Introduction to **R**

Session 3 – Data visualisation

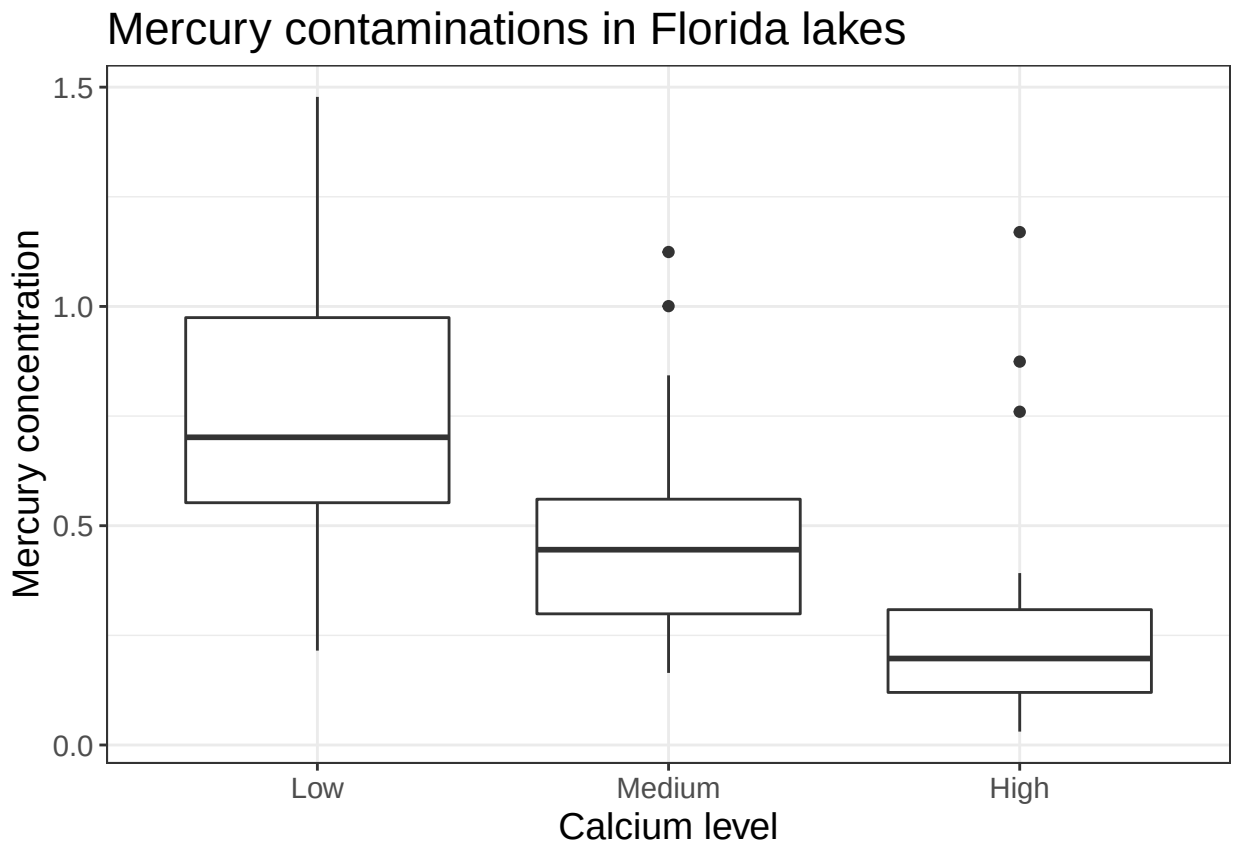
1. Boxplot

With the Florida lakes data set, recreate the figure below of boxplots showing the distribution of Mercury for each of the levels of Calcium.

Remember to always create informative axis labels and ensure your text is large enough to easily read.

