`ggplot` Overview

Monica Alexander

17 January 2024

Contents

1 Overview

This document gives a few more examples on how to use `ggplot` to make a range of graphics.

- `ggplot` is the graphing package that goes with the `tidyverse` in R
- Very powerful to make a wide range of graphics
- Every graph so far this lecture was done in `ggplot`
- `ggplot` code works in layers, with each layer adding complexity
  - start with defining dataset and different variables
  - add on type of plot
  - scales
  - layout (facets)
  - themes, fonts, sizes...

2 Data used

For this lab we will use data from the 2017 Canadian General Social Survey, and a national-level dataset called `country_indicators`, which includes a range of information on mortality and other indicators for each country. Read them in:

```
library(tidyverse)
library(here)

gss <- read_csv(here("data/gss.csv"))
country_ind <- read_csv(here("data/country_indicators.csv"))
```

3 Introductory example

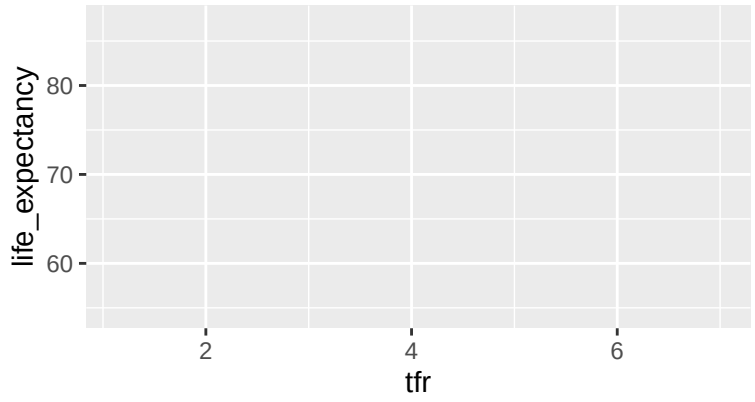
Let's make a scatter plot of TFR versus life expectancy chart, colored by region, for the year 2017. For simplicity, filter the data to just include 2017:

```
country_ind_2017 <- country_ind %>% filter(year==2017)
```

3.1 A blank canvas

`aes` stands for aesthetic and tells ggplot the main characteristics of your plot (x, y, and if the color or fill vary by group)

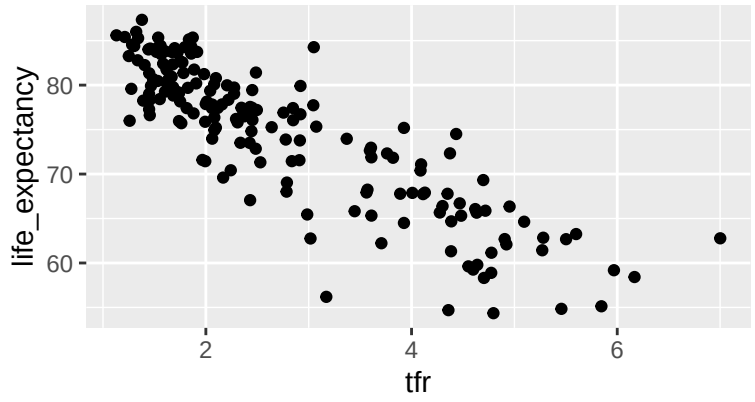
```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy))

#print
plot1
```



3.2 Add the points

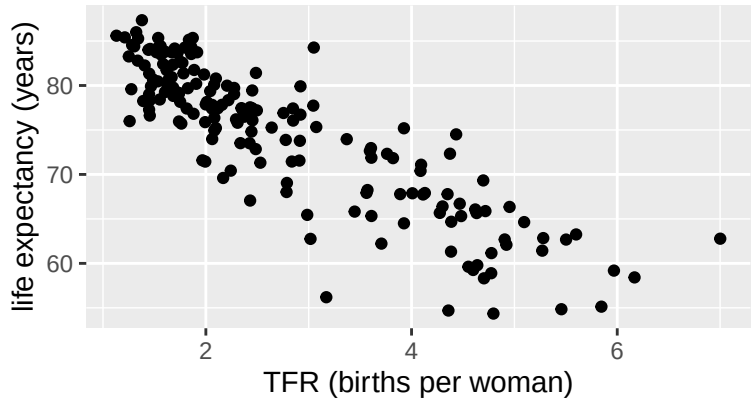
Add layers with ggplot using the +

```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy)) +
  geom_point()

plot1
```



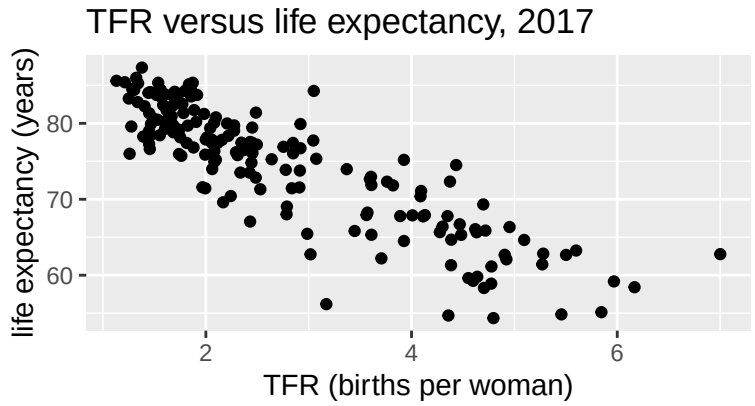
3.3 Tidy up labels

```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy)) +
  geom_point()+
  xlab("TFR (births per woman)")+
  ylab("life expectancy (years)")

plot1
```



3.4 Add a title

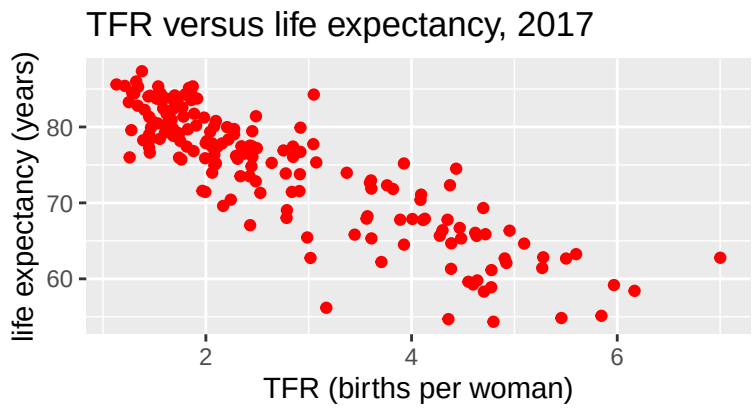
```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy)) +
  geom_point()+
  xlab("TFR (births per woman)")+
  ylab("life expectancy (years)")+
  ggtitle("TFR versus life expectancy, 2017")

plot1
```



3.5 Change color of points

to see all colors, type `colors()`

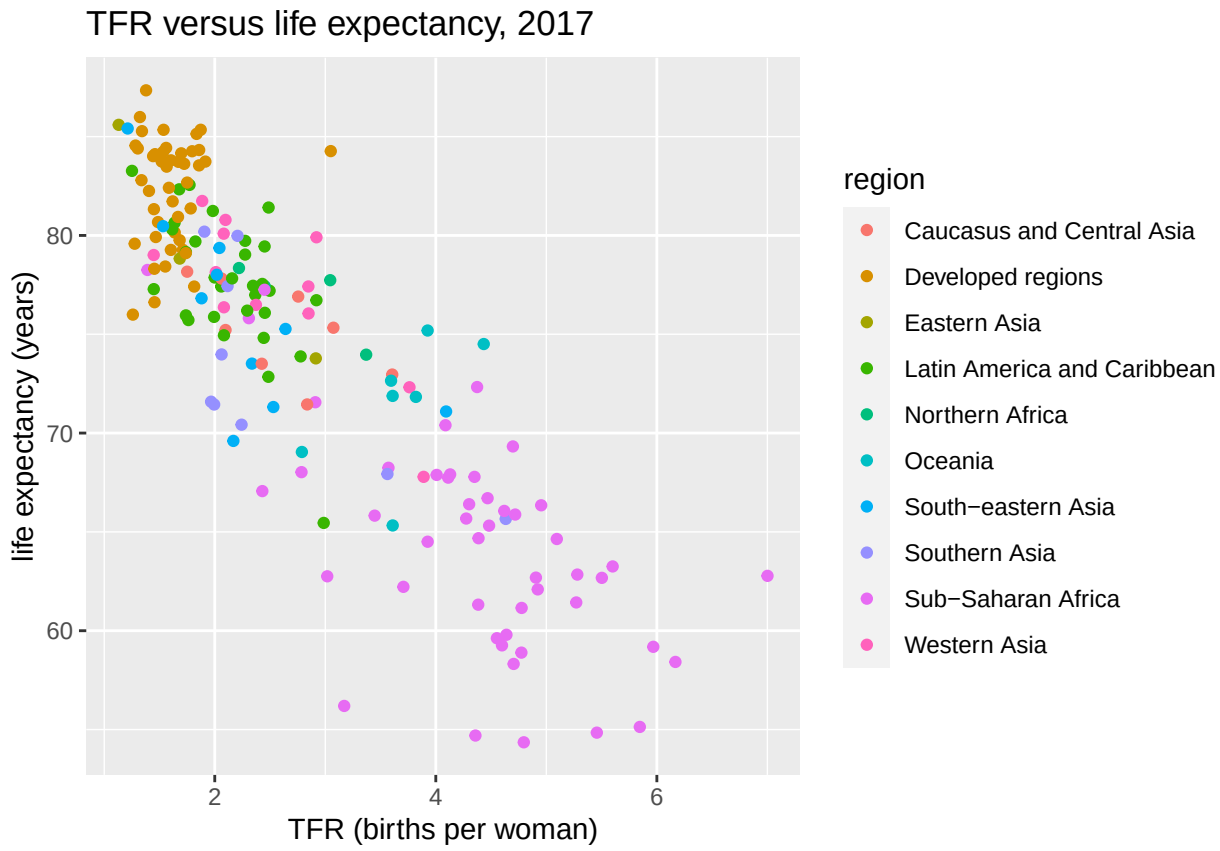
```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy)) +
  geom_point(color = "red")+
  xlab("TFR (births per woman)")+
  ylab("life expectancy (years)")+
  ggtitle("TFR versus life expectancy, 2017")

plot1
```



