
  ) +
  # add a vertical line
  geom_vline(
    # intercept = median
    xintercept = median_town_pop,
    # dashed line
    linetype = "dashed",
    color = "red",
    linewidth = 1
  )

# Arrange figure as two subplots (2 columns)
grid.arrange(fig1, fig2, ncol = 2L)
```




## Exercise 2: scatter plots and regression

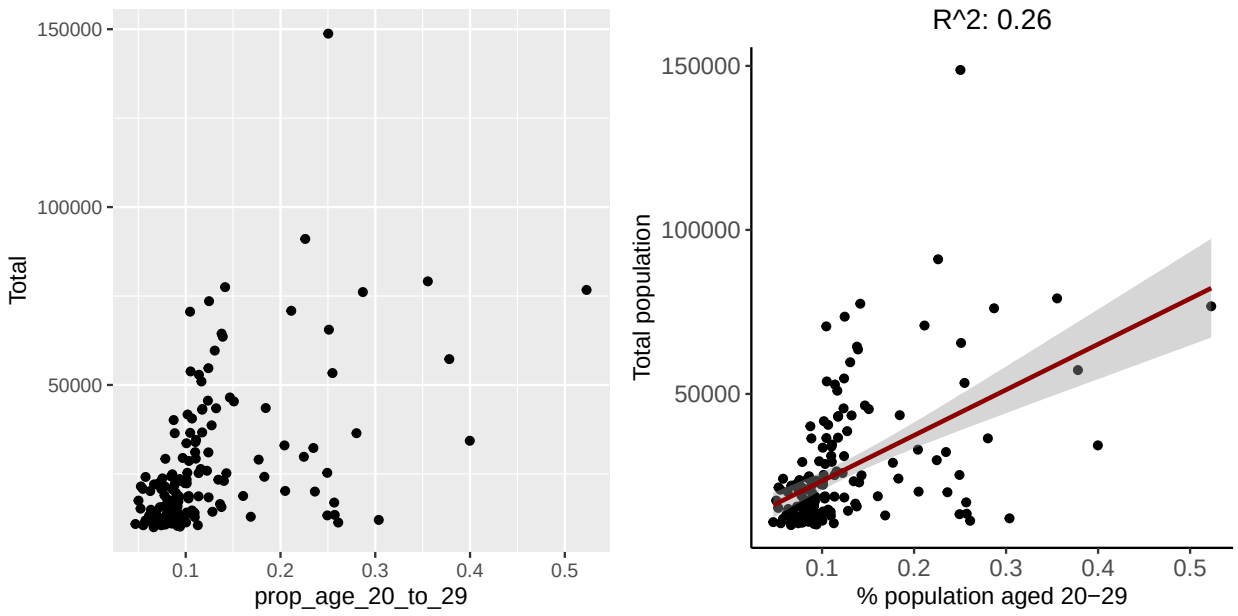
Plot a simple scatter plot (fig1) showing the following: * x-axis: the proportion of inhabitants * y-axis: total population * data: postcodes for which the population is greater or equal than 10000

Augment this plot (fig2) with: * A linear regression model fit, coloured in dark red * A similar theme as Ex. 1 * Title: R^2 coefficient (centered) * Rename the x and y axes with more suitable labels

Plot the two figures (fig1 and fig2) side by side.

```
# People in dense postcodes
dk_dense_pop <- dk_pop[dk_pop$Total >= 10000L, ]

# Proportion of age between 20-29
dk_dense_pop$prop_age_20_to_29 <- ((dk_dense_pop$age20_24 + dk_dense_pop$age25_29) / dk_dense_pop$Total

# Linear regression
mod <- lm(Total ~ prop_age_20_to_29, dk_dense_pop)

# Fig. 1: the simplest scatter plot
fig1 <- ggplot(dk_dense_pop, aes(x = prop_age_20_to_29, y = Total)) +
  geom_point()

# Fig. 2: add stuff to fig 1
fig2 <- fig1 +
  # classic theme
  theme_classic() +
```

```
  # change x-axis, y-axis and title format
  theme(
    axis.text.x = element_text(size = 11L),
    axis.text.y = element_text(size = 11L),
    plot.title = element_text(hjust = 0.5) # centre the title
  ) +
  # Add the regression line
  geom_smooth(method = lm, color = "darkred") +
  # Change x-axis, y-axis and title labels
  labs(
    x = "% population aged 20-29",
    y = "Total population",
    title = paste("R^2:", format(summary(mod)$r.squared, digits = 2L))
  )

# Subplots with 2 columns
grid.arrange(fig1, fig2, ncol = 2L)
```




## Exercise 3: Exploring the European Quality of Life Dataset

Read in the Quality of Life Survey Dataset and answer the following questions:

```
# How many variables are there in the data? And how many observations?
n_variables <- ncol(eqls_data)
n_observations <- nrow(eqls_data)


print(paste("There are ", as.character(n_variables), " variables in the data"))
```

```
## [1] "There are  202  variables in the data"
```

```
print(paste("There are", as.character(n_observations), "observations in the data"))
```

