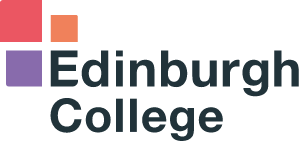
Data Science: ePortfolio 

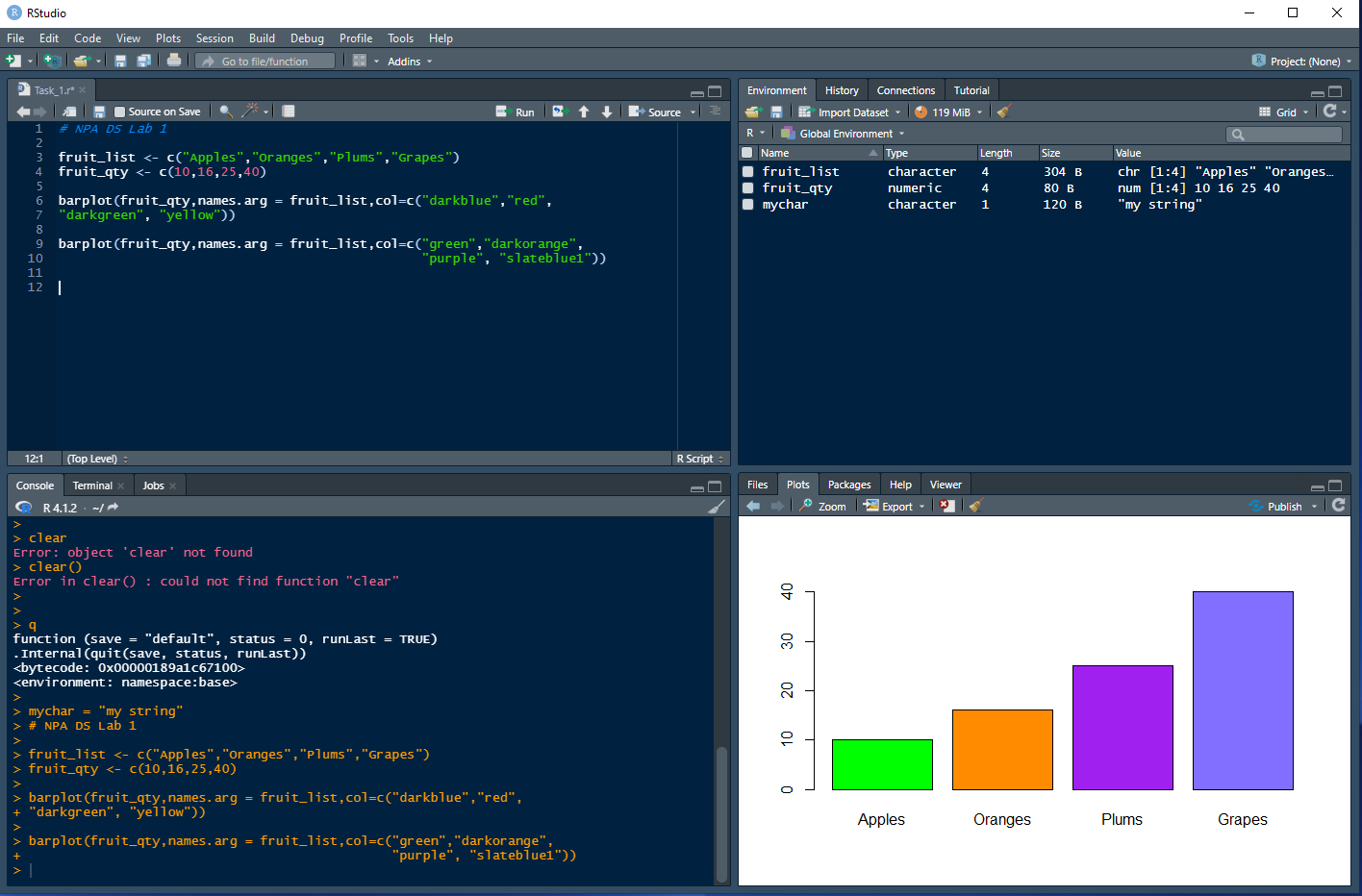
|  |  |
| --- | --- |
| First Name (s) | Kevin |
| Last Name (s) | Gemson |
| EC Email Address | EC2111447@edinburghcollege.ac.uk |

# Guidance - How to complete this document

* Check the course Moodle page to see what evidence is required.
* All screenshots should be cropped and labelled to show the key information.
* The lab title and date of completion should be added above each section.
* Text should be in 11pt or 12pt, any code should be in a monospaced font.
* Each week complete the summary with an overview of the tasks completed.

# Lab 1: Data In Society - Date Completed: 05/02/2022

1. A screenshot of RStudio Running on your computer



|  |
| --- |
| 1. Your plot from step 1.11 of the lab 2. Your plot from step 1.12 of the lab |

## Weekly summary, complete the following:

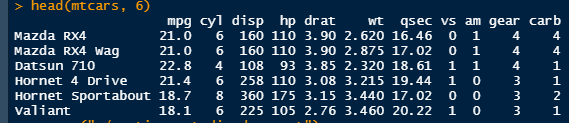
This week I learned about ... principles of Data Citizenship, and how data is collected and used (and can be misused) in Society. Also, learned the definition of Data Science and its relationship to machine learning and artificial intelligence.

