**Data Analytics (CMP330)**

# Practical 3 – Descriptive analysis and visualisation using R

Review the lecture slides on Descriptive data analytics and visualisation to familiarise yourself with the content.

Remember to setup your working directory to your folder where script and data are saved e.g. P:\CMP330\week3

**Part A - Descriptive data analytics:**

**Dataset**

Use “usedcars” dataset – the csv file can be found under this week’s material in Blackboard which contains actual data about used cars advertised for sale on a popular U.S. website.

**Importing the CSV data file, saving the data to the usedcars data frame:**

**usedcars <- read.csv("usedcars.csv")**

**Exploring the structure of data**

The str() function provides a method for displaying the structure of a data frame,

or any R data structure including vectors and lists. It can be used to create the basic

outline for our data dictionary:

**str(usedcars)**

For such a simple command, we learn a wealth of information about the dataset.

The statement 150 obs tells us that the data includes 150 observations, or examples.

The 6 variables statement refers to the six features that were recorded in the data.

These features are listed by name on separate lines.

**Exploring numeric variables**

To investigate the numeric variables in the used car data, we will employ a commonly-used set of measurements for describing values known as summary statistics. The summary() function displays several common summary statistics. The six summary statistics that the summary() function provides are simple, yet powerful tools for investigating data. The summary statistics can be divided into two types: measures of centre and measures of spread.

**summary(usedcars)**

We can also use the summary() function to obtain summary statistics for several

numeric variables at the same time:

**summary(usedcars[c("price", "mileage")])**

**Measuring the central tendency – mean and median**

Measures of central tendency are a class of statistics used to identify a value that falls

in the middle of a set of data. Mean is defined as the sum of all values divided by the number of values. Another commonly-used measure of central tendency is the median, which is the value that occurs halfway through an ordered list of values. To obtain the mean and median in R, the mean() and median() functions can be used.

**mean(usedcars$price)**

**median(usedcars$price)**

**Measuring spread – variance and standard Deviation**

To obtain the variance and standard deviation in R, the var() and sd() functions can be used. When interpreting the variance, larger numbers indicate that the data are spread more widely around the mean. The standard deviation indicates, on average, how much each value differs from the mean. For example, computing the variance and standard deviation on our price and mileage variables, we find:

**var(usedcars$price)**

**sd(usedcars$price)**

**var(usedcars$mileage)**

**sd(usedcars$mileage)**

**Measuring spread – quartiles and the five-number summary**

The span between the minimum and maximum value is known as the range. In R, the range() function returns both the minimum and maximum value. Combining range() with the difference function, diff() allows you to examine the range of data with a single command:

**range(usedcars$price)**

**diff(range(usedcars$price))**

The first and third quartiles, Q1 and Q3, refer to the value below or above which one quarter of the values are found. Along with the median (Q2), the quartiles divide a dataset into four portions, each with the same number of values. The middle 50 percent of data between Q1 and Q3 is of particular interest because it itself is a simple measure of spread. The difference between Q1 and Q3 is known as the interquartile range (IQR), and can be calculated with the IQR() function:

**IQR(usedcars$price)**

The quantile() function provides a robust tool for identifying quantiles for a set of values. By default, the quantile() function returns the five-number summary including minimum, maximum, and three quartiles. Applying the function to the used car data:

**quantile(usedcars$price)**

If we specify an additional probs parameter using a vector denoting cut points, we

can obtain arbitrary quantiles, such as the 1st and 99th percentiles:

**quantile(usedcars$price, probs = c(0.01, 0.99))**

**Part B - Data visualisations:**

**Visualizing numeric variables – boxplots**

Visualizing numeric variables can be helpful for diagnosing many problems with data. A common visualization of the five-number summary is a boxplot or box-and-whiskers plot. The boxplot displays the centre and spread of a numeric variable in a format that allows you to quickly obtain a sense of the range and skew of a variable, or compare it to other variables.