3.6 Coloring by group

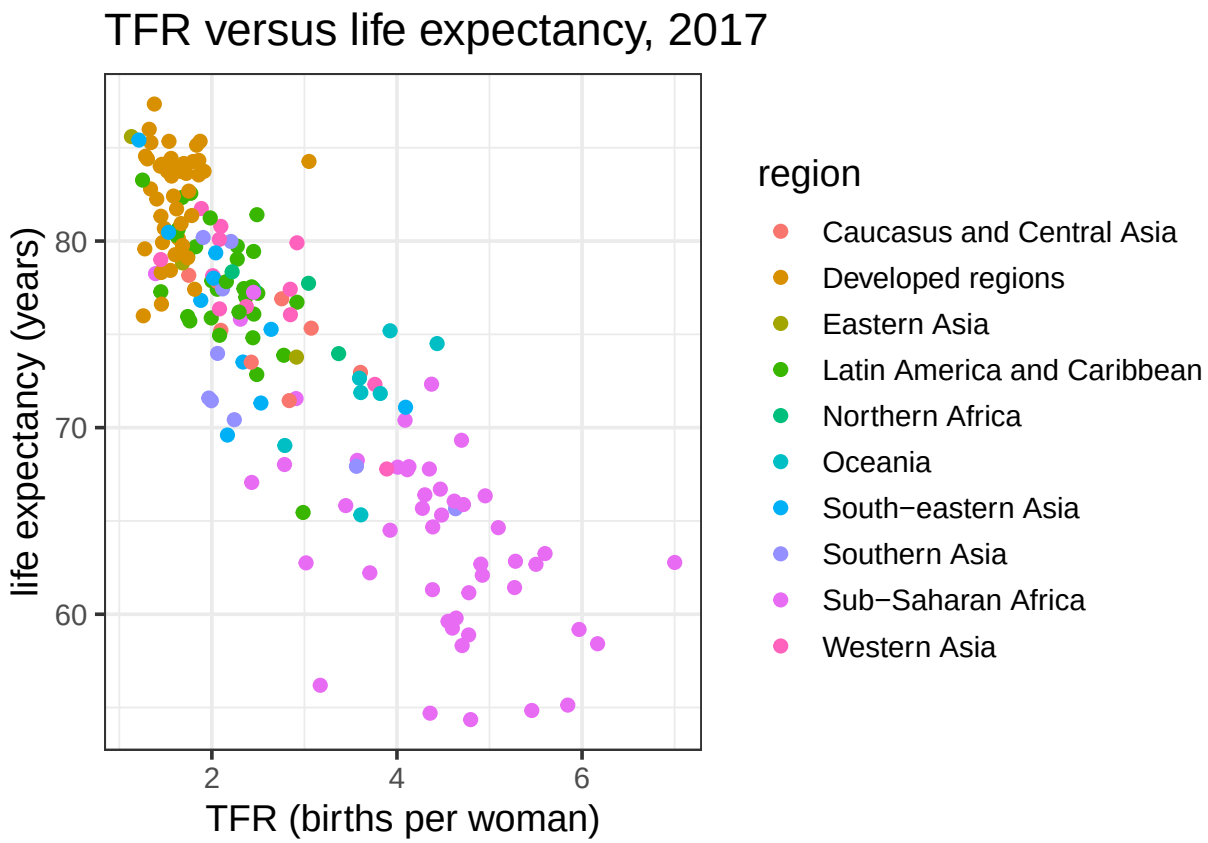
This goes in the `aes()` because it **depends on the data**

```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy, color = region)) +
  geom_point()+
  xlab("TFR (births per woman)")+
  ylab("life expectancy (years)")+
  ggtitle("TFR versus life expectancy, 2017")

plot1
```



3.7 Change theme (optional) and size of points

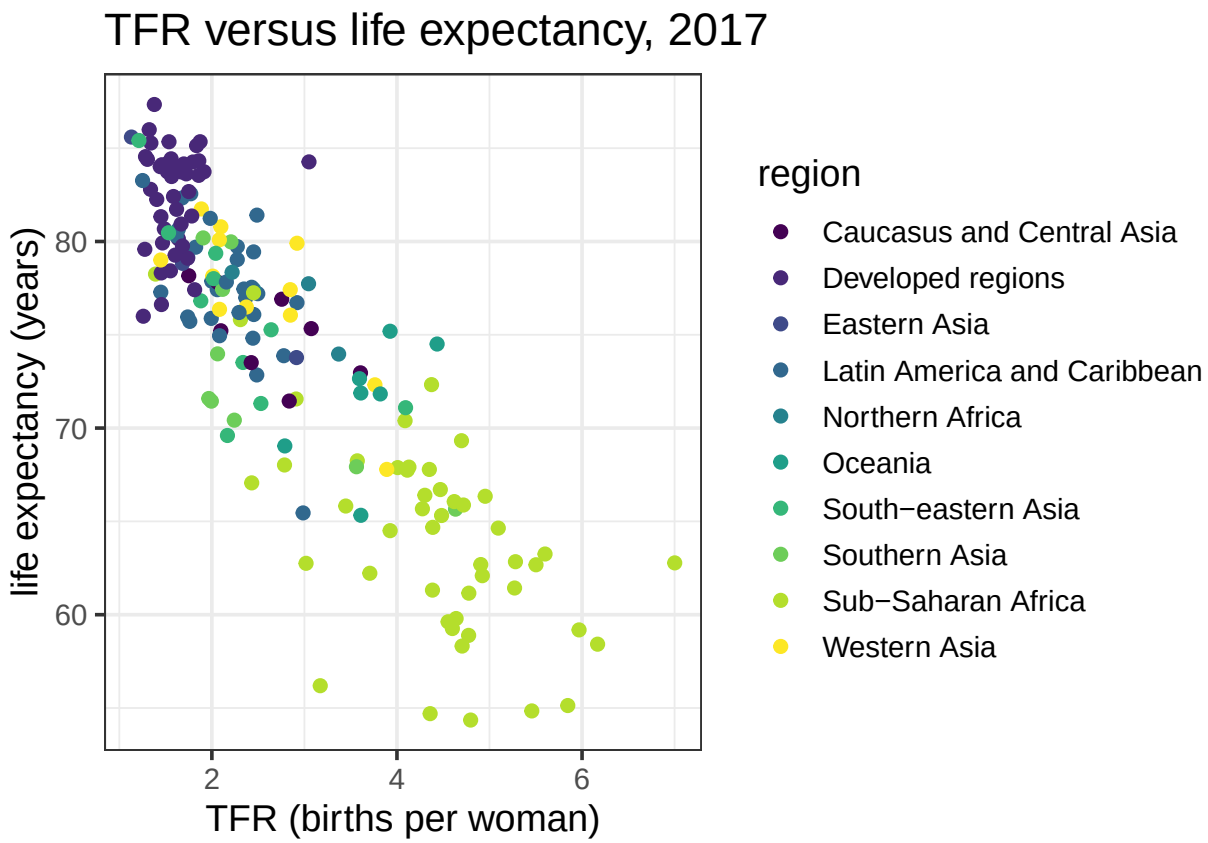
```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy, color = region)) +
  geom_point(size =2)+
  xlab("TFR (births per woman)")+
  ylab("life expectancy (years)")+
  ggtitle("TFR versus life expectancy, 2017")+
  theme_bw(base_size = 14)

plot1
```



3.8 Change color scheme

`viridis` and `brewer` both good options

```
plot1 <- ggplot(data = country_ind_2017, aes(x = tfr, y = life_expectancy, color = region)) +
  geom_point(size =2)+
  xlab("TFR (births per woman)")+
  ylab("life expectancy (years)")+
  ggtitle("TFR versus life expectancy, 2017")+
  theme_bw(base_size = 14)+
  scale_color_viridis_d()

plot1
```