In the lab I … learned how to install R and R Studio, and run a number of tasks to understand how the basic syntax works. Also followed a number of YouTube tutorials including the extra learning module, to try to get a better understanding of what R can do.

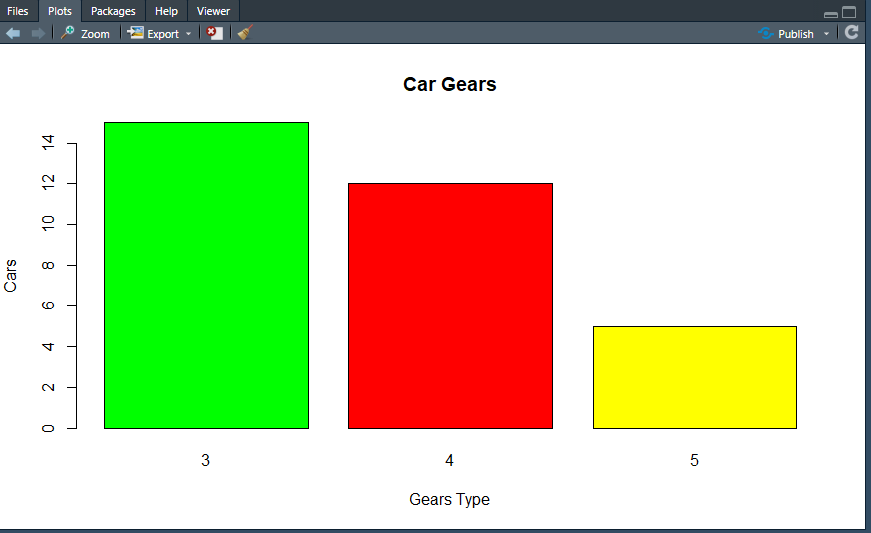
I enjoyed.. getting to see the possibilities of what R can do, being able to produce graphs with some fairly simple commands

I found it challenging when… working through some of the different components of the syntax (vectors matrices, datasets, etc), especially as the syntax is different to other languages I am more familiar with.

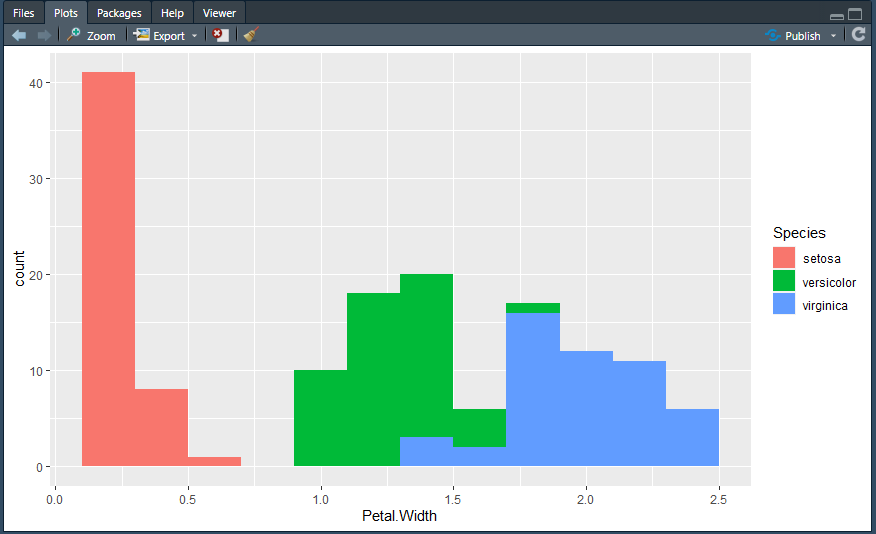
# Lab 2: Working With Data - Date Completed: 10/02/2022

1. A screenshot of running 1.4, how many mpg's was the Mazda RX-4? **- 21.0 mpg**
2. Add an image of, and the associated R code to create the bar plot from step 1.9.

barplot(mtcarsgear, main = "Car Gears", ylab = 'Cars', xlab = "Gears Type", col=c('green','red','yellow'))

