Computer Practical 2: Plotting with ggplot2 (continued) and testing for normality

Dr Alice Risely [a.risely@salford.ac.uk](mailto:a.risely@salford.ac.uk)

Table of Contents

## Learning outcomes for today

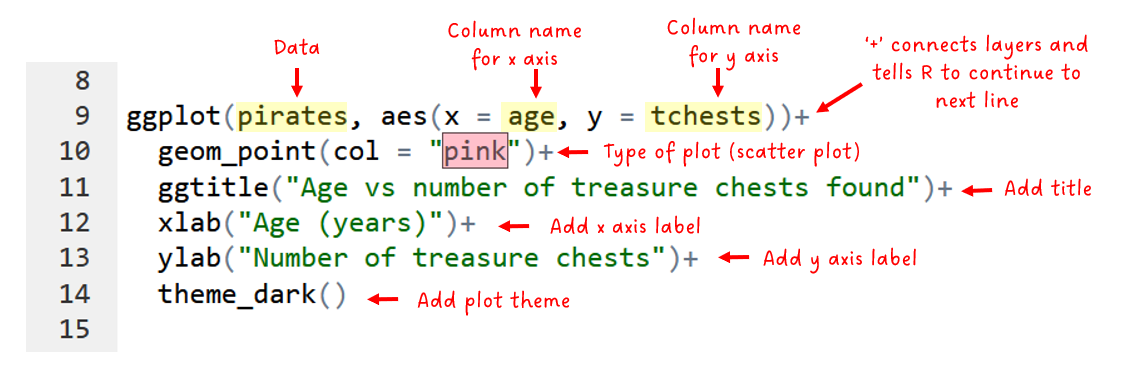
What you will learn in this practical:

* Reminder of how to summarise and plot data
* Learn how to colour plots by a categorical variable
* Learn how to add trend lines and jittered points to plots
* Test numerical data for normality using the Shapiro-Wilk test
* You have a homework task at the end of the worksheet.

## Practical 1 learning outcomes

In practical 1 you learned:

* How to use R as a calculator; e.g., **sqrt(9)**
* How to save *objects* to the working environment,;e.g., **a <- 100**
* How to install and load packages; **install.packages(“ggplot2”)** ; **library(ggplot2)**
* How to view data as a table in a separate tab; **View(pirates)**
* How to summarise numerical and categorical data to get the mean, minimum, and maximum; **summary(pirates$age)** and **table(pirates$sex)**
* How to change the data types to categorical data: **as.factor(pirates$id) ;** count data: **as.integer(pirates$parrots)**; and continuous data: **as.numeric(pirates$weight)**.
* How to make histograms, bar plots, scatter plots, and boxplots in ggplot2 and add axis labels and titles.



## Preliminary tasks

1. I**f you have not completed practical 1, you need to do this first.** Please go back to Blackboard and click on the practical 1 word document and follow instructions.