Note, If there is no colour in a plot, the black and white theme (`theme_bw`) is generally recommended.

```
ggplot(lake.df, aes(x = Calcium, y = Mercury)) +
  geom_boxplot() +
  labs(title ="Mercury contaminations in Florida lakes",
       x = "Calcium level", y = "Mercury concentration") +
  theme_bw() +
  theme(text = element_text(size = 14))
```

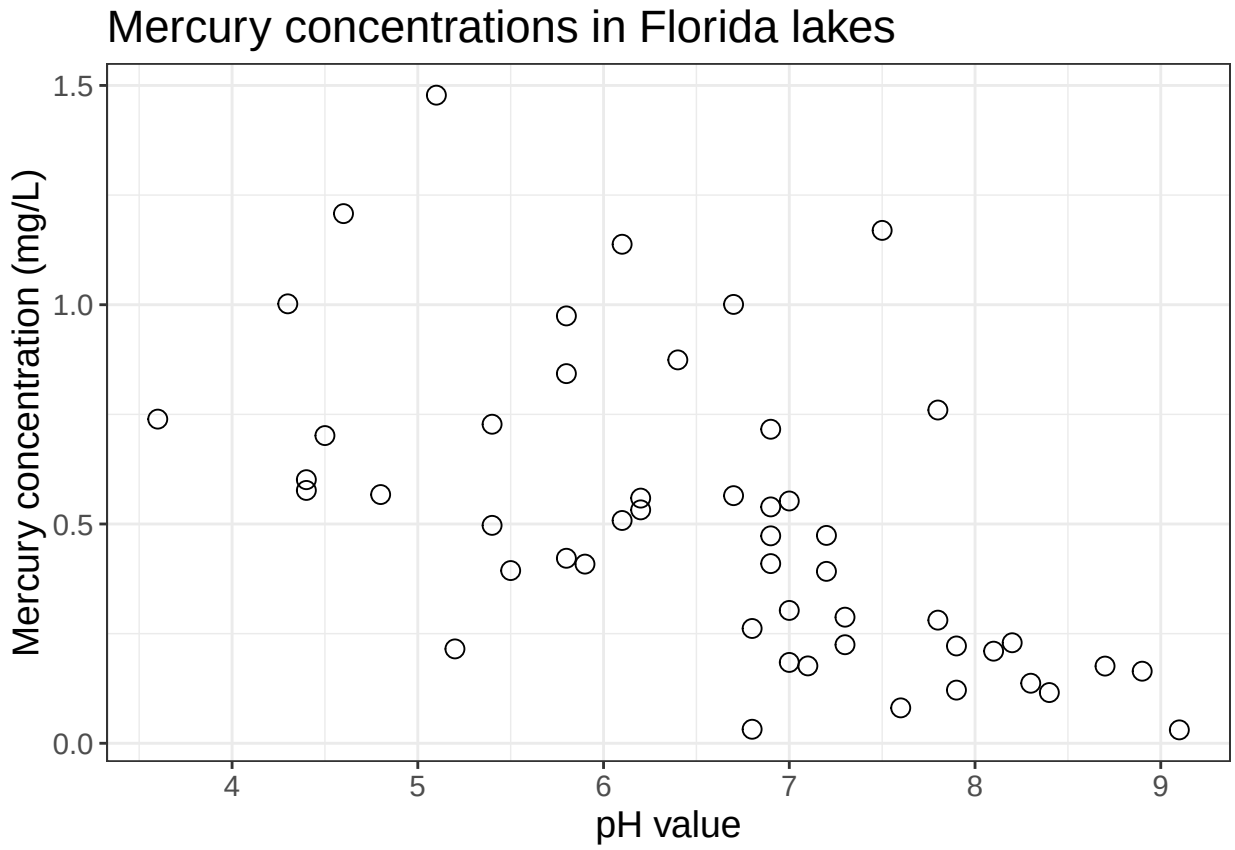


2. Scatterplot