```
## [1] "There are 79270 observations in the data"
```

```r
# How many countries are considered in the dataset?

n_countries <- length(unique(eqls_data$country))

print(paste("There are", as.character(n_countries), "countries in the data"))
```

```
## [1] "There are 35 countries in the data"
```

```r
# How many eqls waves are in the dataset? Create separate dataframes
# corresponding to each wave of the survey. Each wave corresponds to a year that
# the survey was conducted in.
data_2007 <- eqls_data %>% filter(eqls_wave == 2007)
data_2011 <- eqls_data %>% filter(eqls_wave == 2011)

# Is the number of countries in each wave the same?
n_countries_2007 <- length(unique(data_2007$country))
n_countries_2011 <- length(unique(data_2011$country))

# What is the mean number of children that participants had in 2007? How about in 2011?

mean_children_2007 <- mean(data_2007$no_of_children, na.rm = TRUE)
mean_children_2011 <- mean(data_2011$no_of_children, na.rm = TRUE)
```

Have a play around with the data to see what kinds of variables exist. You could use the summary() function to get an idea of the values for each response, or use filter() or select() in dplyr to extract subsets of the data.

```r
 # Have a play around! e.g.
data_explore <- eqls_data %>% filter(country == 'Croatia') %>%
  select(as_much_time_as_would_like_with_family_members,
         worklife_balance_conflict, who5_mental_wellbeing_index)