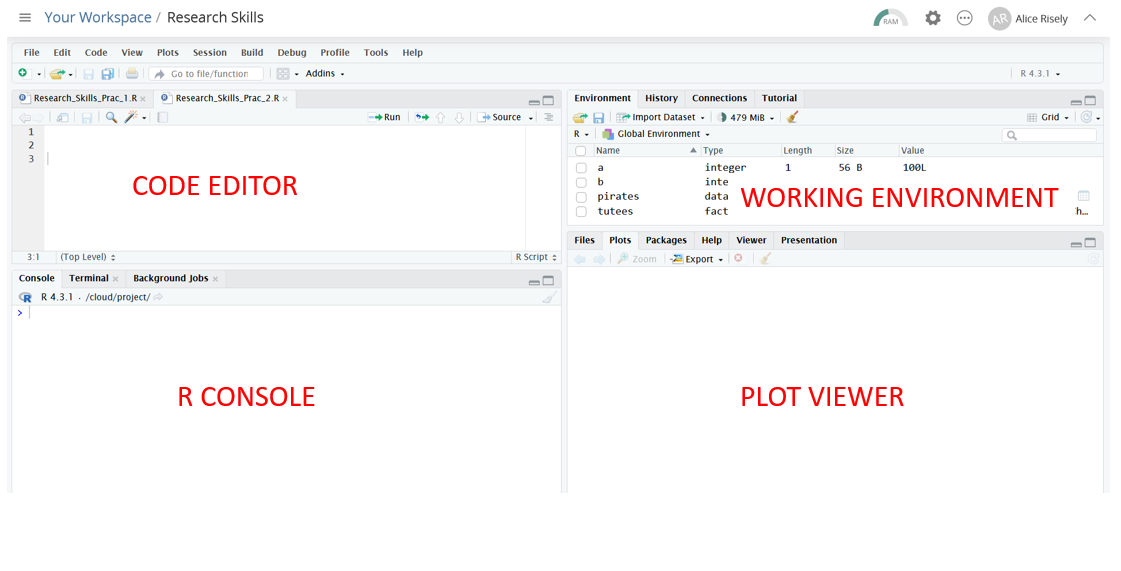
2. If you are ready to start Practical 2, **DOWNLOAD the second practical worksheet** from Blackboard, do not just view it. To download it, click the “…” next to the file and click ‘Download original file’. **Save this file in an appropriately named folder on your OneDrive (e.g., “MODULE NAME/WEEK X”) so you can access it at home.**

3. **Navigate on your web browser to** [**https://login.rstudio.cloud/**](https://login.rstudio.cloud/)**.** Assuming you have already registered for practical 1, then sign in. If prompted, click “Posit Cloud”.

4. **You should already have an existing R studio project from the first practical called ‘R workshop’,** click on this to enter your saved workspace from practical 1.

5. **Start a new R script for this practical.** Once your project has loaded (it takes a few seconds), click ‘File’ then ‘New file’, then ‘R Script’.

7. **Press the Save button and save your script as “Practical\_2.R”**. This will save all your code you write today. You should now have a screen that looks like this:



## Recap: How to run code in R

* To run a line of code in R, put your mouse cursor on the line you want to run and press \*\*Control and Enter\*\* (this is the most common method). Alternatively, press the ‘Run’ button above the code editor.
* **Anything after a # key is NOT code**, it is comments/notes. R will ignore any writing after the # key. You do not need to copy out the comments into R.
* When you run the code, R will process your command and send it to the R console, where you will see both the code and the output.

## Recap: Loading packages

The first thing you do when opening R is to load the packages you need. In this class, we will be using the packages ggplot2 for plotting, and yarrr for accessing the pirate data.

library(ggplot2) # load the package ggplot2  
library(yarrr) # load the package yarrr

## Recap: Summarising data

First we will practice summarising and plotting data using the dataset *pirates*, which we explored in practical 1.

First, lets remind ourselves of this dataset by exploring it again.

# data(pirates) # load the dataset 'pirates' into your environment if it is not tehre already. You should still have it from last week  
  
head(pirates) # check the first six rows of the dataset  
  
names(pirates) # check names of the columns   
  
str(pirates) # look at the structure of the dataset and the data types of each column  
  
View(pirates) # View the dataset in a different tab

Reminder:

* To summarise numerical data, you need the function **summary()**
* To summarise categorical data, you need the funtion **table()**

#### Exercise 1

What is the mean, minimum, and maximum of the number of treasure chests that pirates have found/stolen? Remember, number of treasure chests is count data, therefore it is numerical.