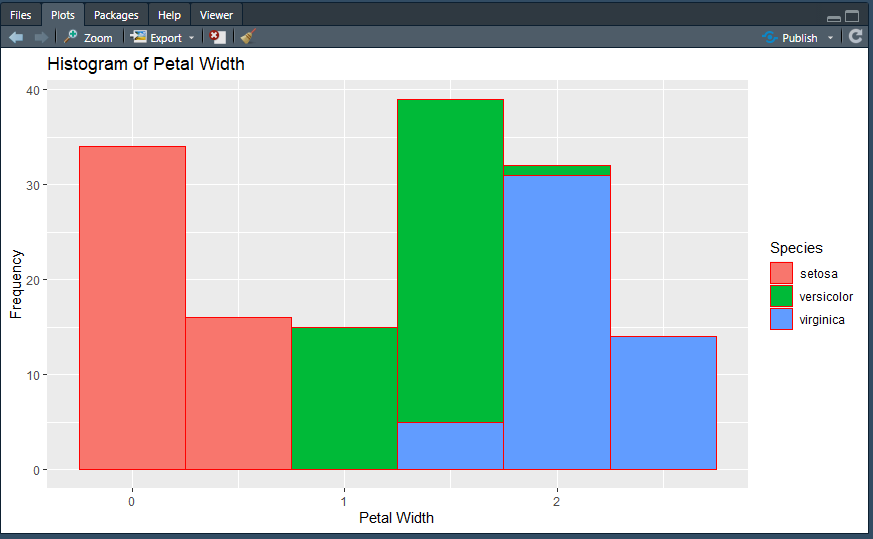


1. Add an image of the histogram from step 2.8.



1. Add an image of the chart from step 2.11, and add the associated code to create this. Add an explanation of what each part of the code does.

histogram + geom\_histogram(binwidth=0.5,color="red",aes(fill=Species)) + xlab("Petal Width") + ylab("Frequency") + ggtitle("Histogram of Petal Width")



Explanation of code -

histogram – variable defined earlier in code which is assigned a ‘ggplot’ graph based on iris’s PetalWidth values

geom\_histogram(binwidth=0.5,color="red",aes(fill=Species)) - displays the counts with bars. Width of bins is 0.5, outline of each bin is to be red. ‘aes’ = aesthetics; using the ‘Species’ data from ‘iris’ as the fill value groups the data together, based on the values in that field and overlays each set of values.

+ xlab("Petal Width") - sets X-axis label

+ ylab("Frequency") - sets Y-axis label

+ ggtitle("Histogram of Petal Width") - sets title

## Weekly summary, complete the following:

This week I learned about ... the applications of Data Science and the Data Science lifecycle, and how this compares with regular software development projects. I also learned about the four Vs of Data, and about different types of bias, and how these can affect outcomes in a Data Science project.

In the lab I … found out how to import datasets and convert these into bar charts with some basic design elements. Then, I was able to import an additional library and use it to convert the data into a histogram with some additional formatting

I enjoyed ... seeing the types of graphs and charts that can be produced with some fairly simple commands, and seeing how these can be enhanced with the addition of extra parameters.

I found it challenging when … trying to work out some of the syntax (for example, the ‘sample’ statement from the ‘iris’ dataset)

# Lab 3: Management And Security - Date Completed: 24/02/2022

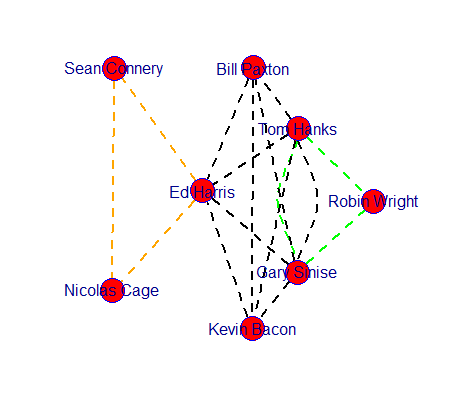
1. A screenshot of running 1.4, the head command running in part 1.4



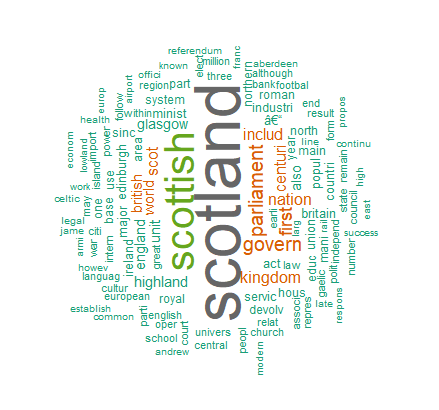
1. The finalised plot from part 1.8, with the code used to create it

plot(actorNetwork, vertex.color="Red", vertex.frame.color = "Blue",

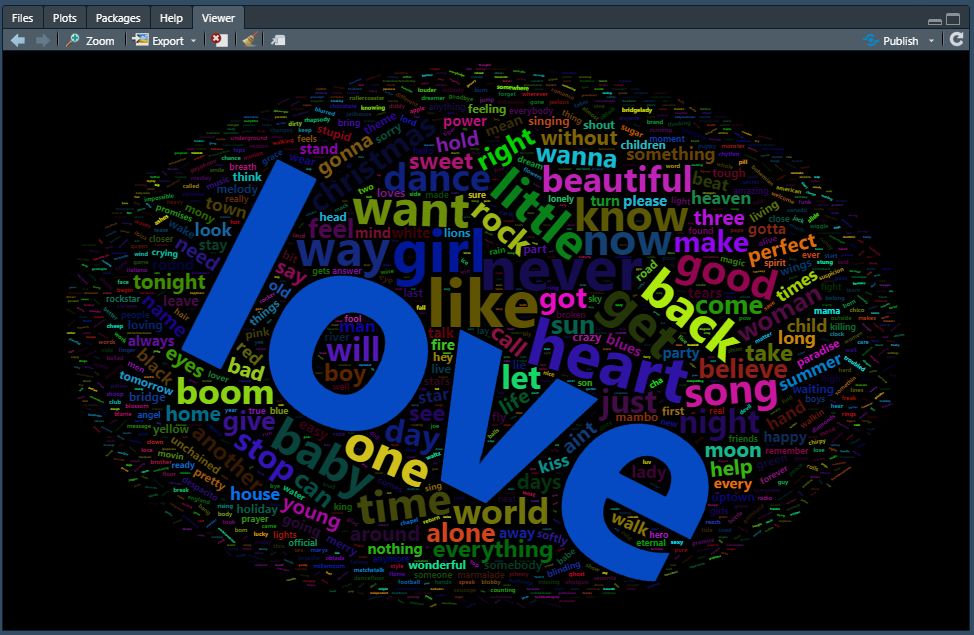
vertex.size = 19, vertex.label.family = "Arial", edge.width=2, edge.lty=2)