# These might be some responses that you could explore the relationships between in the following quest
```

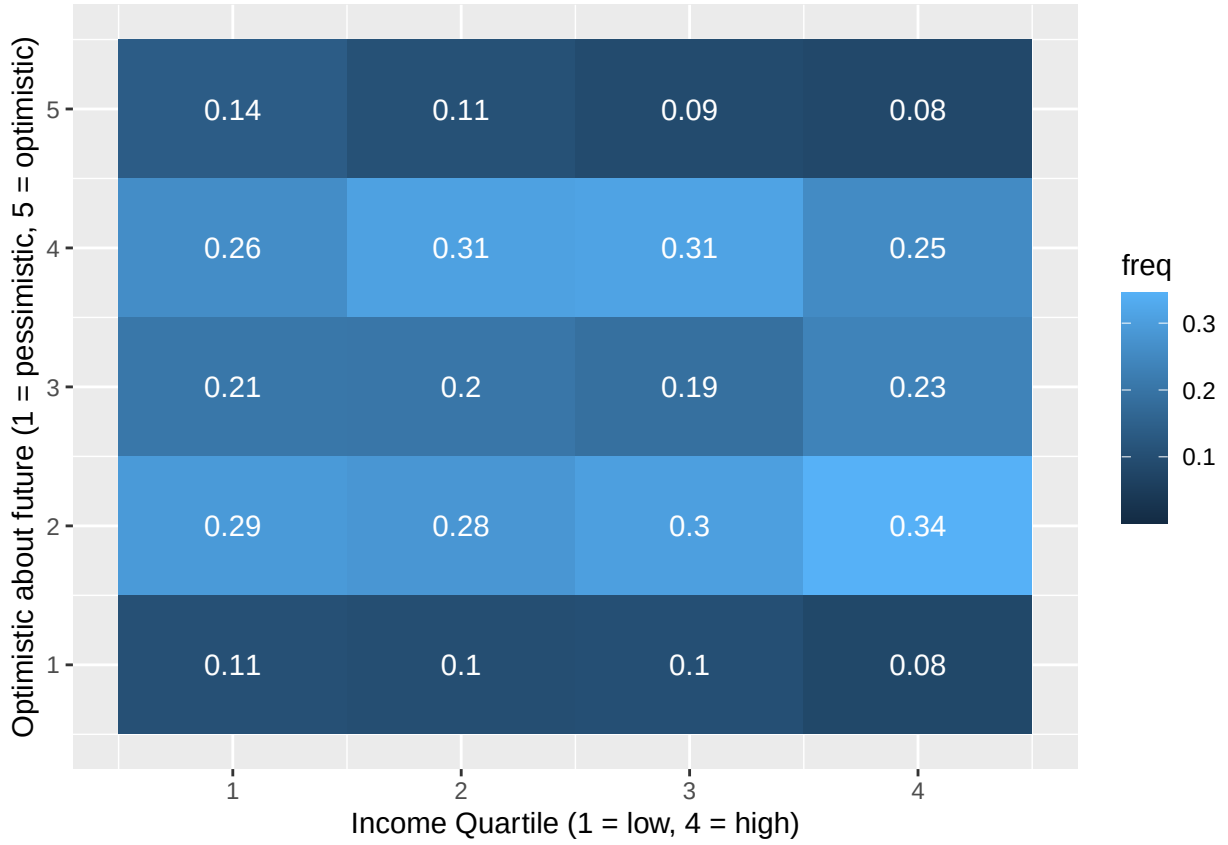
## Exercise 4: EQLS survey - single country

Now choose a single country to work with, at least for the time being. Choose some variables and produce plots using ggplot2 to study the associations between two different repsonses. Choose two responses from the columns the dataframe. You can use the file eqls_2007and2011_ukda_data_dictionary.rtf as a dictionary to find out a bit more about what these variables mean.

Try using different kinds of plots as well, e.g. histograms/density plots, heatmaps, violin plots, or experiment with other plots available in ggplot.

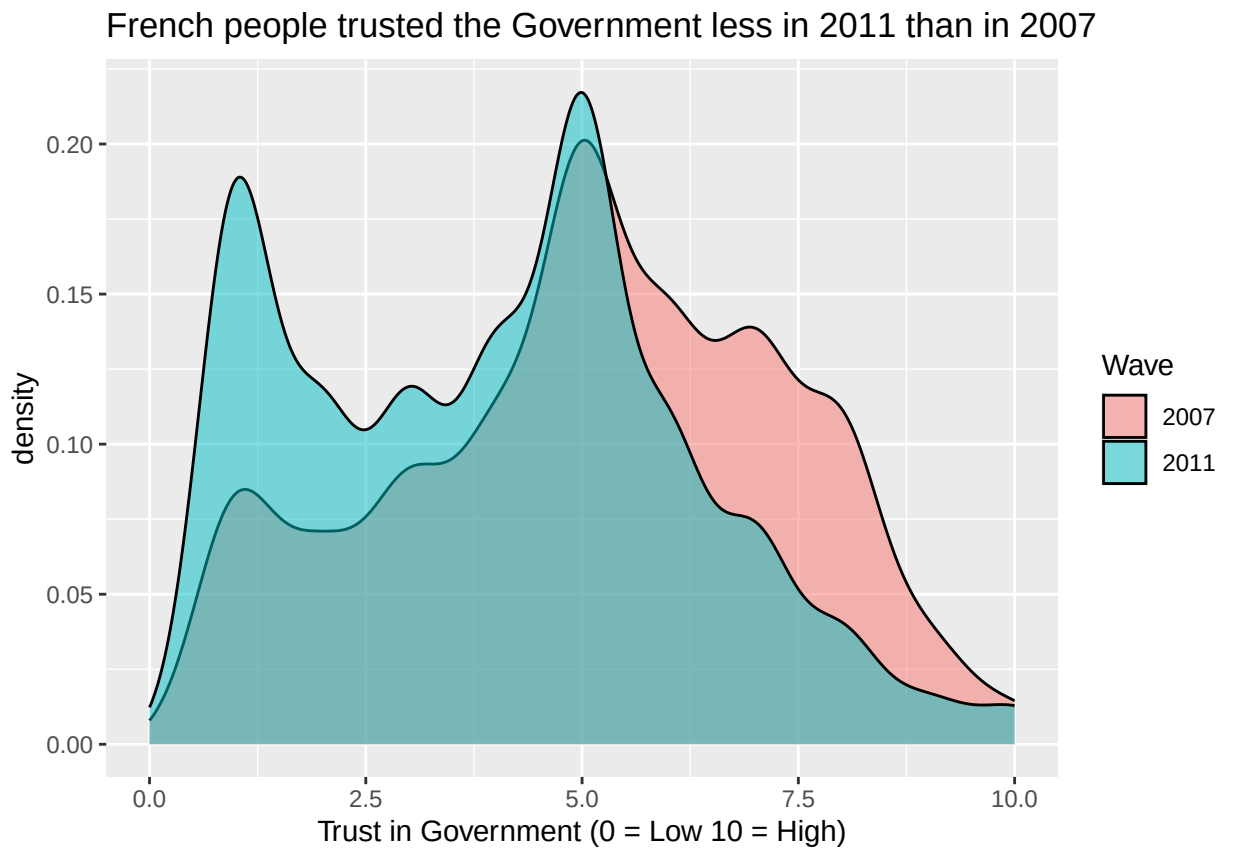
```r
# Let's take the data for France, as an example - use filter in dplyr
data_france <- eqls_data %>% filter(country == 'France')

### We will look at the relationship between an individual's income bracket and
### how optimistic they feel about the future.

# Build a heatmap as we did for the diamonds dataset in the lectures.
# First, we need to get the counts by income quartile and how optimistic they feel.
heatmap <- data_france %>% count(income_quartiles, i_am_optimistic_about_the_future) %>%
  group_by(income_quartiles) %>%
  mutate(freq = n/sum(n)) %>% # Get the frequencies of responses within income groups
  ggplot(aes(x = income_quartiles, y = i_am_optimistic_about_the_future)) +
```