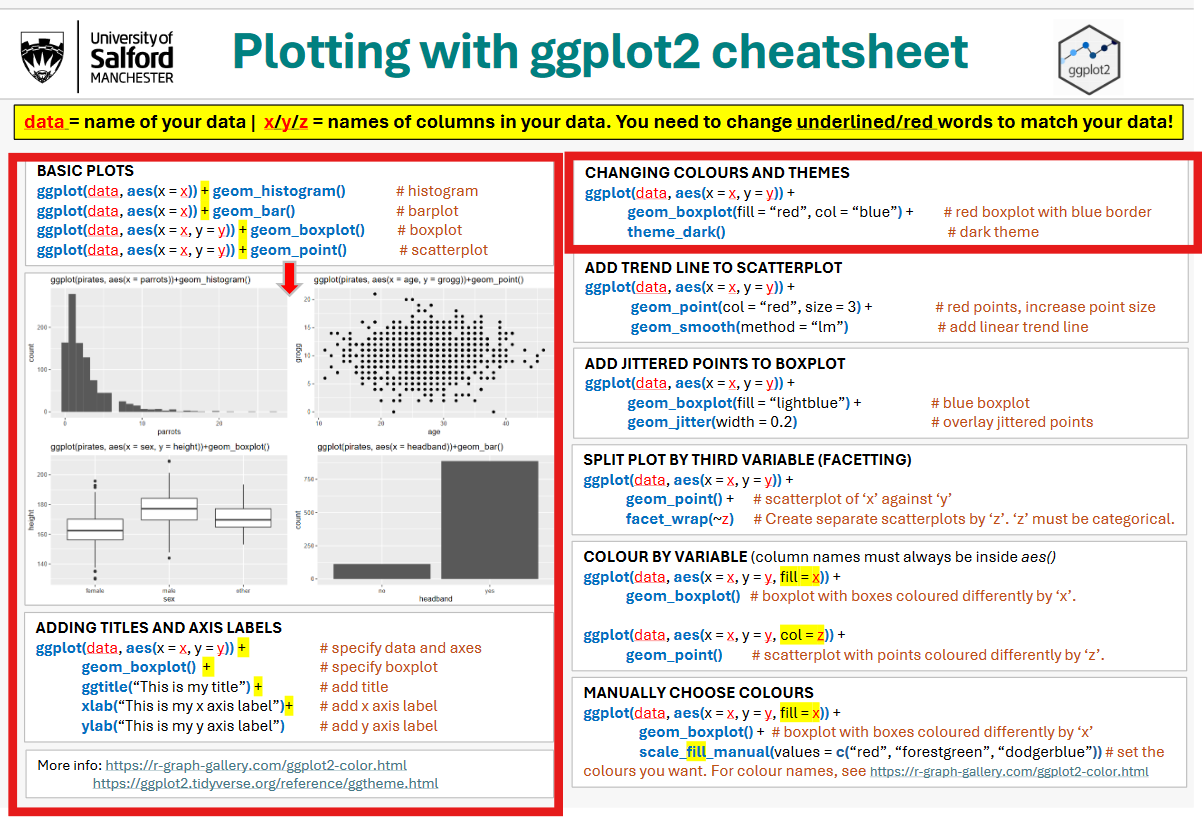
\*\*\* COPY YOUR ANSWERS HERE \*\*\*  
  
  
  
---

What is the most popular pixar movie of pirates? Remember, favourite pixar movie is a categorical variable.

\*\*\* COPY YOUR ANSWERS HERE \*\*\*  
  
  
  
---

## Recap: Basic plotting

We will continue with plotting in ggplot2 using the *pirates* dataset. In practical 1 you learned how to make basic plots, and also how to add axis labels, titles, and change the colours of your plots using the code highlighted in red below.



Practice basic plotting by completing the exercises below. Remember to use the cheat sheet (above or the paper handout) to help you.

#### Exercise 2

Make a histogram of pirate height. Add the title “Histogram of pirate height” and change the x axis label to “Height (cm)”.

\*\* COPY PLOT HERE \*\*  
  
  
  
  
---

#### **Exercise 3**

Make a bar plot showing the popularity of favourite pixar movies amongst pirates. Add the title “Favourite pixar movies of pirates” and change the x axis label to “Pixar movies”.