1. One of the word clouds created in step 2.5

Example 3 - Improved cloud with colour palette:



1. Find a news or magazine article, convert the article to a text file and use the skills from this lab to create a wordcloud of this.



1. Add an explanation of why you chose the text that you have chosen, what are the most common words, was this expected? - the text I chose is a page listing all the UK number 1 singles since the music charts were first compiled in 1952. I thought it would be interesting to see which words occur the most in the titles of these singles. The results show that, as expected, the word ‘love’ is by far and away the most commonly used word (97 times), but following that, there is a huge gap to the next most common word (‘like’, used 23 times), then there is very small differences in the number of word following that (‘heart’ used 22 times, ‘never’ used 21 times, ‘girl’ used 20 times). I had to try changing some of the settings, such as with stemming and removing stopwords to get a decent result, but this may be skewing the results (for example, ‘the’ and ‘I’ are very common words that are removed from the final output). I would maybe have to try changing the settings to get satisfactory results, but think the wordcloud above shows some useful information.

## Weekly summary, complete the following:

This week I learned about ... a number of aspects of Data Management and Security, including methods of data management/data governance and data security and understanding data privacy. Also, the rights and responsibilities of data subjects and data owner and understanding what legislation is in place to protect data.

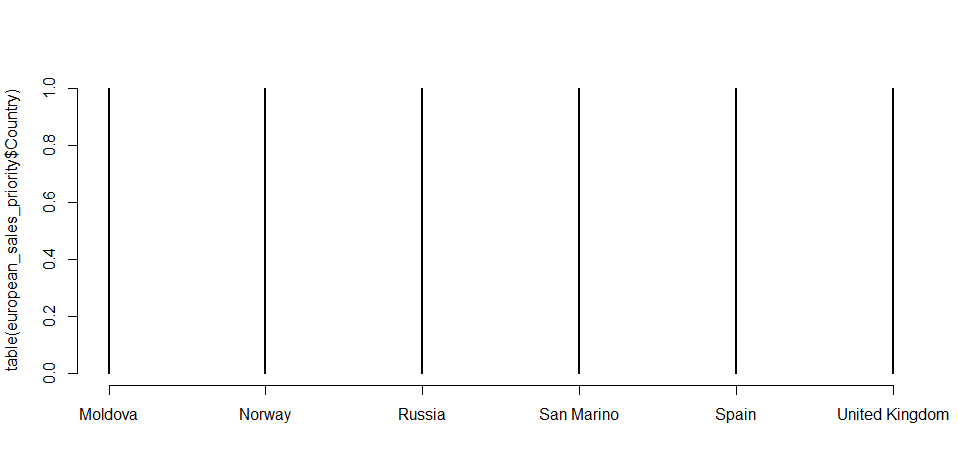
In the lab I … tried to use various R packages and libraries to produce visualizations of textual information, i.e. igraph functions and wordcloud packages

I enjoyed... seeing what can be done with the various packages available to R – the wordcloud, for example, seemed to produce some impressive output with a small number of commands

I found it challenging when … it took some time to try to understand what was happening in the igraph section, as some of the parameters being passed in were not obvious. Also, I had a number of technical issues with installing some of the libraries and ended up having to reinstall some of the software. There were a few other issues such as getting RStudio to find font packages, which required a bit of effort to try to understand and fix. This is probably all good practice in identifying faults, reading error messages and trying to find fixes, however.

# Lab 4: Title – Open Data - Date Completed: 07/03/2022

1. The plot created in step 1.13...



… and the code used to create this -

attach(X100\_sales\_records)

european\_sales\_priority <- X100\_sales\_records[Region == "Europe" & `Order Priority` == "L" , ]