```
  geom_tile(aes(fill = freq)) + xlab("Income Quartile (1 = low, 4 = high)") +
  ylab('Optimistic about future (1 = pessimistic, 5 = optimistic)') +
  geom_text(aes(label=round(freq, 2)), color = 'white') # add the labels for the frequencies
show(heatmap)
```




Now produce a plot that compares the repsonses to a single question across different years of the survey in one country. You can choose a different response to the ones that you chose for the previous question. How have the responses changed between 2007 and 2011?

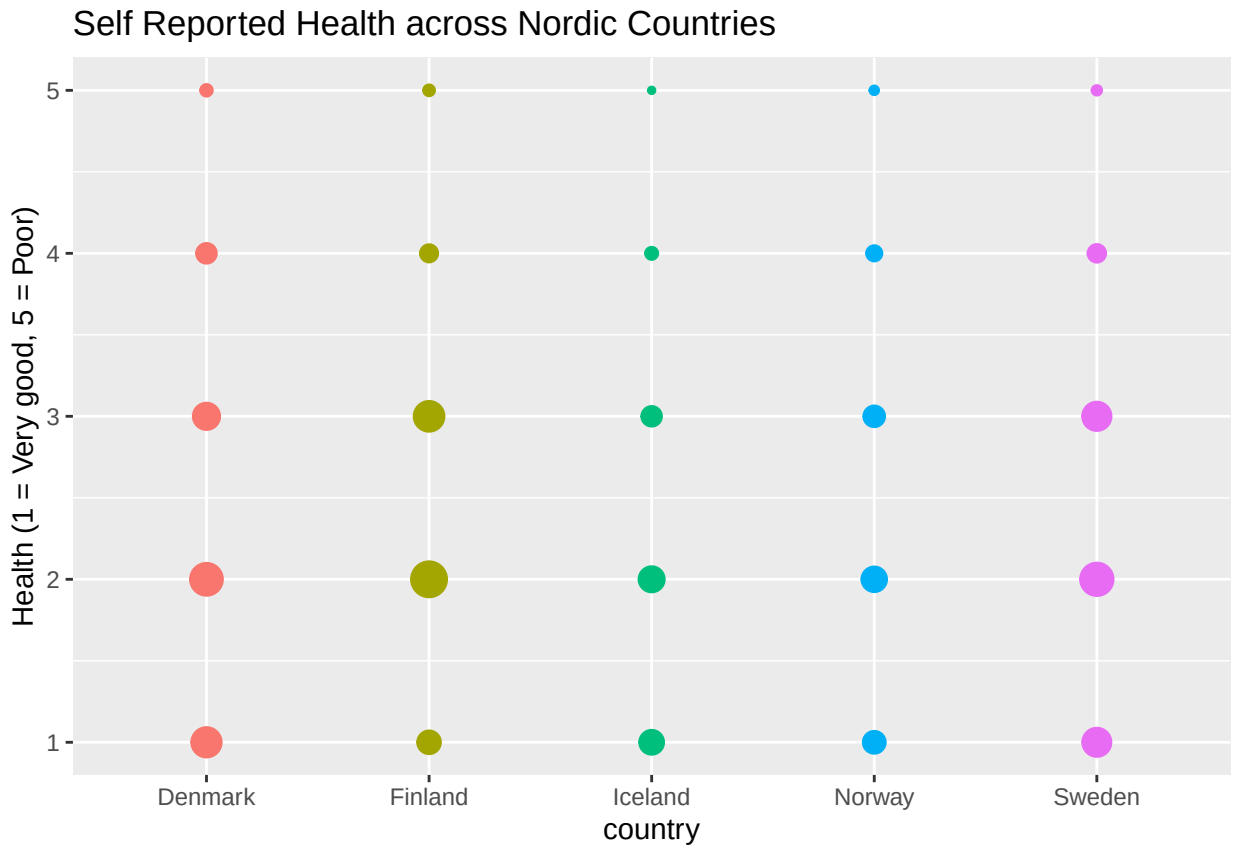
```
densityplot <- ggplot(data = data_france,
                      aes(x = how_much_trust_the_government,
                        fill = as.factor(eqls_wave))) +