**HINT: Because the column names are so long, try putting favourite pixar movie on the y axis instead.** Normally the categorical variable should go on the x axis, but when variable names are long it is neater to put the categorical variable on the y axis.

\*\* COPY PLOT HERE \*\*  
  
  
  
---

#### **Exercise 4**

Make a scatter plot showing the relationship between weight and height in pirates. Add a title and sensible axis labels (with units).

\*\* COPY PLOT HERE \*\*  
  
  
  
---

#### **Exercise 5**

Make a boxplot to see whether the age of pirates differs by what college they went to. Add a title and sensible axis labels (with units).

\*\* COPY PLOT HERE  
  
  
  
---

## Adding a trend line to a scatterplot

Sometimes when you have made a scatter plot it is useful to visualise a trendline. To do this, you add the layer: **geom\_smooth():**

ggplot(pirates, aes(x = height, y = weight))+  
 geom\_point()+ # scatter plot  
 geom\_smooth(method = "lm") # trend line

**NOTE: The *method = “lm”* argument ensures the line is straight - not wobbly.** In most cases, you want a linear trend line, not a wobbly one unless you suspect a non-linear relationship.

## Overlaying jittered points to a boxplot

Sometimes when you have made a boxplot, it is useful to overlay the raw datapoints over the boxplot so you can see the variation in the data. To do this, you can use the layer **geom\_jitter().** Below is a boxplot of age distribution of pirates that went to different schools, overlayed by the raw data points:

ggplot(pirates, aes(x = college, y = age))+  
 geom\_boxplot()+ # boxplot  
 geom\_jitter() # jittered points  
  
#make points a bit narrower  
ggplot(pirates, aes(x = college, y = age))+  
 geom\_boxplot(fill = "red")+  
 geom\_jitter(width = 0.2, col = "pink", alpha = 0.3) # make points a bit narrower, colour them pink, make a bit transparent (with alpha)

## Colouring plots by a categorical variable

So far, we have used the arguments *col = “yellow”* or *fill = “pink”* to change the colour or fill of our plots elements. **However, sometimes it is useful to colour your data by a categorical variable, rather than just plot a single colour.**

For example, lets make a boxplot that compares beard length across pirate sex. Compare the colours of these two plots:

# colour the boxes light blue and the points red  
  
ggplot(pirates, aes(x = sex, y = beard.length))+  
 geom\_boxplot(fill = "lightblue")  
  
  
# colour the boxes by sex (note = you dont have to specify any colours! It does that automatically)  
  
ggplot(pirates, aes(x = sex, y = beard.length,   
   
 fill = sex))+ # <-- NOTICE THIS BIT  
 geom\_boxplot()

In the first plot, the fill of all the boxes is light blue. However, in the second plot, you have told ggplot to colour the boxes by pirate sex by adding ‘fill = sex’ in your mapping (aes) function in the first line.

ggplot(pirates, aes(x = sex, y = beard.length, ***fill = sex***)) + geom\_boxplot()

**Notice it will automatically provided a *legend* on the right.**

**Notice that because you are referring to the column ‘sex’ in your pirates dataset, the instruction *fill = sex* MUST always be inside the *aes()* brackets. This is because ggplot will only know you are referring to a column name if it is inside the *aes()* brackets.**

Now lets practice with a scatter plot:

ggplot(pirates, aes(x = age, y = tchests,   
 col = sex))+ # <-- NOTICE THIS BIT  
 geom\_point()

Note: For colouring your plots, use the argument ***col*** for scatter plots and ***fill*** for all other plot types:

**col = sex** : scatter plots

**fill = sex** : histograms, bar plots, and boxplots

#### Exercise 6

Make a barplot showing how many pirates wear headbands compared to those who don’t, and colour the plot by headband. Note, for barplot, you use the argument *fill*, not *col*.