1. With the Florida lakes data set, recreate the scatter plot below showing the relationship between `pH` and `Mercury`.

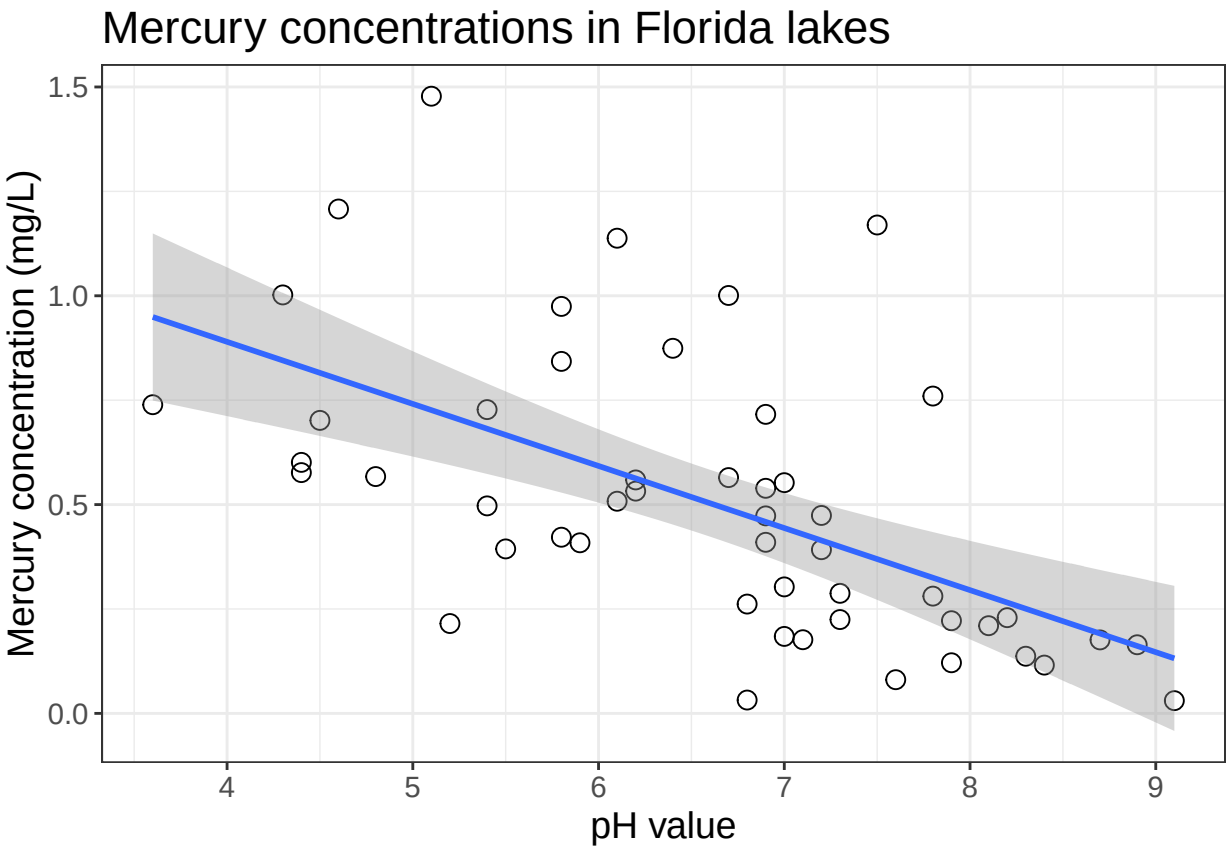
Hint: `shape = 1` corresponds to a hollow circle.

```
ggplot(lake.df, aes(x = pH, y = Mercury)) +
  geom_point(shape = 1, size = 3) +
  labs(title ="Mercury concentrations in Florida lakes",
       x = "pH value", y = "Mercury concentration (mg/L)") +
  theme_bw() +
  theme(text = element_text(size = 14))
```