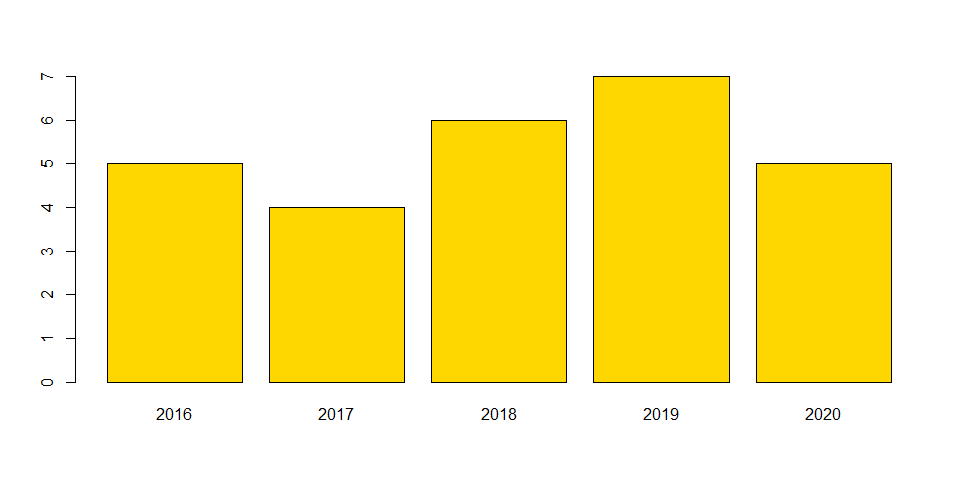
plot(table(european\_sales\_priority$Country))

1. The command used to create the table from step 1.14

barplot(table(X100\_sales\_records$Country,X100\_sales\_records$`Item Type`))

1. The plot from 3.6 and the code used to create it

barplot(my\_group\_data\_new,col="gold1",names.arg=colnames(df\_group[-1]))



1. An explanation of the $, then the [ ] and  [[ ]] and what we can use these for in R

R has a number of datatypes which contain multiple data elements, including vectors, lists, data frames and matrices. R provides several constructs which allow access to individual elements or subsets of these datatypes through indexing operations. These are the [, [[ and $ operators. For example -

x[i]

x[i, j]

x[[i]]

x[[i, j]]

x$a

x$"a"

These are all used to access elements of a data structure, but with some differences. Generally, the single [ will return a subset of an object of the same type. So using [] on a vector will return a vector, and using [] on a data frame will return a data frame. The single square bracket should be used for singular indexing, i.e. returning a single object from a vector or matrix, where all elements are of the same type. It can also be used for indexing by vectors, and will return a list of all the indexed elements.

The double square brackets are used to index one element from a recursive data structure (i.e. where elements may not be of the same type) such as a list or data frame. It will return the individual element, such as a character or an integer, rather than a list or other data structure. The double square bracket cannot be used with a vector index.

$ is a special case of [[ in which you access a single item by actual name. It is used for recursive objects such as lists or data frames. It allows only a literal character string or a symbol as the index, such as a column header.

1. An explanation of the indexing of a vector in R, how are vectors indexed?

Vectors are data structures containing elements of the same type, such as integers or characters. Vectors are indexed using single square brackets. The index value within the square brackets can be a single integer or a vector, such as a range of elements to be selected. It could also be a negative integer, and this will return all elements except those indicated by the negative value.

## Weekly summary, complete the following:

This week I learned about ... different data types such as qualitative and quantitative, structured and unstructured etc. Also learned about types and sources of open data and about data ethics, i.e. the rights and wrongs of holding and using data

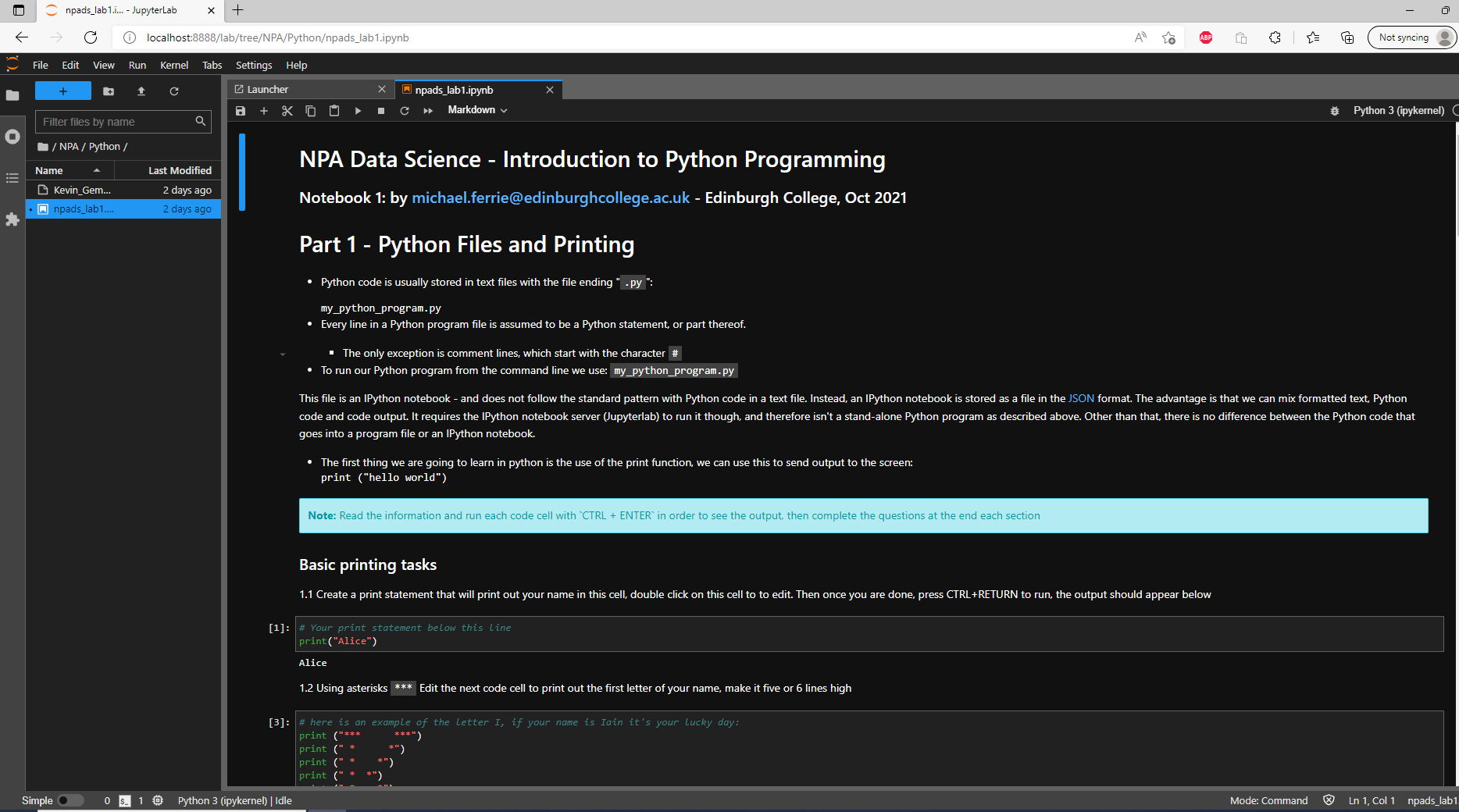
In the lab I … imported data from outside sources, converted this into R objects and used these to access individual elements and produce visual representations of the data.