\*\* COPY PLOT HERE \*\*  
  
  
  
---

#### Exercise 7

Make a scatterplot of age against height, and colour the points by sex. Add a trendline and colour this by sex too (i.e, there should be three trend lines in your plot, one per sex).

\*\* COPY PLOT HERE \*\*  
  
  
  
---

## Manually assigning colours

Unfortunately, the automatic colours chosen by ggplot2 when colouring by a categorical variable are rather ugly in my opinion. For those who like creating pretty plots, you can manually tell ggplot what colours you want it to use when colouring by a categorical variable, by using the layer ***scale\_fill\_manual()****,* or ***scale\_colour\_manual()***, depending on if you have used *col* or *fill* to specify colour. However, this is not essential.

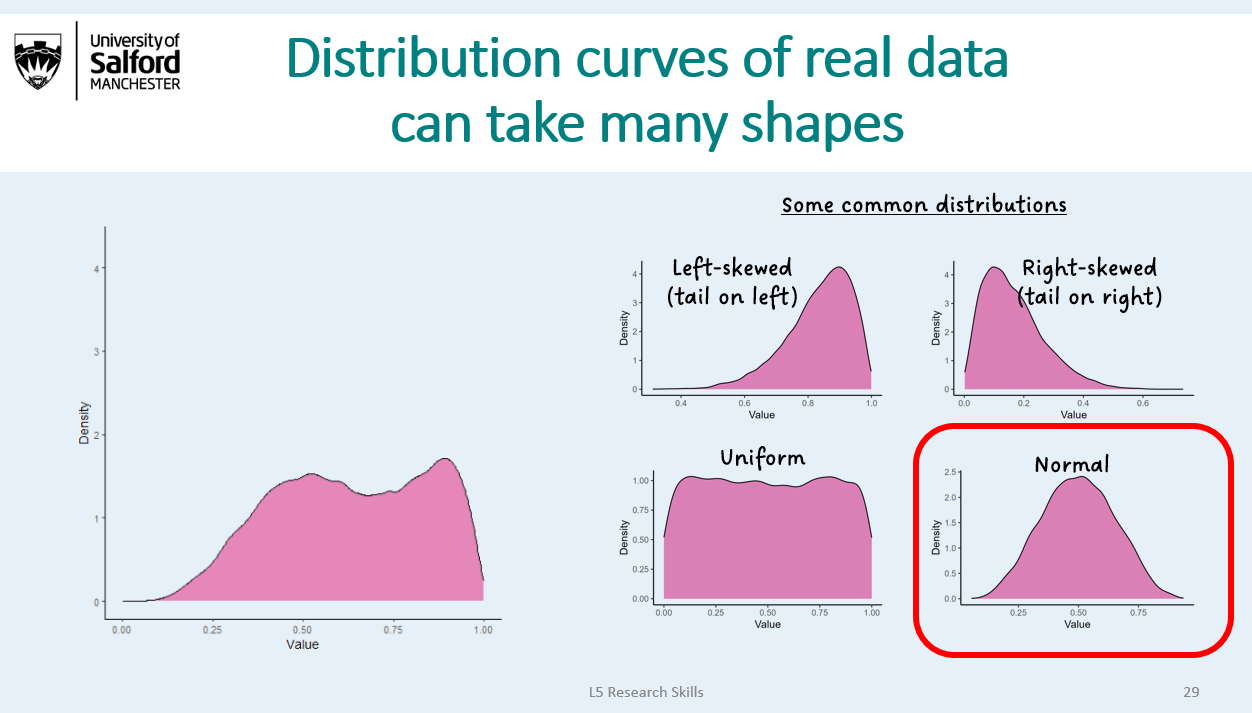
# Boxolot with colours chosen manually:  
  
ggplot(pirates, aes( x = sex, y = beard.length, fill = sex))+  
 geom\_boxplot()+   
 scale\_fill\_manual(values = c("darkblue", "darkcyan", "cadetblue2"))  
  