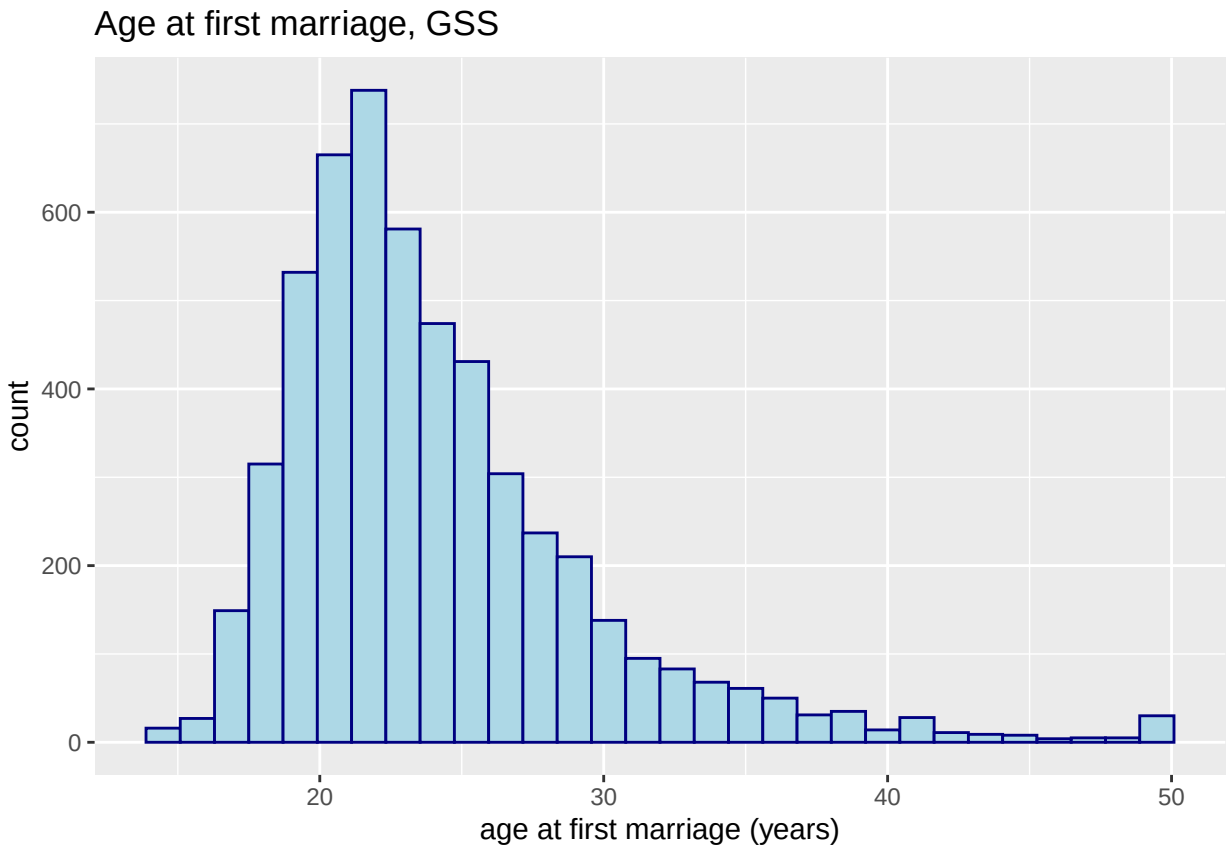
4 Plot Types

This section briefly illustrates how to do the main types of plots in `ggplot`: histograms, bar charts, box plots, line graphs and scatter plots. Finally, faceting is illustrated.

4.1 Histograms

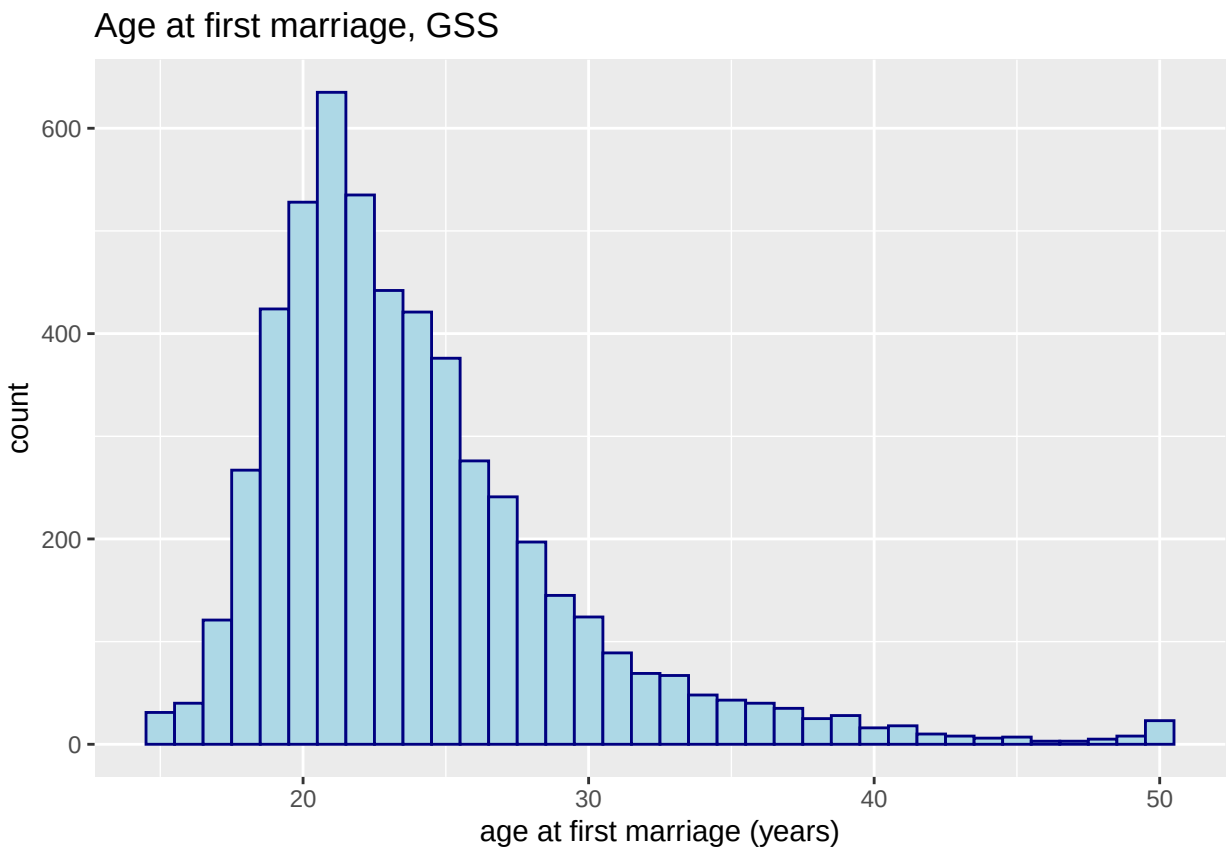
Note for histograms, bar chats, box plots, `fill` is the main color choice (`color` changes the outline)

```
ggplot(data = gss, aes(age_at_first_marriage)) +
  geom_histogram(fill = "lightblue", color = "navy") +
  ggtitle("Age at first marriage, GSS") +
  xlab("age at first marriage (years)")
```