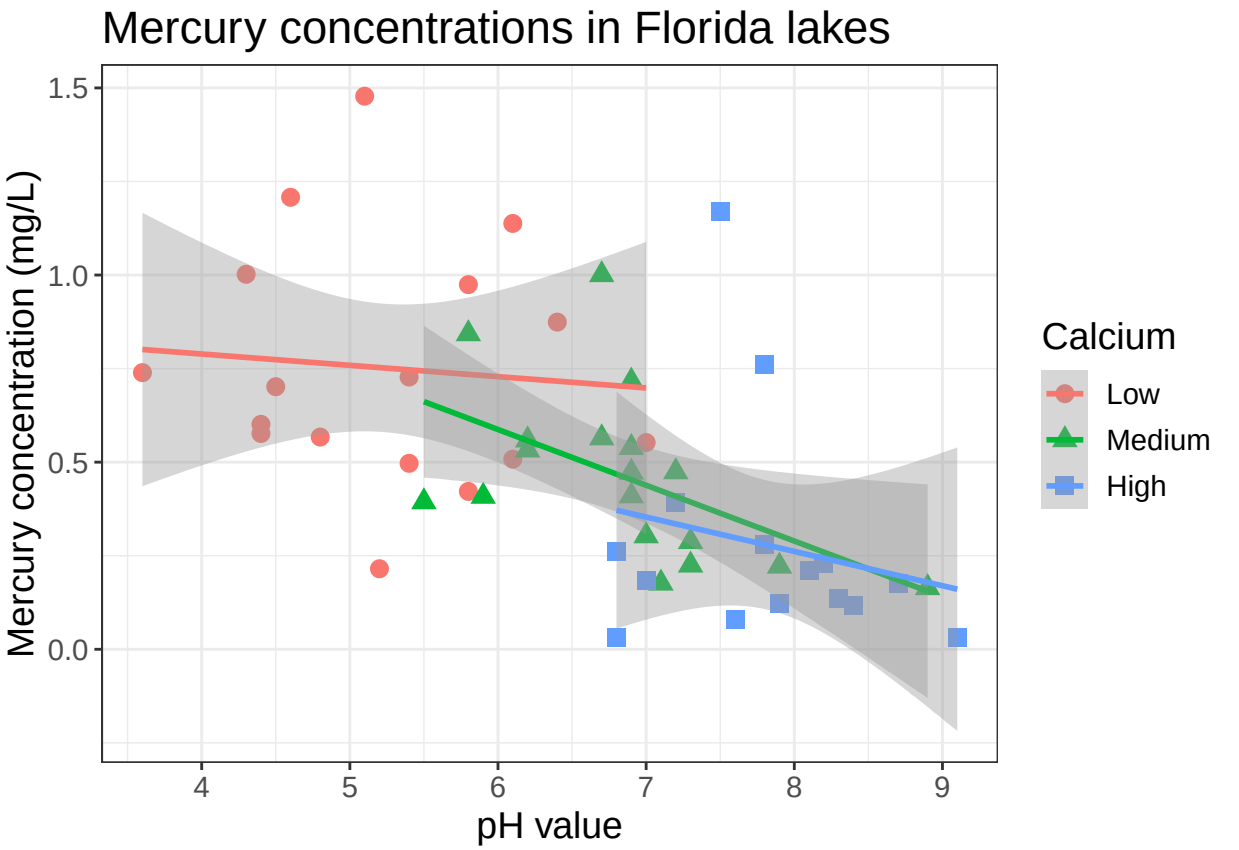
2. Add a linear smooth to the plot with a standard error region.

```
ggplot(lake.df, aes(x = pH, y = Mercury)) +
  geom_point(shape = 1, size = 3) +
  labs(title ="Mercury concentrations in Florida lakes",
       x = "pH value", y = "Mercury concentration (mg/L)") +
  geom_smooth(method = "lm") +
  theme_bw() +
  theme(text = element_text(size = 14))
```



3. Include a legend showing different shapes and colours for the points corresponding to the different levels of calcium. Also include different coloured smooths for the different levels of calcium.

```
ggplot(lake.df, aes(x = pH, y = Mercury, colour = Calcium)) +
  geom_point(aes(shape = Calcium), size = 3) +
  labs(title ="Mercury concentrations in Florida lakes",
       x = "pH value", y = "Mercury concentration (mg/L)") +
  geom_smooth(method = "lm") +
  theme_bw() +
  theme(text = element_text(size = 14))
```



3. Barplot

1. Using the `ifelse` function, create a variable named `age_group` in the `patient.df` data set. This variable will have 3 levels:
   - "Under 35"
   - "36 to 60"
   - "Over 60"

```
patient.df$age_group = with(patient.df,
                            ifelse(Age < 35, "Under 35",
                                   ifelse(Age > 60, "Over 60", "36 to 60")))
```