  
# Scatterplot with colours chosen manually:  
  
ggplot(pirates, aes( x = height, y = weight, col = sex))+  
 geom\_point()+   
 scale\_colour\_manual(values = c("aquamarine3", "cornflowerblue", "brown4"))

Here are the names of some colours in R, and you can find more here: <https://bookdown.org/hneth/ds4psy/D-3-apx-colors-basics.html>



## Distribution curves

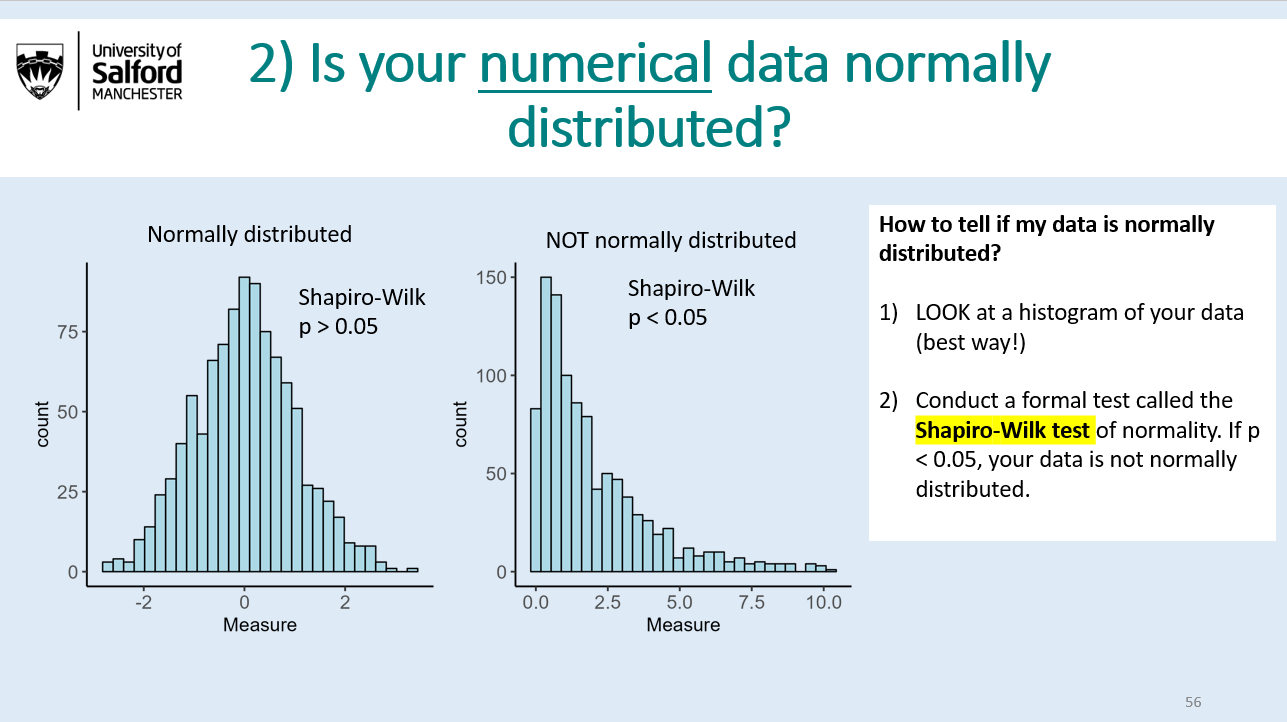
Think back to the part in last weeks lecture about common distribution curves (which can be gauged from histograms):



The most important distribution curve is the NORMAL DISTRIBUTION (or bell-shaped curve). To be able to choose a statistical test, you need to know if your numerical data is normal or not.