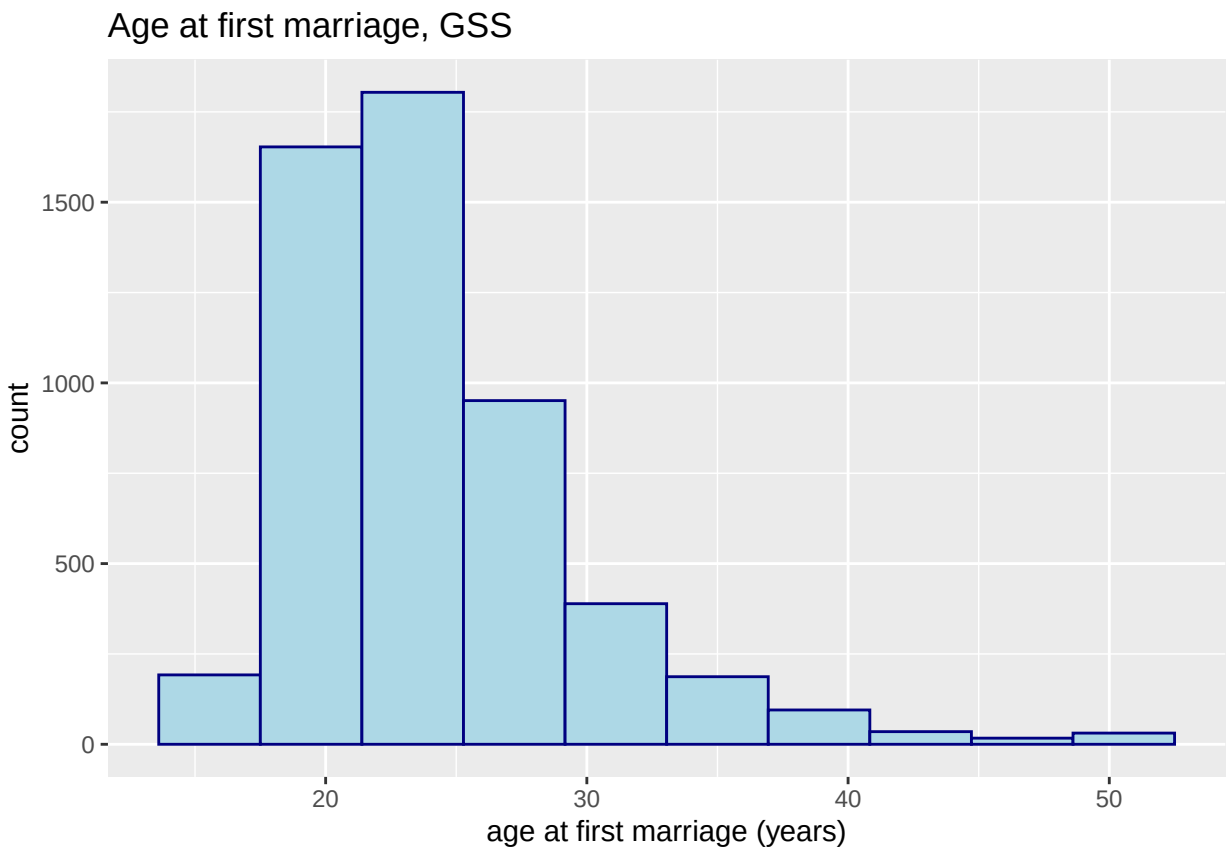


Histograms select a `binwidth` or section of the data and then count how many of the observations fall within that. Histograms look different depending on the size of the bins. You can also supply the number of bins that you want to create.

```
ggplot(data = gss, aes(age_at_first_marriage)) +
  geom_histogram(fill = "lightblue", color = "navy", binwidth = 1) +
  ggtitle("Age at first marriage, GSS") +
  xlab("age at first marriage (years)")
```

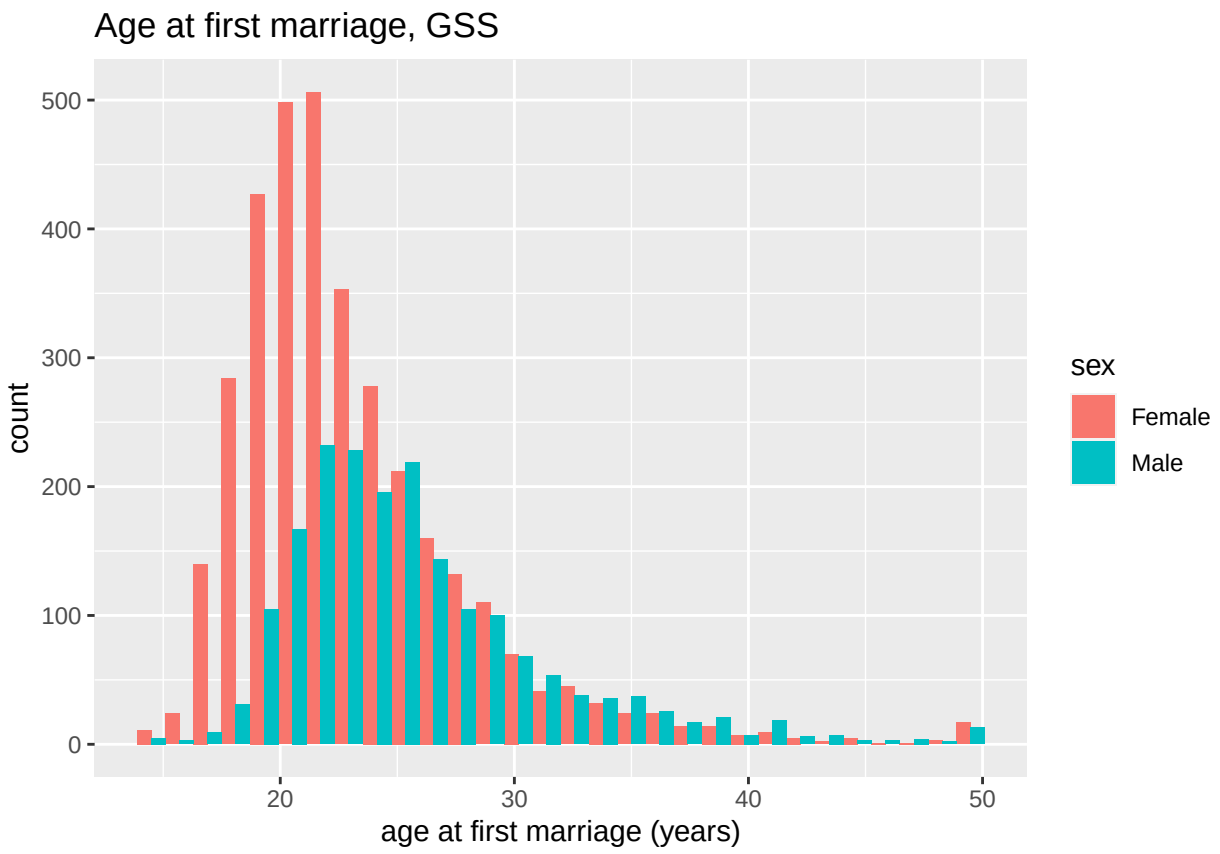



```
ggplot(data = gss, aes(age_at_first_marriage)) +
  geom_histogram(fill = "lightblue", color = "navy", bins = 10)+
  ggtitle("Age at first marriage, GSS") +
  xlab("age at first marriage (years)")
```



We can also plot by another variable to compare the plots by the categories of the variable. For example, we look at plots by sex:

```
ggplot(data = gss, aes(age_at_first_marriage, fill = sex)) +
  geom_histogram(position = 'dodge') +
  ggtitle("Age at first marriage, GSS") +
  xlab("age at first marriage (years)")
```



4.2 Bar charts

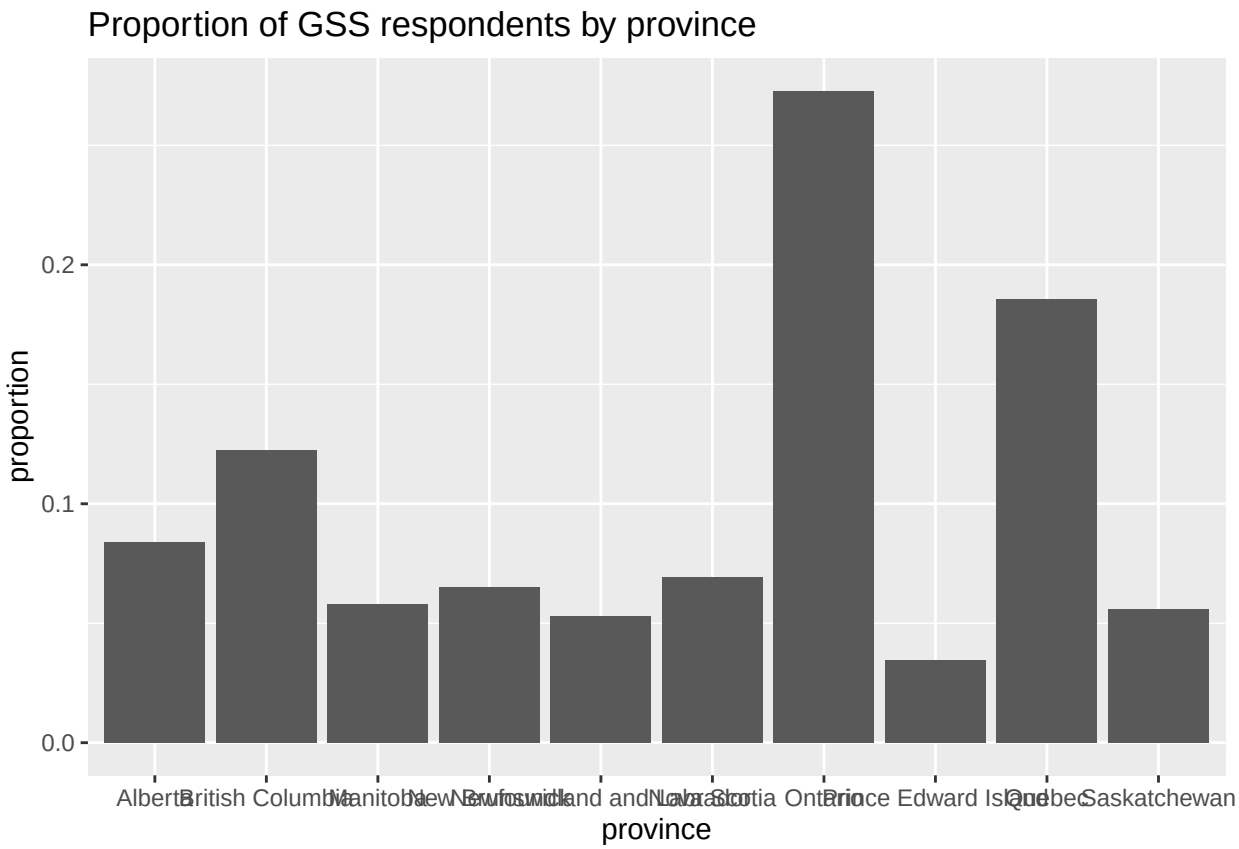
Let's plot the proportion of respondents by province as a bar chart. First save the proportions as a new data frame

```
resp_by_prov <- gss %>%
  group_by(province) %>%
  tally() %>%
  mutate(prop = n / sum(n))

summary(resp_by_prov)
```

```
##    province                 n             prop
##  Length:10          Min.   : 708   Min.   :0.03437
##  Class :character   1st Qu.:1163   1st Qu.:0.05644
##  Mode  :character   Median :1381   Median :0.06703
##                     Mean   :2060   Mean   :0.10000
##                     3rd Qu.:2324   3rd Qu.:0.11278
##                     Max.   :5621   Max.   :0.27284
```