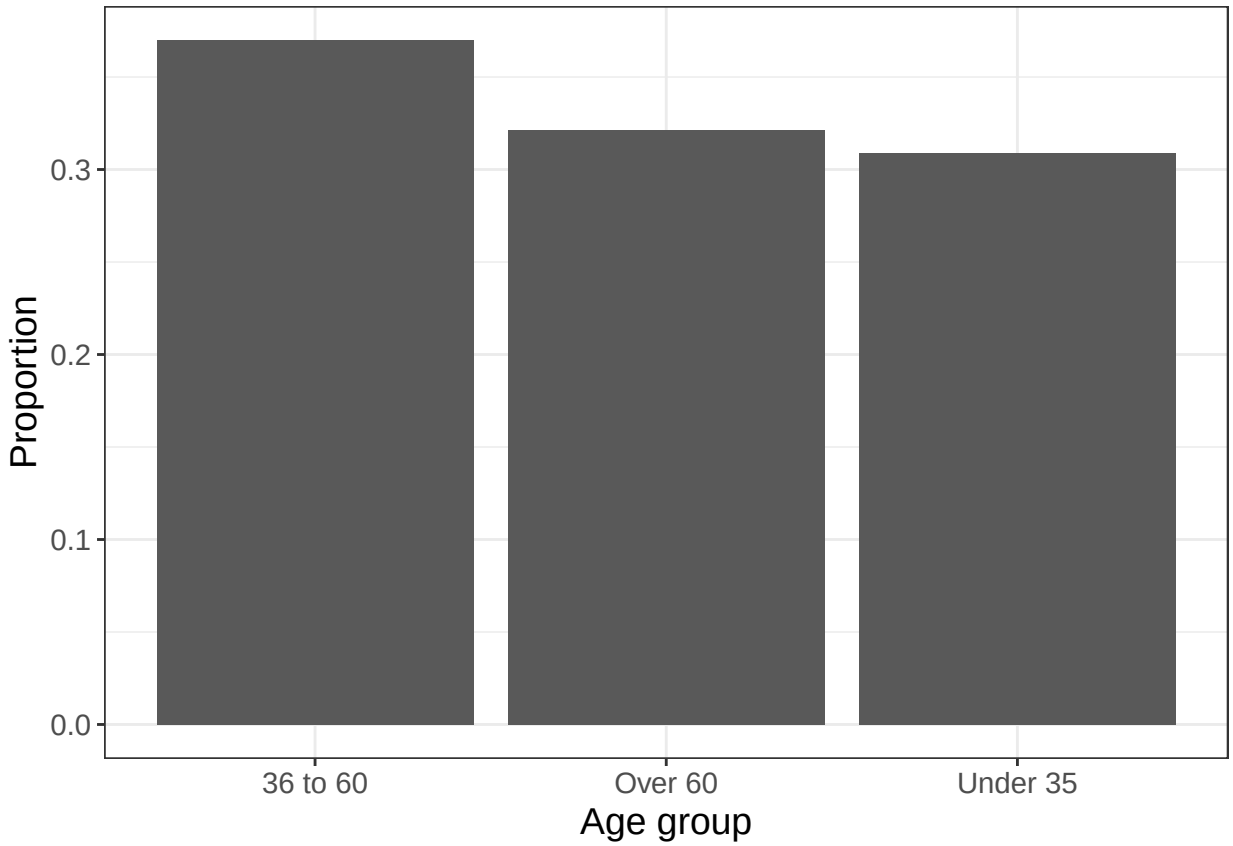
2. Run the following code: `levels(factor(patient.df$age_group))`.

This shows that the default levels are ordered as "36 to 60", "Over 60", and "Under 35", i.e. they aren't ordered correctly. If left unchanged, this default order `ggplot2` will be used when plotting this variable. Convert `age_group` to a factor variable with a sensible order.