To obtain a boxplot for a variable, we will use the boxplot() function. We will also specify a

couple of extra parameters, main and ylab, to add a title to the figure and label the y axis (the vertical axis), respectively. The commands for creating price and mileage boxplots are:

**boxplot(usedcars$price, main="Boxplot of Used Car Prices", ylab="Price ($)")**

**boxplot(usedcars$mileage, main="Boxplot of Used Car Mileage", ylab="Odometer (mi.)")**

**Visualizing numeric variables – histograms**

A histogram is another way to graphically depict the spread of a numeric variable. It is similar to a boxplot in that it divides the variable's values into a predefined number of portions, or bins that act as containers for values. We can create a histogram for the used car price and mileage data using the hist() function. As we had done with the boxplot, we will specify a title for the figure using the main parameter and label the x axis with the xlab parameter. The commands for creating the histograms are:

**hist(usedcars$price, main = "Histogram of Used Car Prices", xlab = "Price ($)")**

**hist(usedcars$mileage, main = "Histogram of Used Car Mileage", xlab = "Odometer (mi.)")**

You might also notice that the shape of the two histograms is somewhat different. It seems that the used car prices tend to be evenly divided on both sides of the middle, while the car mileages stretch further to the right.

**Visualizing relationships – scatterplots**

A scatterplot is a diagram that visualizes a bivariate relationship. It is a two-dimensional figure in which dots are drawn on a coordinate plane using the values of one feature to provide the horizontal x coordinates, and the values of another feature to provide the vertical y coordinates. Our hypothesis is that price depends on the odometer mileage. Therefore, we will use price as the y, or dependent, variable.

**plot(x = usedcars$mileage, y = usedcars$price,**

**main = "Scatterplot of Price vs. Mileage",**

**xlab = "Used Car Odometer (mi.)",**

**ylab = "Used Car Price ($)")**

**Ggplot visualisations**

Install ggplot by typing in the following code:

**install.packages("ggplot2")**

Once installed, you can **activate** the library using the following code:

**library(ggplot2)**

Now try this code (same as above only in ggplot):

**p <- ggplot(usedcars, aes(transmission, price))**

**p + geom\_boxplot()**

Now add a jitter to show density:

**p <- ggplot(usedcars, aes(transmission, price))**

**p + geom\_boxplot() + geom\_jitter(width = 0.2)**

Add a third dimension – model of used cars (model):

**p <- ggplot(usedcars, aes(transmission, price))**

**p + geom\_boxplot(aes(colour = model))**

**Exercise 1. Use the “train.csv” – the csv file for the Titanic dataset can be found under this week’s material in Blackboard. You should try to reproduce the results shown for this dataset in the week 3 slides.**

**R Notebooks**

So far you have been using script files for your code, but you can also create R Notebooks which allow code cells and output to be combined with text using Markdown (somewhat similar to Jupyter Notebooks). It is easy to create an R Notebook in R Studio. Go to *File – New File – R Notebook*. You can save this as a *.Rmd* file. You will see a code cell that looks like this:

```{r}

plot(cars)

```

You can click the green triangle at the right (“*Run Current Chunk*”) to execute this cell and generate the output (you can also use *Ctrl + Shift + Enter*). You can insert new code cells by going to *Code – Insert Chunk*. You can also click on *Preview* to see the notebook as an html file.

**Exercise 2. Spend some time familiarizing yourself with R Notebooks, see** [**https://bookdown.org/yihui/rmarkdown/notebook.html#using-notebooks**](https://bookdown.org/yihui/rmarkdown/notebook.html#using-notebooks)

**For example add some more code cells to include code from exercise 1 and generate the output. You should also explore how to format text in Markdown, see** [**http://rmarkdown.rstudio.com**](http://rmarkdown.rstudio.com)

**Exercise 3. Produce a report in the form of an R Notebook on exploratory data analysis of the “airquality” dataset which is available in R. Your notebook should include code, output generated from the code and formatted text to describe your work including observations about the dataset, i.e. what the results tell us about the dataset. Your report should include the following:**

* **General information about the dataset – examples of rows or data, details about the variables and number of observations**
* **Descriptive statistics – measures of central tendency and variation for relevant variables (the sapply function could be used)**
* **Data cleaning – replace missing values with the mean for the variable (see section 7 of Practical 2)**
* **Data visualization – scatter plots, histograms and boxplots**
* **Correlations between relevant variables**
* **Boxplots by month (you will need to convert the Month variable from an integer variable to a categorical variable – see slide 53, week 3)**
* **Density plots by month (see slide 48, week 3)**