I enjoyed ... finding out more about how to import data from files and access individual elements of it to produce R data structures

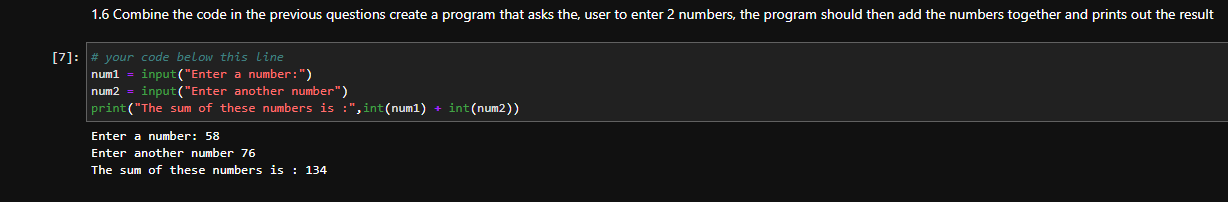
I found it challenging when … trying to understand some of the syntax such as indexing data frames and accessing individual column elements, particularly when using more than one condition in the command.

# Lab 7: Data Capture & Modelling - Date Completed: 31/03/2022

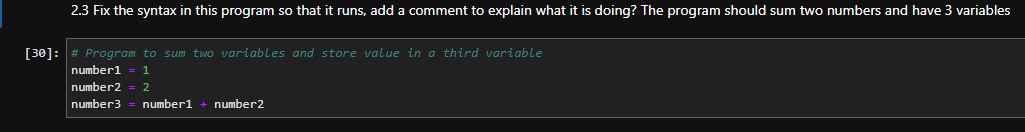
1. A screenshot of Jupyterlab running on your computer



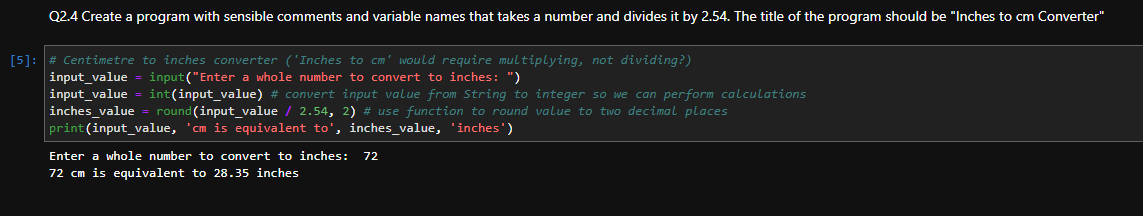
1. A screenshot of your answer for question 1.6



1. A screenshot of your answer for question 2.3



1. A screenshot of your answer for question 2.4



## Weekly summary, complete the following:

This week I learned … how to install Jupyterlab and get Python scripts running on it. From the PDF, I learned about different data capture techniques, as well as different ways of cleaning and validation. Also reviewed Python basics from the YouTube video

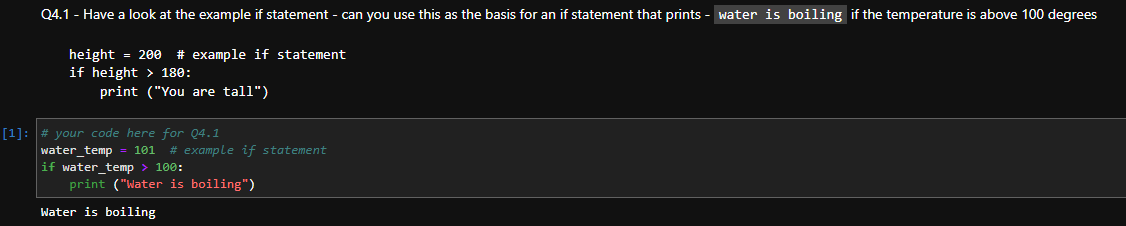
In the lab we … installed Jupyterlab and ran some basic Python commands

I enjoyed… learning about different data capture and cleaning techniques; having worked in a company that handles and processes large volumes of data, it is interesting to see the different techniques available, many of which I have not come across before. The company has tended to use traditional methods of handling data in the past, and it will be interesting to see which of these I may come across in the future.