  geom_density(alpha = 0.5) +
  scale_fill_discrete(name = "Wave") +
  xlab('Trust in Government (0 = Low 10 = High)') + xlim(0, 10) +
  ggtitle('French people trusted the Government less in 2011 than in 2007')
show(densityplot)
```



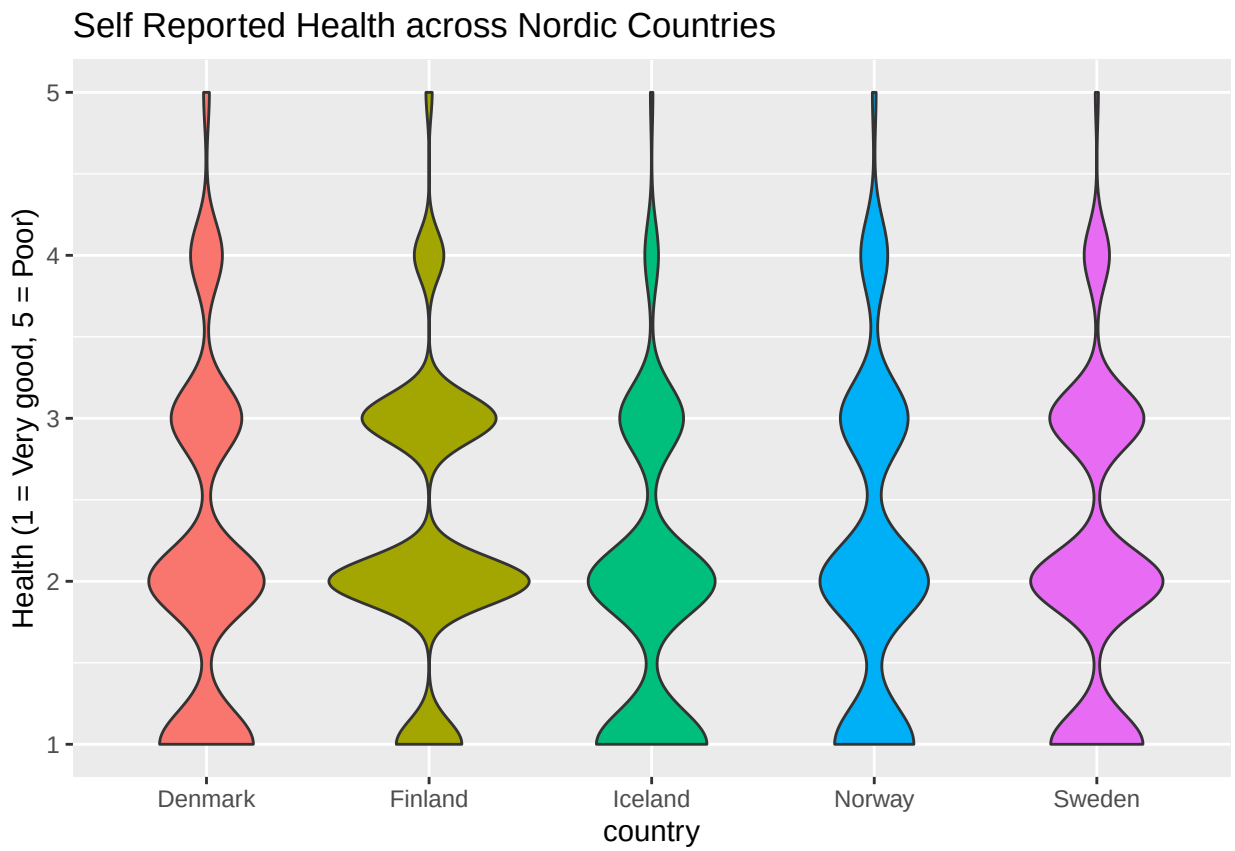

## Exercise 5: EQLS survey - multiple countries:

Now, investigate the responses to a single question from a group of different countries. For example, you could look at self-reported health in a few Nordic countries (the corresponding column for self reported health is health_condition)

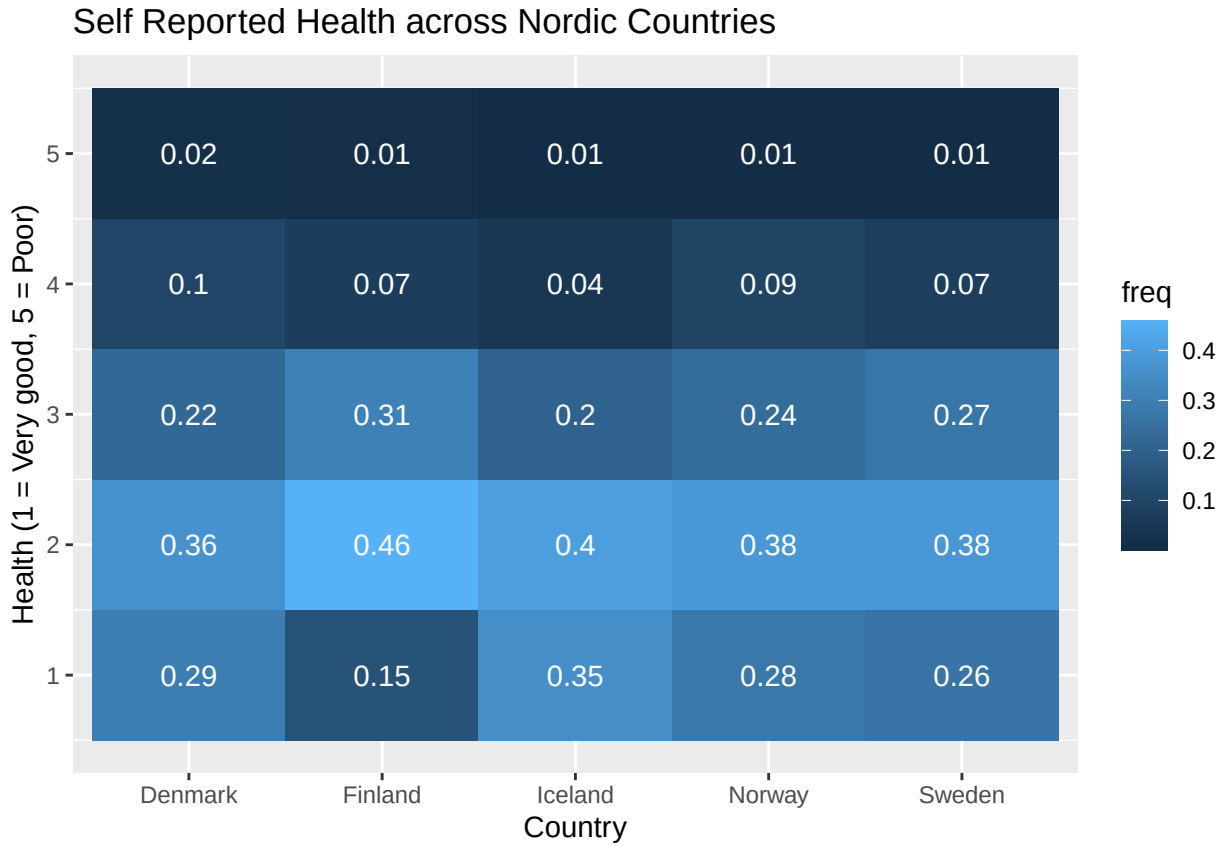
```
# Data from Nordic countries
data_region <- eqls_data %>% filter(country %in% c('Denmark', 'Sweden', 'Norway', 'Finland', 'Iceland')