Now plot

```r
ggplot(data = resp_by_prov, aes(x = province, y = prop)) +
  geom_bar(stat = "identity") +
  ylab("proportion")+
  ggtitle("Proportion of GSS respondents by province")
```

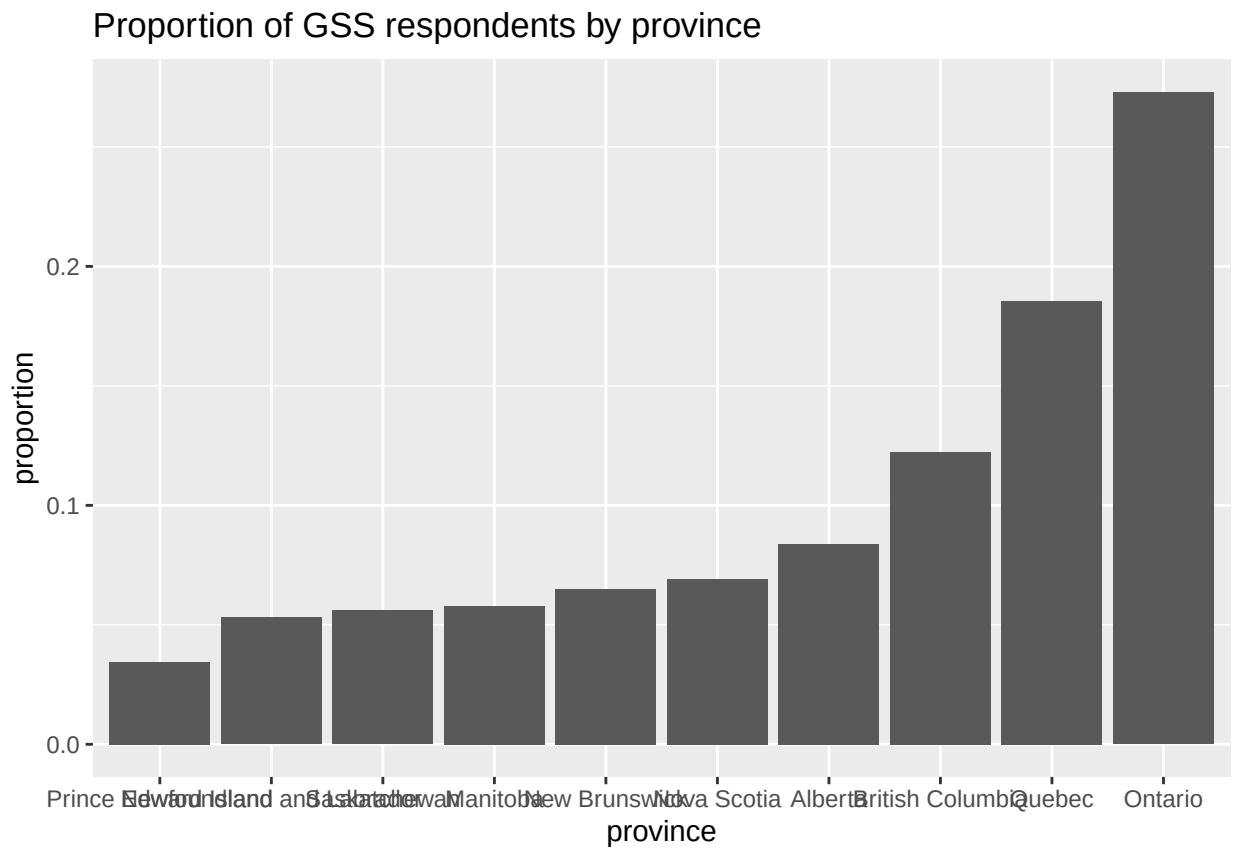


There are a few things here that would be nice to fix. Firstly, the categories are ordered alphabetically, which is the default. It would be better visually to order by proportion. We can do this using the `fct_reorder` function to alter (mutate) the province variable.

```r
resp_by_prov <- resp_by_prov %>%
  mutate(province = fct_reorder(province, prop)) # order by proportion
```

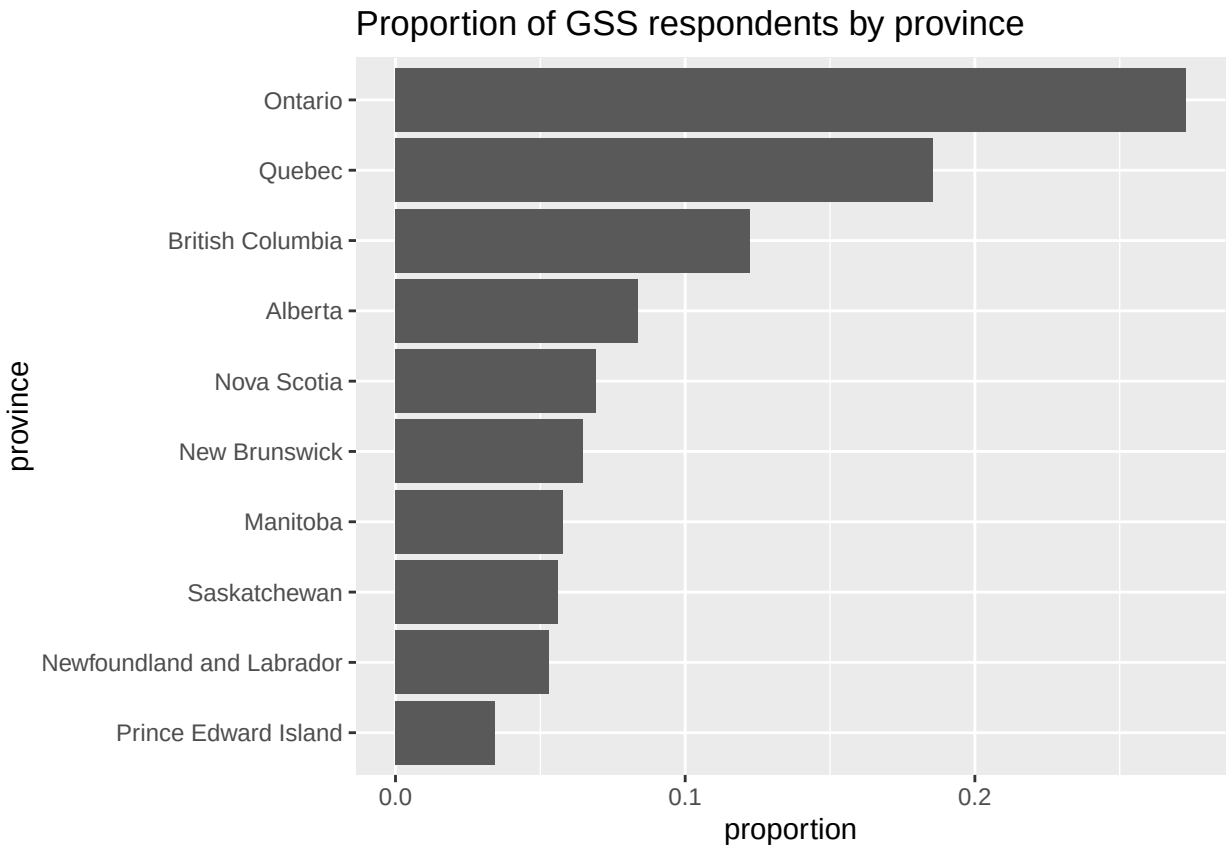
Now try plotting again.

```r
ggplot(data = resp_by_prov, aes(x = province, y = prop)) +
  geom_bar(stat = "identity") +
  ylab("proportion")+
  ggtitle("Proportion of GSS respondents by province")
```



To improve readability, could change to horizontal bar chart.