**To do this, you need to 1) plot a histogram of your numerical data, and 2) perform a Shapiro-Wilk test of normality**.

If the p value from your test is above 0.05, you can be confident that your data IS normally distributed. Remember this slide from the lecture:



Lets now test some the numerical variables in the *pirates* dataset for normality. First, lets make two histograms:

# histogram of pirate weight  
  
ggplot(pirates, aes(x = weight))+  
 geom\_histogram(col = "white")  
  
# histogran of number of parrots owned by pirates  
  
ggplot(pirates, aes(x = parrots))+  
 geom\_histogram(col = "white")

#### Exercise 8

Take a look at the histograms of weight and number of parrots you just made. What types of distributions do you think each of these variables have?

\*\* WRITE ANSWER HERE \*\*  
  
  
  
---

## The Shapiro-Wilk Test of Normality

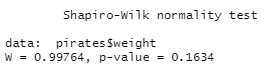
Whilst it is possible to look at the histograms and take a guess at their distributions, there is an official way to test whether a distribution is ‘normal’ or not. This is important, because whether data is normally distributed or not will guide which statistical models you can apply to your data later.

The statistical test that tests data for normality is called the **Shapiro-Wilk normality test,** and it will tell you whether the data is significantly different from normal.

**Lets jump in and test whether pirate weight is normally distributed:**

shapiro.test(pirates$weight)

You will get an output like this:



Your effect size is essentially 1 (= no effect), and your p-value is 0.16. This means that your null hypothesis (that your data is normal) is likely to be true.

**Use this table to help you interpret the results of the test:**

|  |  |
| --- | --- |
| p-val | Conclusion |
| < 0.05 | Data is NOT normally distributed |
| > 0.05 | Data IS normally distributed |

#### **Exercise 9**

Use a Shapiro Wilk test of normality to test whether the number of parrots pirates have is normally distributed. Is it?

**NOTE: 2.2e-6 is scientific notation for very small numbers. It means you need to move your decimal place 6 places to the right. For example, 2.2e-6 = 0.0000022. Any p value with an “e” is definitely less than 0.05!**

\*\* COPY YOUR ANSWERS BELOW \*\*  
  
  
  
  
---

#### Exercise 10

Test whether pirate height is normally distributed by plotting a histogram and conducting a Shapiro-Wilk test. It is normally distributed?

\*\* COPY YOUR PLOT AND ANSWER BELOW \*\*  
  
  
  
  
---

#### Exercise 11

Test whether the time it takes for a pirate to draw their sword (‘sword.time’) is normally distributed by plotting a histogram and conducting a Shapiro-Wilk test. Is it normally distributed?

\*\* COPY YOUR PLOT AND ANSWER BELOW \*\*  
  
  
  
  
---

## Homework

As with last weeks homework, lets use the penguin dataset from the package palmerpenguins to practice some of the things you learned in this practical on a real biological dataset.

#install.packages("palmerpenguins") # install palmerpenguins package if you haven't already   
  
library(palmerpenguins) # load palmerpenguins package  
  
# note- if you get an error it means you haven't installed it. Go back to the homework section of Prac 1 to do this.   
  
data(penguins) # load the penguin dataset  
  
?penguins # check out information for the dataset  
  
View(penguins) # check out the dataset, making a note of the column names

#### Exercise 12

Plot the relationship between flipper length and bill length in penguins, and colour the points by species. Add a trendline, with the trendlines also coloured by species. Make sure the x and y axes are labelled neatly, with units. Colour the different species with the colours “brown3”, “blue4”, and “darkcyan”.

\*\* COPY YOUR PLOT AND ANSWER BELOW \*\*  
  
  
  
  
---

#### Exercise 13

Assess if penguin bill length is normally distributed by plotting a histogram and doing a Shapiro-Wilk test.

\*\* COPY YOUR PLOT AND ANSWER BELOW \*\*  
  
  
  
  
---

Remember, in scientific notation, 5e-5 means you move the decimal point five places to the left. So, 5e-5 = 0.00005. Why do you think bill length has an odd distribution? To find out, colour the histogram by species (using *fill = species*).