# Ideally, for categorical data, we should use a count plot to get the counts
# in each category

count_plot_health <- ggplot(data_region, aes(x = country,
                                             y = health_condition,
                                             color = as.factor(country)))  +
  geom_count(show.legend = FALSE) + ylim(1, 5) +
  ylab('Health (1 = Very good, 5 = Poor)') +
  ggtitle('Self Reported Health across Nordic Countries')
show(count_plot_health)
```



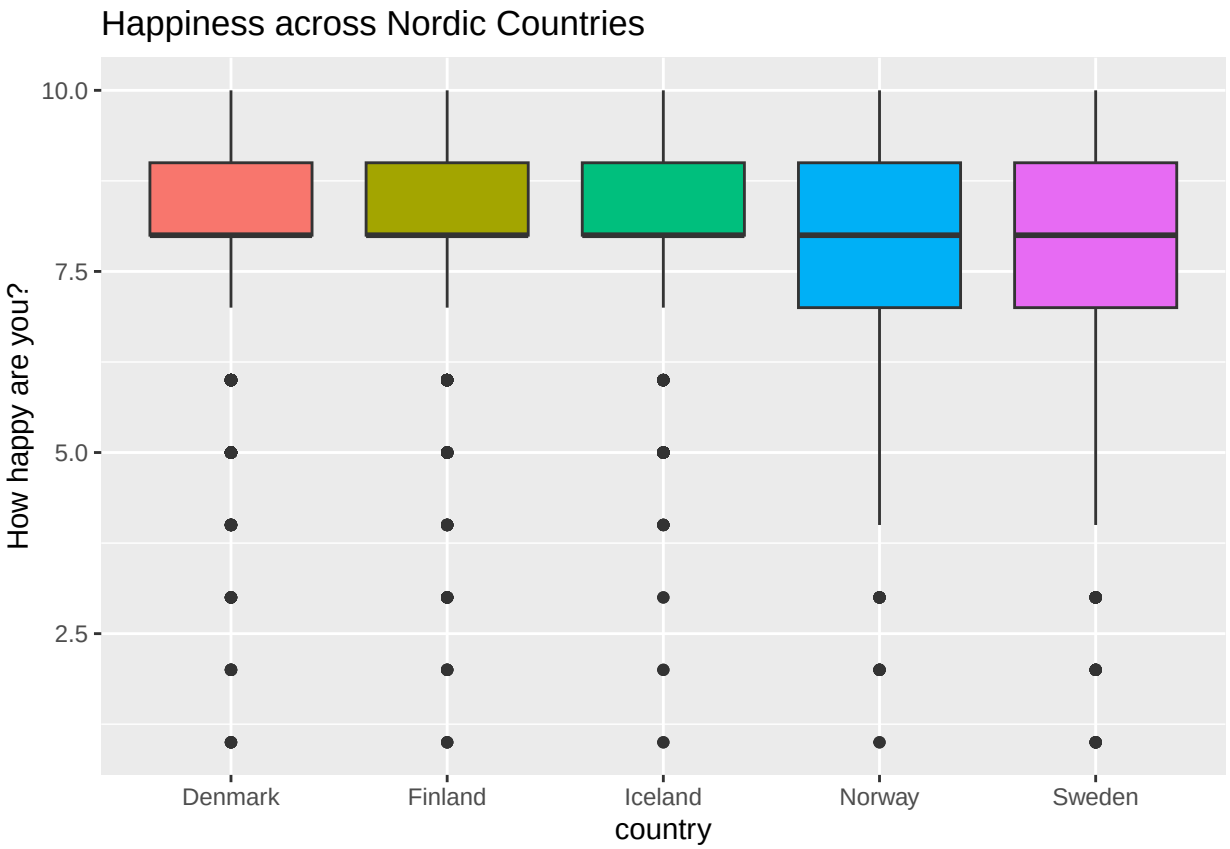