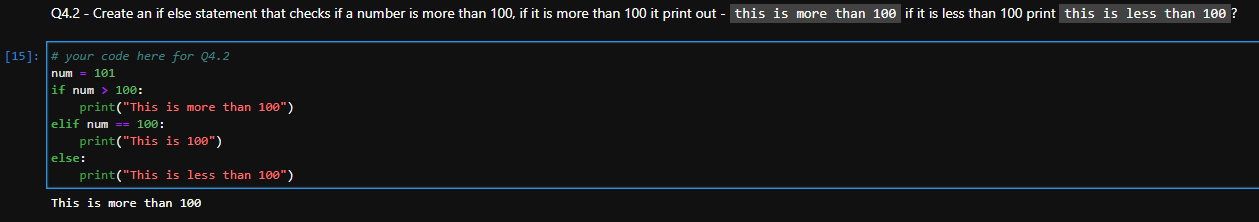
I found it challenging when… I had some issues getting Jupyterlab to work; I had used Python before, running scripts through Visual Studio Code, but Jupyterlab works in a different way and took a bit of getting used to.

# Lab 8: Data Analysis & Statistics - Date Completed: 03/04/2022

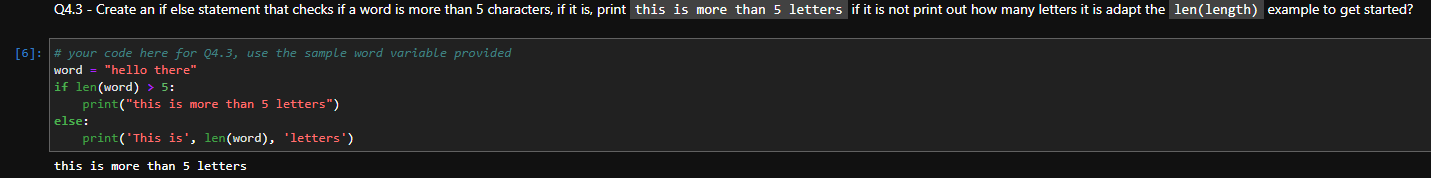
1. A screenshot of the code in the code cell and the output for question 4.1



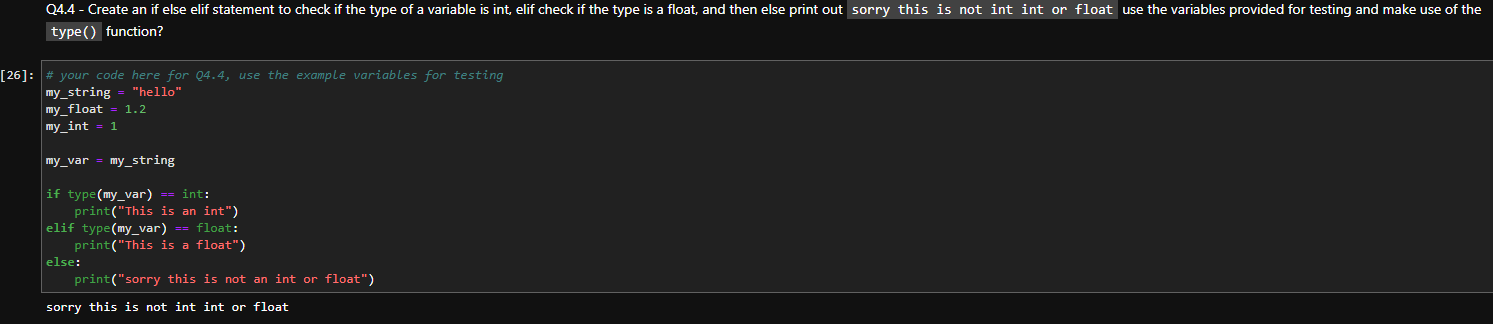
1. A screenshot of the code in the code cell and the output for question 4.2



1. A screenshot of the code in the code cell and the output for question 4.3



1. A screenshot of the code in the code cell and the output for question 4.4



## Weekly summary, complete the following:

This week I learned … about the different steps in Data Analysis and how these can lead to allowing us to making predictions about future trends.

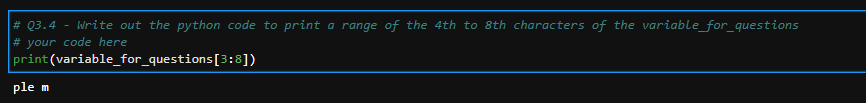
In the lab we … revised some basic operators and ran some scripts to demonstrate handling of different data types and conditionals

I enjoyed… revising some of the basic commands and taking on some of the programming challenges.