3. Draw a barplot showing the proportion of patients in each age group.

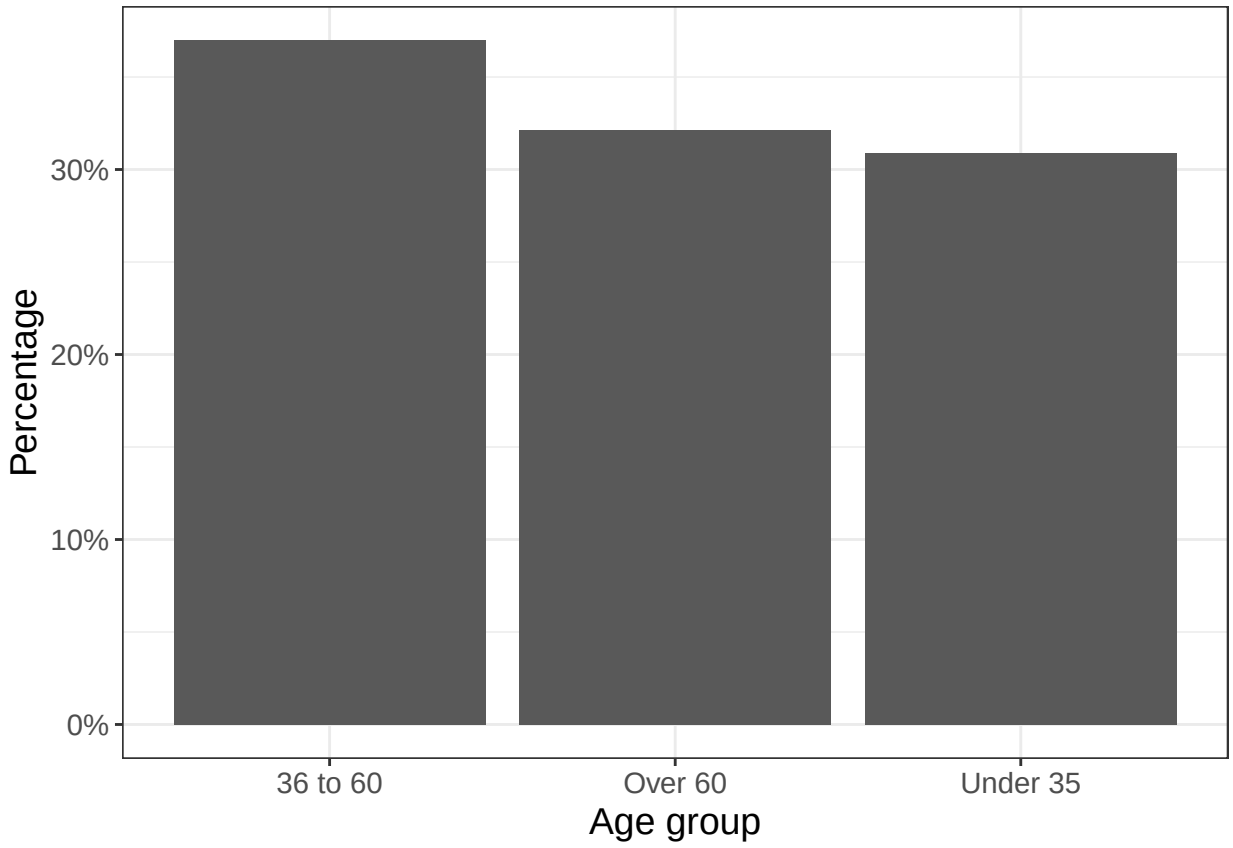
Hint: you will need to include `group = 1` in your code.

```
ggplot(patient.df, aes(x = age_group, y = ..prop.., group = 1)) +
  geom_bar() +
  labs(x = "Age group", y = "Proportion") +
  theme_bw() +
  theme(text = element_text(size = 14))
```



4. Install and load the `scales` package. The `scales` package will allow us to easily change the proportions on the y-axis to percentages. Add `scale_y_continuous(labels = percent)` to your code to see the effect.

```
ggplot(patient.df, aes(x = age_group, y = ..prop.., group = 1)) +
  geom_bar() +
  labs(x = "Age group", y = "Percentage") +
  scale_y_continuous(labels = percent) +
  theme_bw() +
  theme(text = element_text(size = 14))
```


30%
20%
10%
0%
Percentage
36 to 60
Over 60
Under 35
Age group