```
# But we could also use a violin plot, even though it's not strictly speaking
# approporiate for this type of data

violin_plot_health <- ggplot(data_region, aes(x = country,
                                              y = health_condition,
                                              fill = as.factor(country)))  +
  geom_violin(show.legend = FALSE) + ylim(1, 5) +
  ylab('Health (1 = Very good, 5 = Poor)')+
  ggtitle('Self Reported Health across Nordic Countries')

show(violin_plot_health)
```




```
# Finally, we could also produce a heatmap for this data:

data_region %>% count(country, health_condition) %>% group_by(country) %>%
  mutate(freq = n/sum(n)) %>%
  ggplot(aes(x = country, y = health_condition)) +
  geom_tile(aes(fill = freq)) + ylab('Health (1 = Very good, 5 = Poor)') +
  xlab('Country') +
  geom_text(aes(label=round(freq, 2)), color = 'white') +
  ggtitle('Self Reported Health across Nordic Countries')
```




```
# We could also produce a boxplot, this time for the question
# "How happy are you?"
box_plot_happy <- ggplot(data_region, aes(x = country,
                                          y = how_happy_are_you,
                                          fill = as.factor(country)))  +
  geom_boxplot(show.legend = FALSE) + ylim(1, 10) + ylab('How happy are you?') +
  ggtitle('Happiness across Nordic Countries')

show(box_plot_happy)
```




## Exercise 6: EQLS survey - all countries:

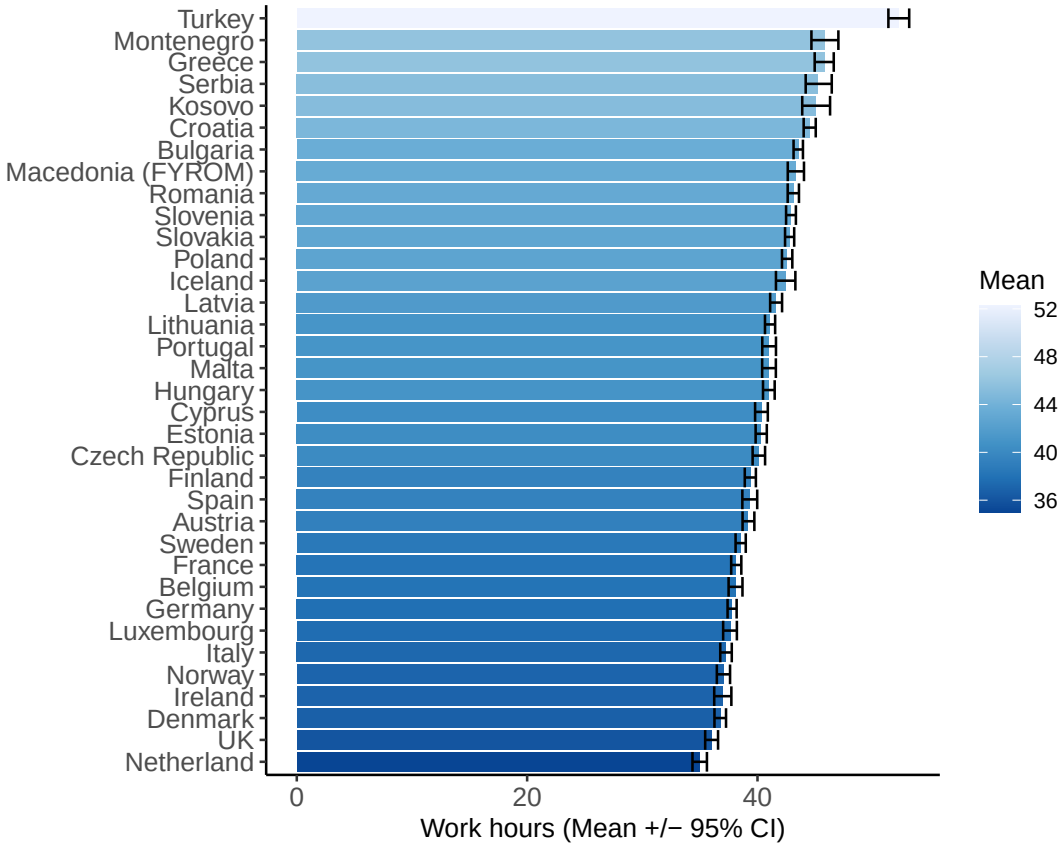
Now we will compare responses across all countries included in the survey. Draw a bar plot of the average number of work hours in the 1st job for each country.

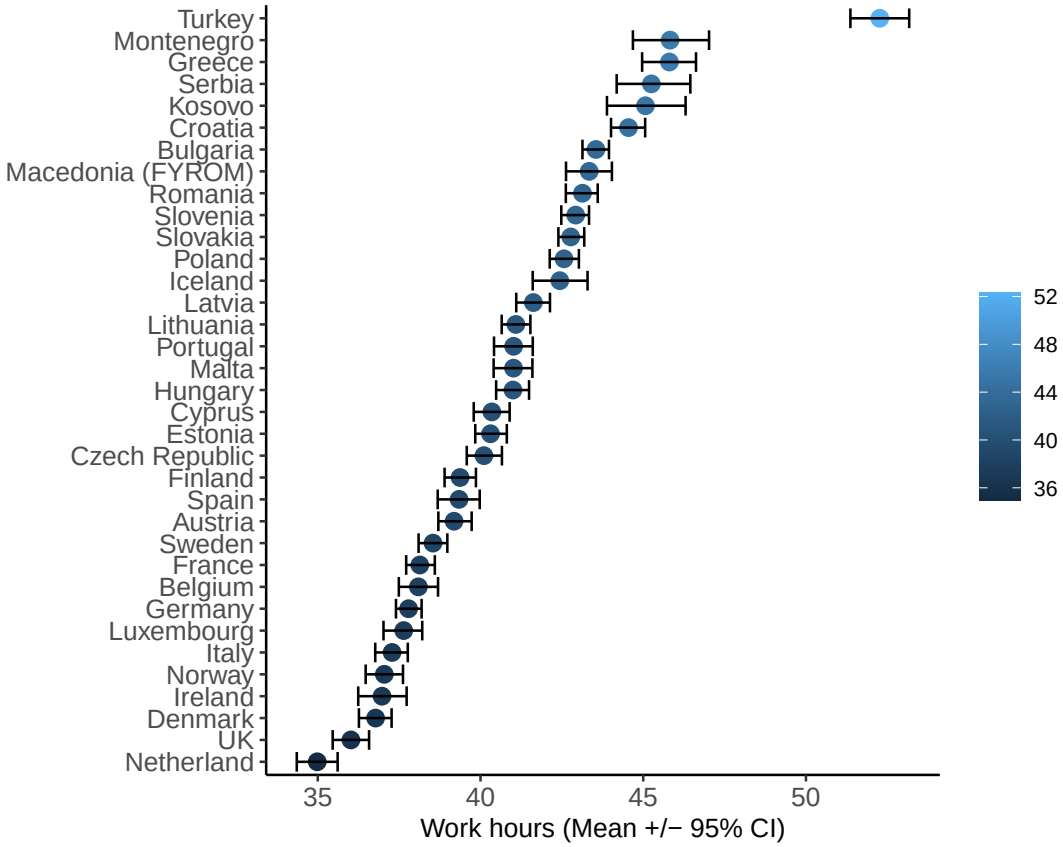
- Sort the bars by their value in descending order
- Include error bars to represent uncertainty, e.g., standard deviation or 95% bootstrapped confidence intervals (hint: use the geom_errorbar() function added in as a layer to your ggplot)
- Add a sequential palette such that higher (resp. lower) work hours are coloured with a lighter (resp. darker) colour

Optional: bars take a lot of (unnecessary?) ink. How can you optimise the data-to-ink ratio here?

