TUTORIAL 3

Aim:

Practice data visualisation using the `ggplot2` package, and describing data numerically in R.

Mode of study:

You can work through this tutorial in your own time, or in one of the assigned venues during the tutorial slot. Help is available during the tutorial slot in the venues.

Solutions will be made available in Week 4.

Before you start

If you have not worked through Practical 3 yet, you should do so before starting this tutorial. You should also load the `tidyverse`.

INTRODUCTION

In Tutorial 2 you used the `diamonds` and `iris` datasets. We will now use these two datasets again, so it might be useful to **look at the help documentation for them** again, to refresh your memory regarding what the various variables represent.

You will not receive step-by-step instructions to create the plots as in Practical 3. You should try to construct the plots yourself, using the knowledge you acquired in Practical 3. Also remember that there may be more than one way to construct the same plot, and that you can customise your plot as you see fit. Solutions to the tutorial will be provided in due course, but keep in mind that your code and resultant plots might not look exactly the same as those in the solution. The solution is meant to provide a model answer, but variations are possible.

Tip:

RStudio contains some cheat sheets that can come in very handy, especially when working in the `tidyverse`. There is a cheatsheet for `ggplot` specifically, summarising all the important aspects of plotting in `ggplot2`. You can access the cheat sheet in RStudio by clicking on Help -> Cheat Sheets -> Data Visualization with ggplot2. This should open the cheat sheet in your browser. This is a good reference to keep handy; you should look at the one for `dplyr` as well.

IRIS DATA

1. Calculate the following:

1.1 The average petal length and petal width of all irises in the dataset.
[Hint: the function `mean()` can be used.]

1.2 The standard deviation of petal length and petal width.
[Hint: the function `sd()` can be used.]

1.3 The correlation between petal length and petal width.
[Hint: the function `cor()` is used in R to calculate the correlation between vectors.]

2. Create a scatterplot of the petal length (*x*-axis) and petal width (*y*-axis) variables, using different coloured points for the different iris species.

3. If you study the plot created in (2), you will see that there is some overplotting taking place. Fix this by **adding a small amount of random noise to each point.**
[Hint: you first encountered overplotting in Practical 3.]

4. Add a horizontal line to the plot created in (3), showing the average petal width of irises in the dataset as well as a vertical line showing the average petal length of the irises. [Hint: use `geom_hline()` and `geom_vline()`. You might have to look at the help files for these functions to see exactly how you can accomplish this.]

DIAMONDS DATA

5. Histograms can be plotted using `geom_histogram()`. The syntax of everything else stays the same.

5.1 **Explore the distribution of the numerical variable `price` by constructing a histogram.**

5.2 In Lecture 6 the importance of the number of choosing a suitable number of bins when constructing a histogram was discussed. In `geom_histogram()` a default number of bins will be used unless you explicitly specify the number of bins to use. You can do this using the `bins` argument within the `geom_histogram()` function call.
Change the number of bins to 12 in the histogram you plotted in question 6.

---

6. Create suitable charts to visualise the following variables:

6.1 `cut`

6.2 `color`

6.3 `clarity`

[Hint: consider the data type of each variable before deciding on a suitable plot type.]

---

7. Instead of specifying the number of bins, you can specify the binwidth, using the `binwidth` argument in `geom_histogram()`.

Draw a histogram of `carat`, using a binwidth of 0.01.

[You will see that this is clearly not a suitable binwidth, but it does reveal an interesting pattern… what is this?]

---

8. Use boxplots to explore the relationship between price and colour.

[Hint: You have to use `geom_boxplot()` for this.]

Tip:

Remember that a boxplot can be used for a numerical variable. If you use `price` as your *y* aesthetic and colour as your *x* aesthetic, it will plot a separate boxplot (based on `price`) for each different colour category.

---

9. Explore the relationship between the price, carat and colour of diamonds in a suitable plot.

---

10. If you have time, you can try enhancing the plots you created above by adding suitable titles and labels.

After completing this week's practical and tutorial you should be able to:

* Calculate basic numerical summaries of data in R.

* Use plots to visually explore data

* Apply the general syntax of `ggplot2` to create different plot types

* Be able to customise plots by adding colour and labels