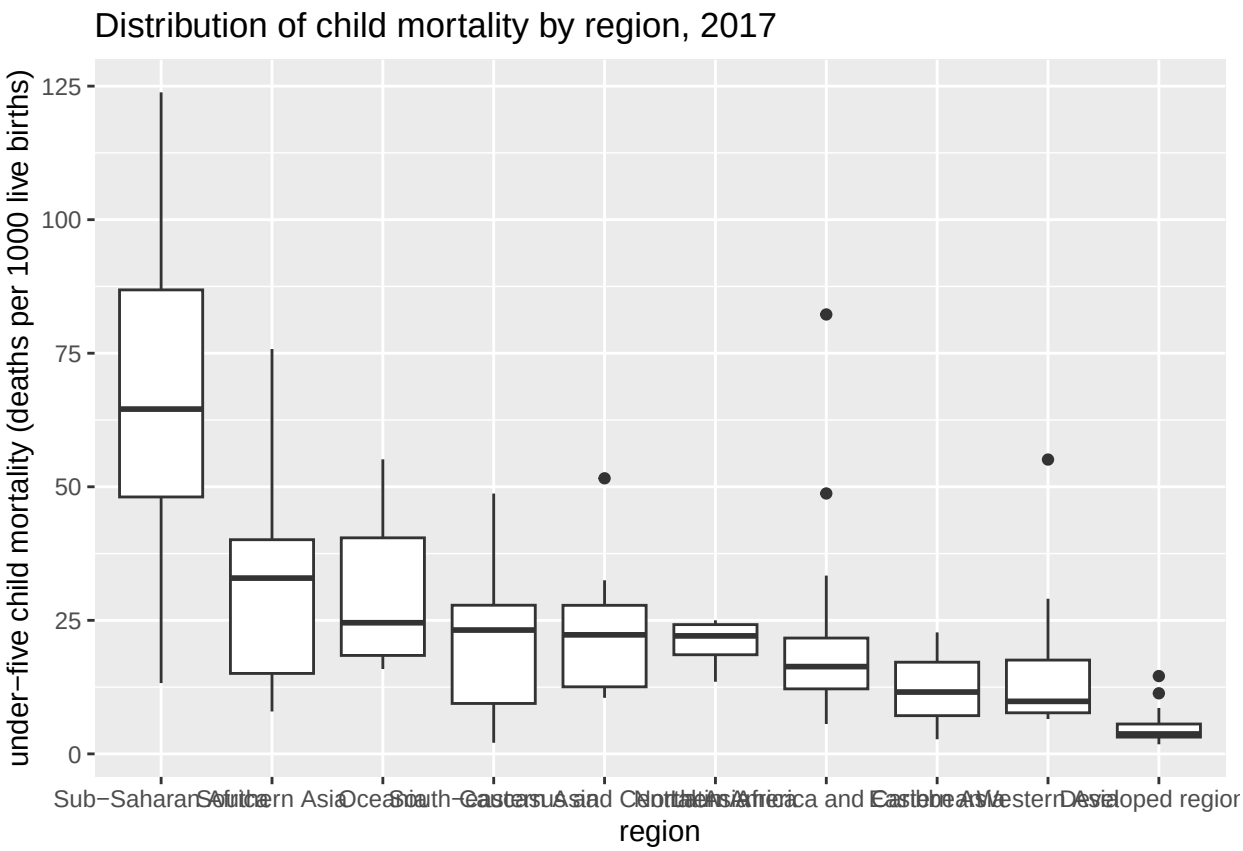
```
ggplot(data = resp_by_prov, aes(x = province, y = prop)) +
  geom_bar(stat = "identity") +
  ylab("proportion")+
  ggtitle("Proportion of GSS respondents by province") +
  coord_flip()
```



4.3 Box plots

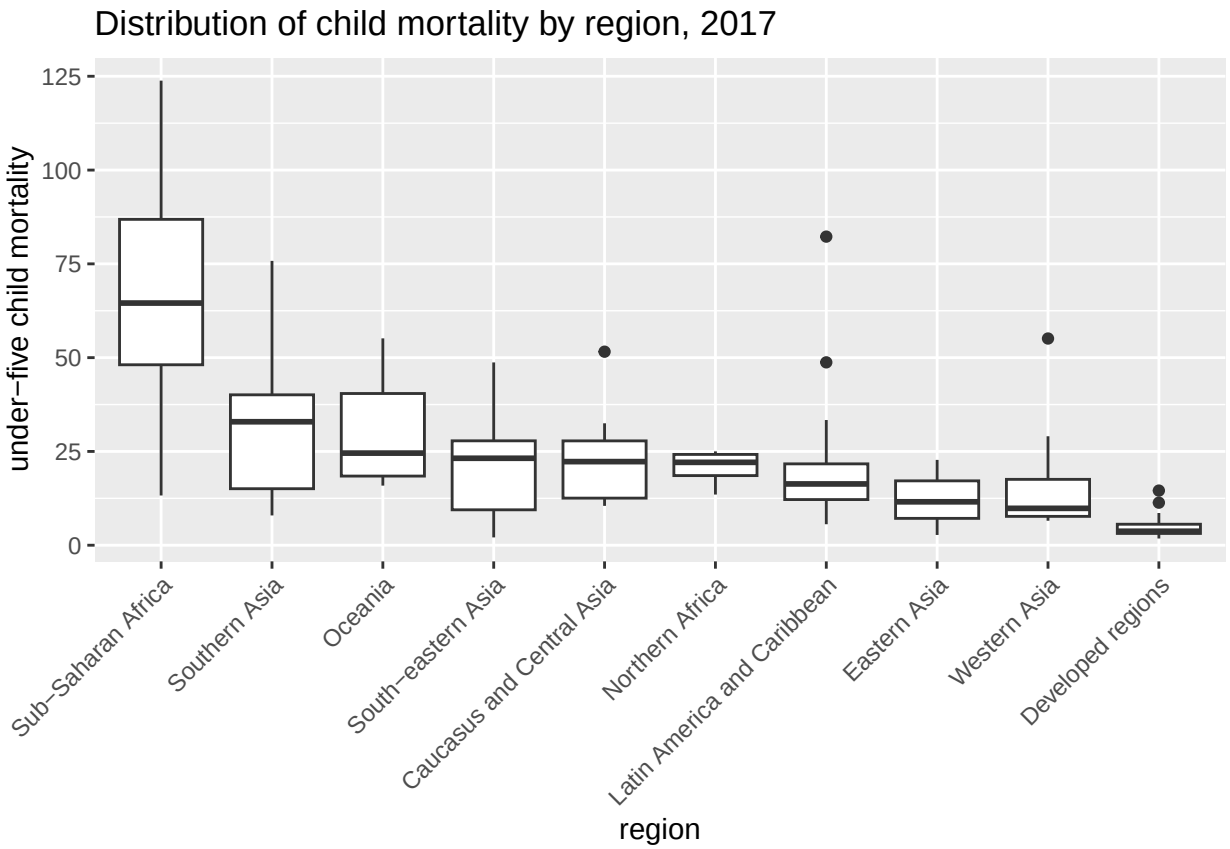
Let's use the country indicators dataset here and do boxplots of child mortality in 2017 over regions. Like the bar chart example, best to reorder the regions by the variable we are interested in

```
country_ind_2017 <- country_ind %>%
  filter(year==2017) %>%
  mutate(region = fct_reorder(region, -child_mort)) # descending order

ggplot(data = country_ind_2017, aes(x = region, y = child_mort)) +
  geom_boxplot() +
  ylab("under-five child mortality (deaths per 1000 live births)") +
  ggtitle("Distribution of child mortality by region, 2017")
```



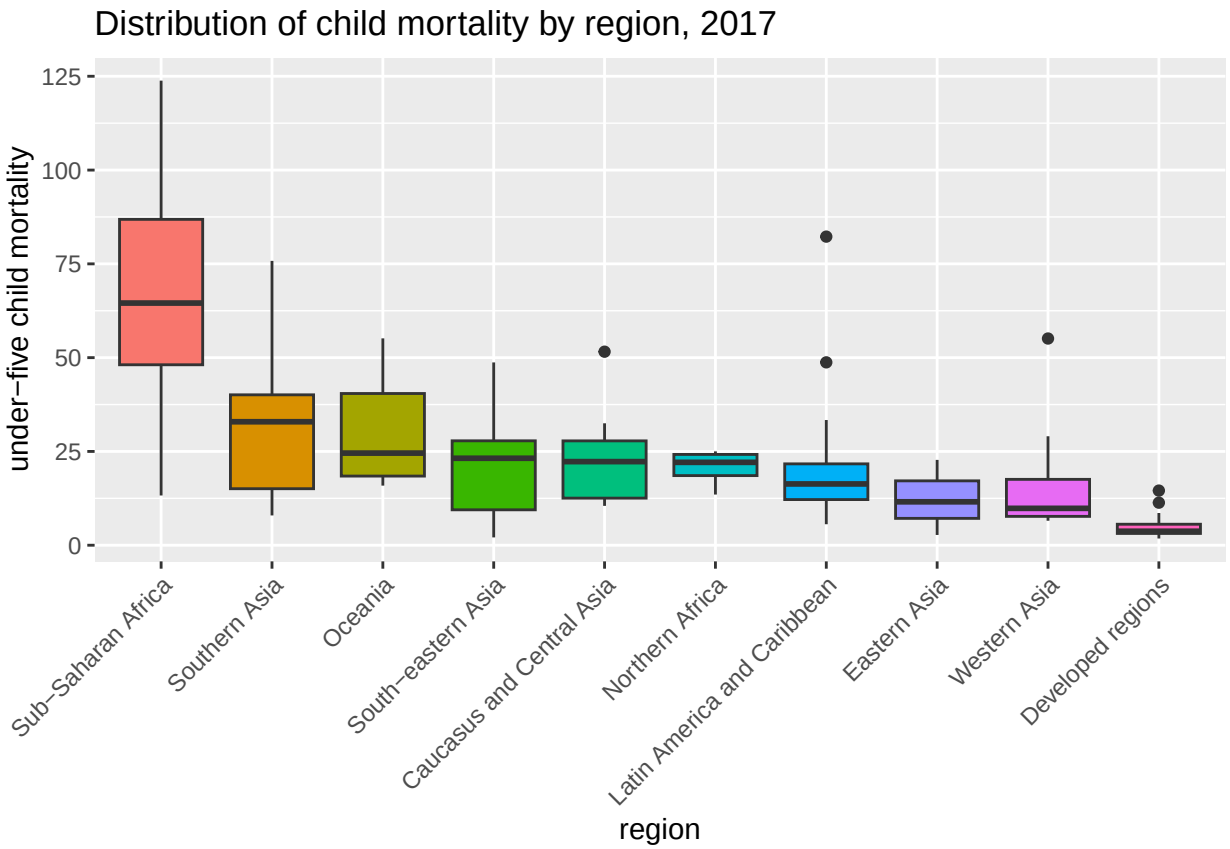
The labels on the x axis are hard to read. We could do the same as last time (switch to horizontal), or we can change the alignment of the labels:

```
ggplot(data = country_ind_2017, aes(x = region, y = child_mort)) +
  geom_boxplot() +
  ylab("under-five child mortality") +
  ggtitle("Distribution of child mortality by region, 2017") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Note if you want to color the boxes, use `fill`, and then remove the legend (not needed)