```
summary_work_hours_first_job <- eqls_data %>%
  # Select how many hours they work in 1st job and the country name
  select(how_many_hours_work_per_week_in_1st_job, country) %>%
  # Group by country name
  group_by(country) %>%
  # Find confidence intervals
  summarise(data = list(smean.cl.boot(cur_data(), conf.int = .95, B = 1000L, na.rm = TRUE))) %>%
  tidyr::unnest_wider(data)

fig1 <- ggplot(summary_work_hours_first_job, aes(x = Mean, y = reorder(country, Mean))) +
  # A bar plot, x = Mean number of hours, y = country ordered by their mean (descending)
  geom_bar(stat = "identity", aes(fill = Mean)) +
  # Error bars, Lower = 95% CI lower bound, Upper = 95% CI upper bonud
  geom_errorbar(aes(xmin = Lower, xmax = Upper, y = country)) +
```

```
  # Palette of blues
  scale_fill_distiller(type = "seq", palette = "Blues", direction = -1) +
  # Classic theme
  theme_classic() +
  # Change x and y axis format
  theme(
    axis.text.x = element_text(size = 11L),
    axis.text.y = element_text(size = 11L)
  ) +
  # Change x and y axis labels
  labs(
    x = "Work hours (Mean +/- 95% CI)",
    y = "",
  )

# We can just change the first layer (the bar plot) to a scatter plot!
fig2 <- fig1
fig2$layers[[1]] <- geom_point(aes(color = Mean), size = 3L)

# Subplots with 1 column
grid.arrange(fig1, fig2, ncol = 1L)
```


Turkey
Montenegro
Greece
Serbia
Kosovo
Croatia
Bulgaria
Macedonia (FYROM)
Romania
Slovenia
Slovakia
Poland
Iceland
Latvia
Lithuania
Portugal
Malta
Hungary
Cyprus
Estonia
Czech Republic
Finland
Spain
Austria
Sweden
France
Belgium
Germany
Luxembourg
Italy
Norway
Ireland
Denmark
UK
Netherland
0
20
40
Work hours (Mean +/− 95% CI)
Mean
52
48
44
40
36


Turkey
Montenegro
Greece
Serbia
Kosovo
Croatia
Bulgaria
Macedonia (FYROM)
Romania
Slovenia
Slovakia
Poland
Iceland
Latvia
Lithuania
Portugal
Malta
Hungary
Cyprus
Estonia
Czech Republic
Finland
Spain
Austria
Sweden
France
Belgium
Germany
Luxembourg
Italy
Norway
Ireland
Denmark
UK
Netherland
35
40
45
50
Work hours (Mean +/− 95% CI)
52
48
44
40
36