Tutorial 3 STQD6214

1. The `plot()` function serves as a basic tool for drawing a figure, either drawing points in the figure, or drawing a line.
   a) Run the following code first to assign the values to each variable. The values shown below are the number of students registered in a university for the year 2012 to 2019, for each category.

```
year <- 2012:2019
engineering <- c(20397, 21432, 22206, 20778, 22362,
                 23669, 23400, 24137)
science <- c(10680, 11561, 11690, 10592, 11858, 13100,
             11796, 12267)
technology <- c(5124,  6679,  6828,  6200,  9553, 10809,
                11396, 10682)
```

   b) Produce a scatterplot between the number of engineering students and the years.
   c) Modify your code to include a suitable title and the labels for *x*-axis and *y*-axis.
   d) How would you modify your code if you want to draw lines connecting each subsequent point?
   e) Play around and experiment with the arguments for the `plot()` function. You can change the arguments `type`, `col`, `cex`, `lty`, `lwd`, `pch`, `xlim`, `ylim`, and many more.
   f) Now, using the functions `points()` or `lines()`, add the plots showing the number of science and technology students for each year to the existing diagram. The plot should now consist of three lines/points, one for each category. Make sure the colours and characters used for each category are distinguishable. You may need to modify the `ylim` in the original `plot()` to fit all the lines/points.
   g) Using the `legend()` function, add suitable legends in the plot to show what each line/point refers to.

2. Load the `mtcars` dataset by running `data(mtcars)` in the R console. The help documentation gives details regarding this dataset. You can read it by running `?mtcars` or `help(mtcars)`.
   a) Draw a histogram for the miles per gallon (`mpg`) of the cars.
   b) The variable `am` is 0 if the car has an automatic transmission, and 1 if the car has a manual transmission. Draw a side-by-side boxplot to compare the miles per gallon (`mpg`) and the transmission of the car (`am`). Modify the labels for the boxplot.
   c) Draw a scatterplot for miles per gallon (`mpg`) vs displacement (`disp`).
   d) Draw a scatterplot matrix with the variables `mpg`, `disp`, `hp`, and `wt` in the diagram.

e) The code `table(mtcars$gear)` gives the frequency of cars with each of the number of forward gears. Using this frequency, draw a bar graph for the number of cars with the number of forward gears.

3. Import the file "property_sales.csv" into R, which we used in the previous tutorial, and save it to a data frame called `property_sales`. The dataset shows the sales for residential properties in four neighbourhoods together with their land and improvement values.
   a) Draw a scatterplot matrix to show the relationships between sales, land, and improvement values. Give this figure a good title.
   b) Create a new column in the data frame named `total` which calculates the sum of land and improvement values. The new column is the total value of the property. Draw a scatterplot of sales in the *y*-axis vs total values in the *x*-axis. Give the figure a good title and axis labels.
   c) Using `curve()` function with the argument `add=TRUE`, add a straight line $y = -16.5 + 1.36x$ to the figure drawn in (b). Give this line a suitable colour and other properties to make it nicer.
   d) Add a legend in the top left corner of the figure, to show the line created in (c) is a fitted line.
   e) Run the following code which calculates the difference between the sales value and the given straight line. Then, draw a histogram of the residuals, setting the *y*-axis to be the probability for each bin. Give this figure a good title.

```
residuals <- property_sales$Sales -
             (-16.5+1.36* property_sales$total)
```

   f) The code below gives the pdf of a Normal distribution with the given mean and standard deviation. Using this function, add the pdf line in the histogram drawn in (d).

```
dnorm(x, mean=mean(residuals), sd=sd(residuals))
```