```
ggplot(data = country_ind_2017, aes(x = region, y = child_mort, fill = region)) +
  geom_boxplot() +
  ylab("under-five child mortality") +
  ggtitle("Distribution of child mortality by region, 2017") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1) ,
        legend.position = 'none')
```



4.4 Line graphs

Let's look at the mean age at marriage by age of respondent. Firstly, let's make a new variable in the `gss` dataset that groups people into 5-year age groups. Here's the code to do this:

```
age_groups <- seq(15, 80, by = 5)
gss$age_group <- as.numeric(as.character(cut(gss$age,
                    breaks= c(age_groups, Inf),
                    labels = age_groups,
                    right = FALSE)))
```

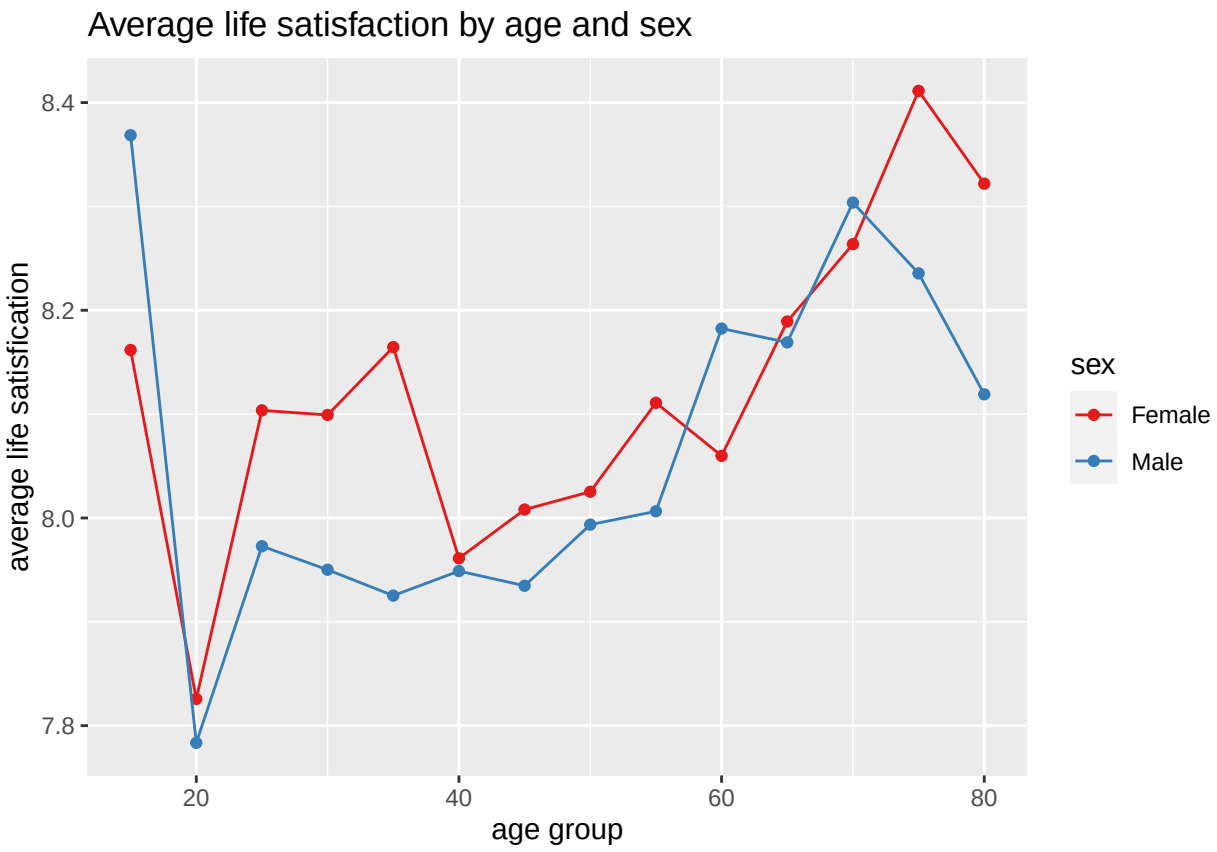
Now let's calculate the average of the 'life satisfaction' variable by age group and sex. This involves a `group_by` by two variables:

```
life_satis_age_sex <- gss %>%
  group_by(age_group, sex) %>%
  summarise(mean_life_satis = mean(feelings_life, na.rm = TRUE))
```

Plot as a line chart over age, coloring by sex, for this example we use a different colour palette called "Set1":

```
ggplot(data = life_satis_age_sex, aes(x = age_group, y = mean_life_satis, colour = sex)) +
  geom_point() +
  geom_line() +
  scale_color_brewer(palette = "Set1") +
```

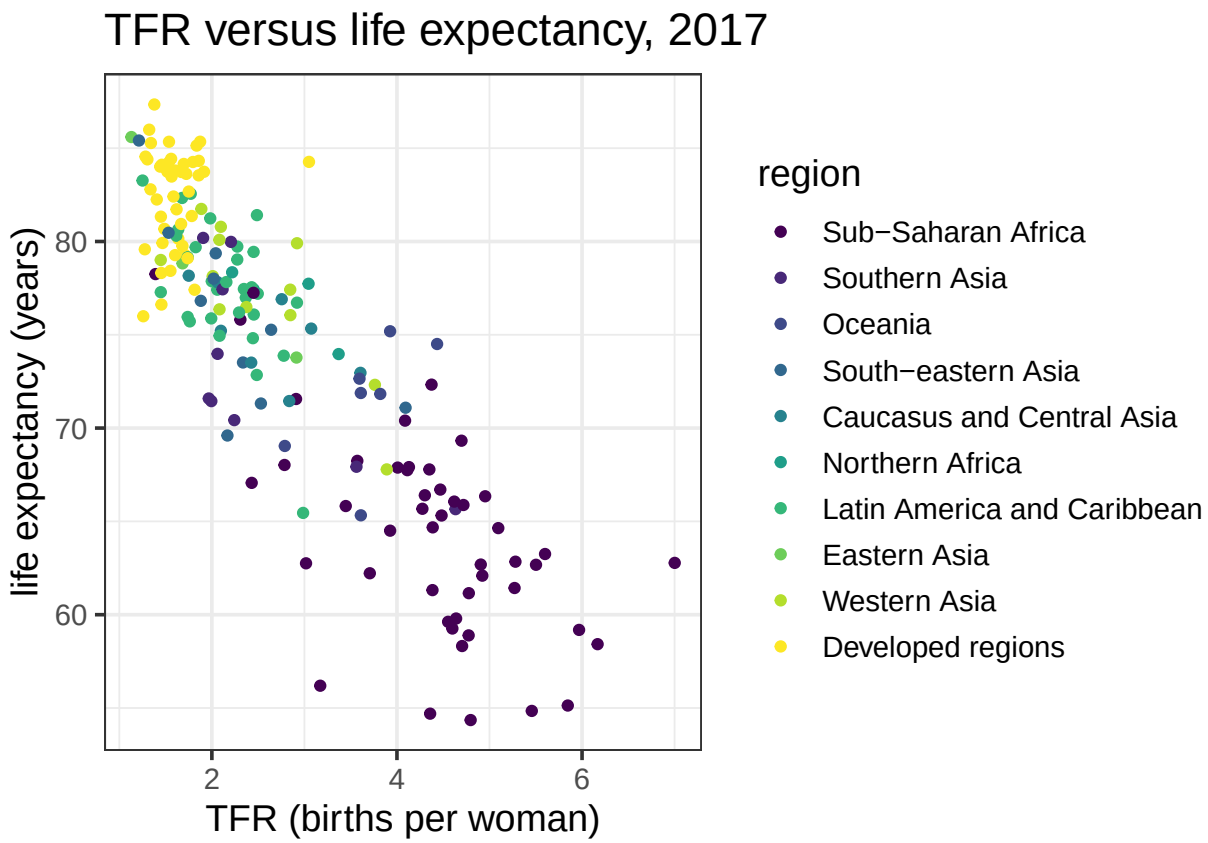
```
ylab("average life satisfication") +
xlab("age group") +
ggtitle("Average life satisfaction by age and sex")
```



4.5 Scatter plots

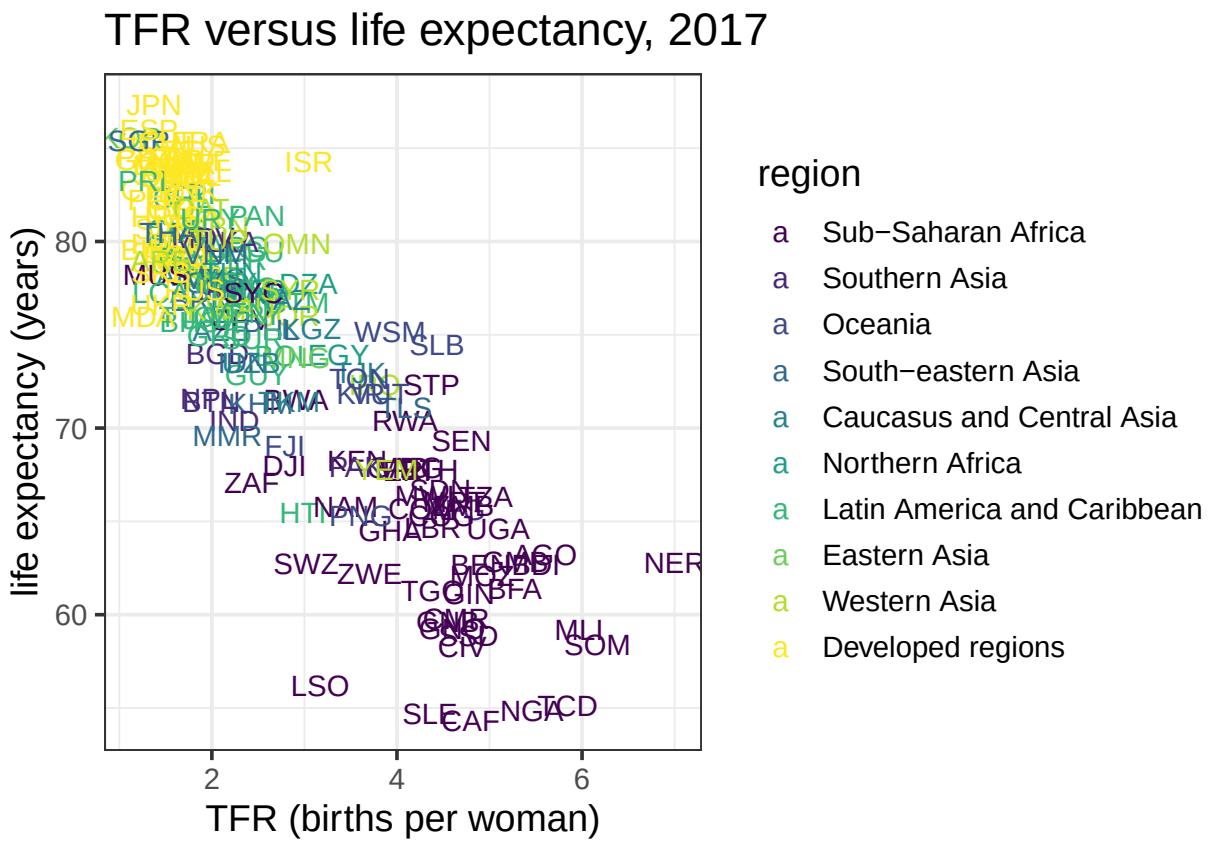
Start with the scatter plot from the introductory example:

```
ggplot(country_ind_2017, aes(tfr, life_expectancy, color = region,)) +
geom_point() +
ggtitle("TFR versus life expectancy, 2017")+
theme_bw(base_size = 14) +
ylab("life expectancy (years)") +
xlab("TFR (births per woman)") +
scale_color_viridis_d()
```



Instead of dots could have country codes (although becomes hard to read, but easy to see outliers)

```
ggplot(country_ind_2017, aes(tfr, life_expectancy, color = region, label = country_code)) + # adding
geom_text() +
ggtitle("TFR versus life expectancy, 2017")+
theme_bw(base_size = 14)+
ylab("life expectancy (years)") +
xlab("TFR (births per woman)") +
scale_color_viridis_d()
```



4.6 Faceting

Changing the color and fills is useful to show one other variable on a graph. For more complicated set-ups, faceting graphs by an additional variable becomes useful.

For example let's go back to plotting a histogram of age at first marriage by sex, but also add in whether or not the respondent was born in Canada. First, look at the unique values of the `place_birth_canada` variable:

```
gss %>%
  select(place_birth_canada) %>%
  unique()
```

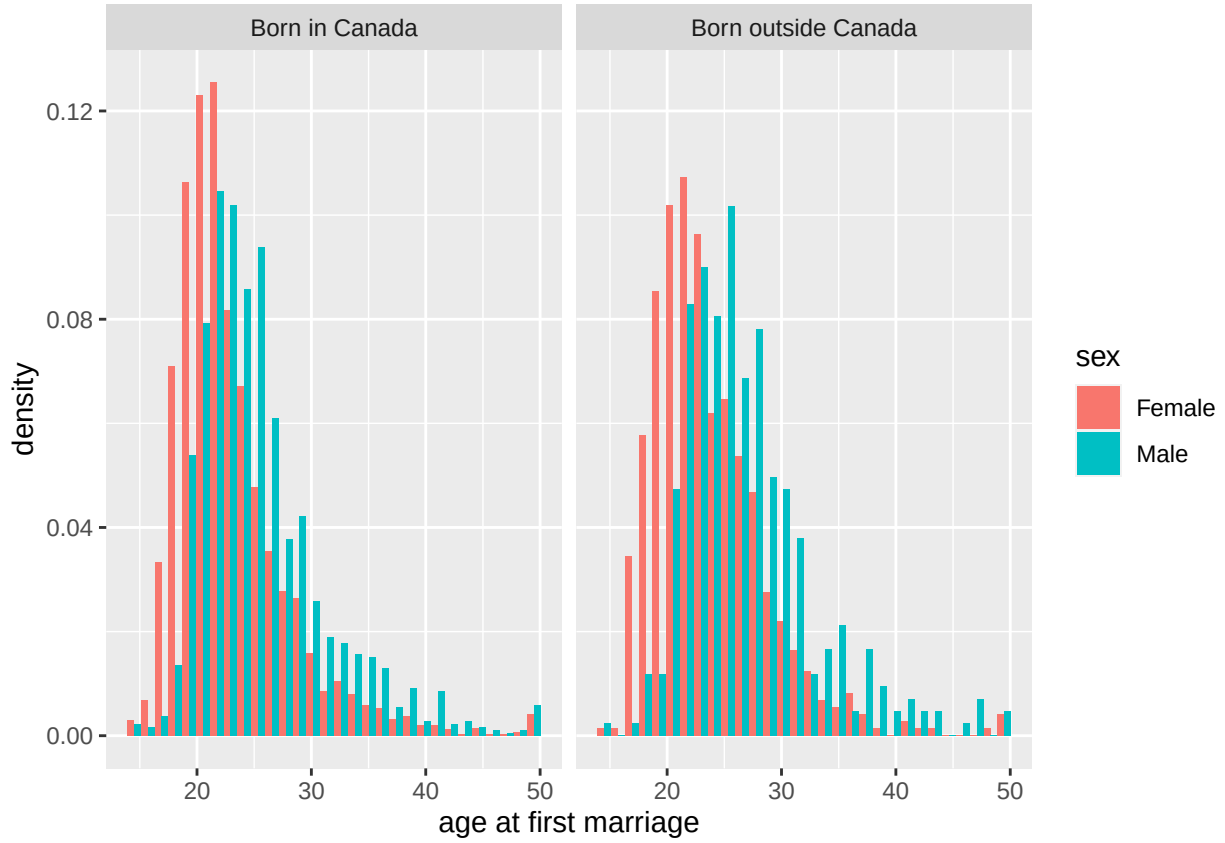
```
## # A tibble: 4 x 1
##   place_birth_canada
##   <chr>
## 1 Born in Canada
## 2 Born outside Canada
## 3 <NA>
## 4 Don't know
```

For now, filter the data to only include the first two categories. To do this, use the `%in%` function within filter:

```
gss_subset <- gss %>%
  filter(place_birth_canada %in% c("Born in Canada", "Born outside Canada"))
```

Now plot the histograms as before, but now also facet by place of birth. Note we are plotting the density here.

```
ggplot(data = gss_subset, aes(age_at_first_marriage, fill = sex)) +
  geom_histogram(position = 'dodge', aes(y = ..density..)) +
  facet_wrap(~place_birth_canada) +
  xlab("age at first marriage")
```



5 Review Questions

1. Using the country_indicator dataset, create a scatter plot of GDP over life expectancy by region for the year 2014. Edit the labels, set a title, and make sure the graph is colour coded.

2. Using the GSS dataset, create a bar graph of non-missing values for the province of birth (`place_birth_province`) and then arrange the proportions from high to low. Make sure to colour code and make all labels are readable.