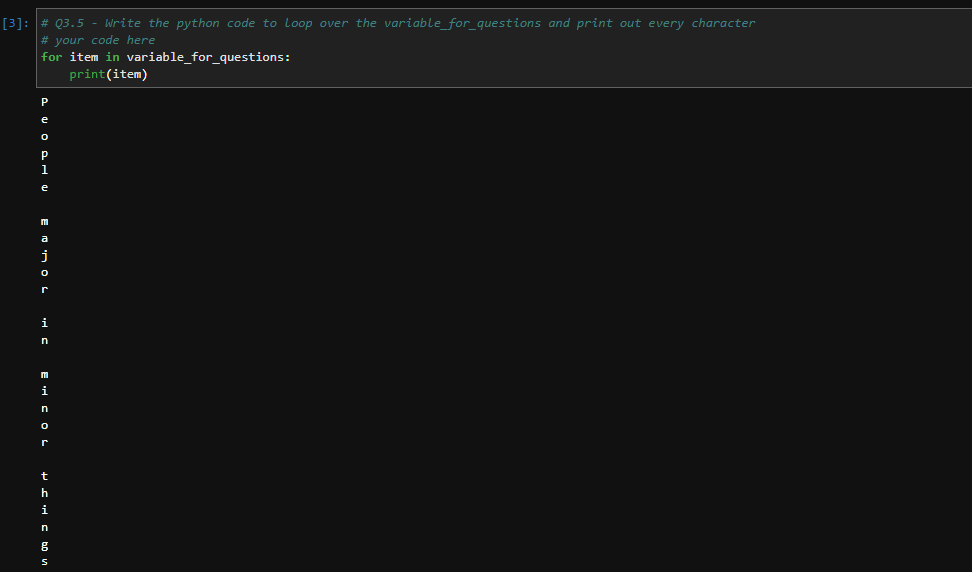
I found it challenging when … trying to get the appropriate data type for many of the scripts (e.g. converting string to int and performing division to get a float). Also considering the different options when using conditionals to ensure all outcomes are catered for.

## Lab 9: Title – Data Visualisation and Storytelling - Date Completed: 07/04/2022

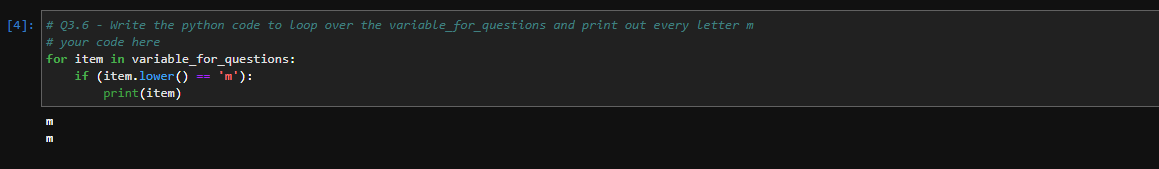
1. A screenshot of the code in the code cell and the output for question 3.4



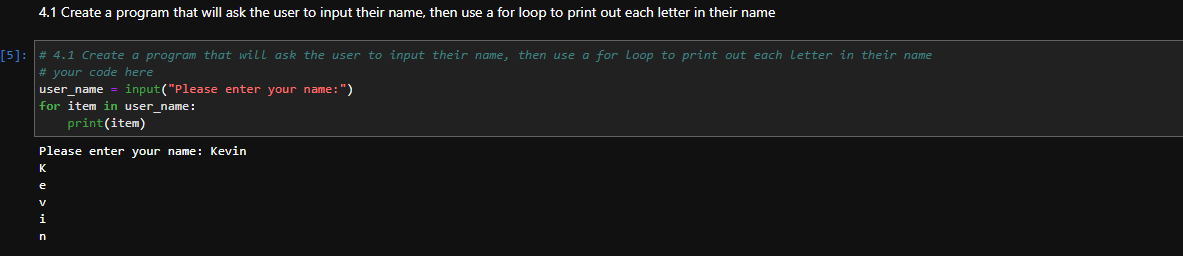
1. A screenshot of the code in the code cell and the output for question 3.5



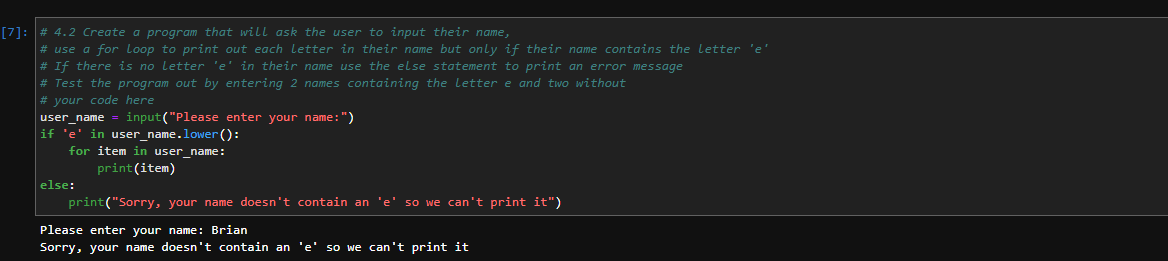
1. A screenshot of the code in the code cell and the output for question 3.6



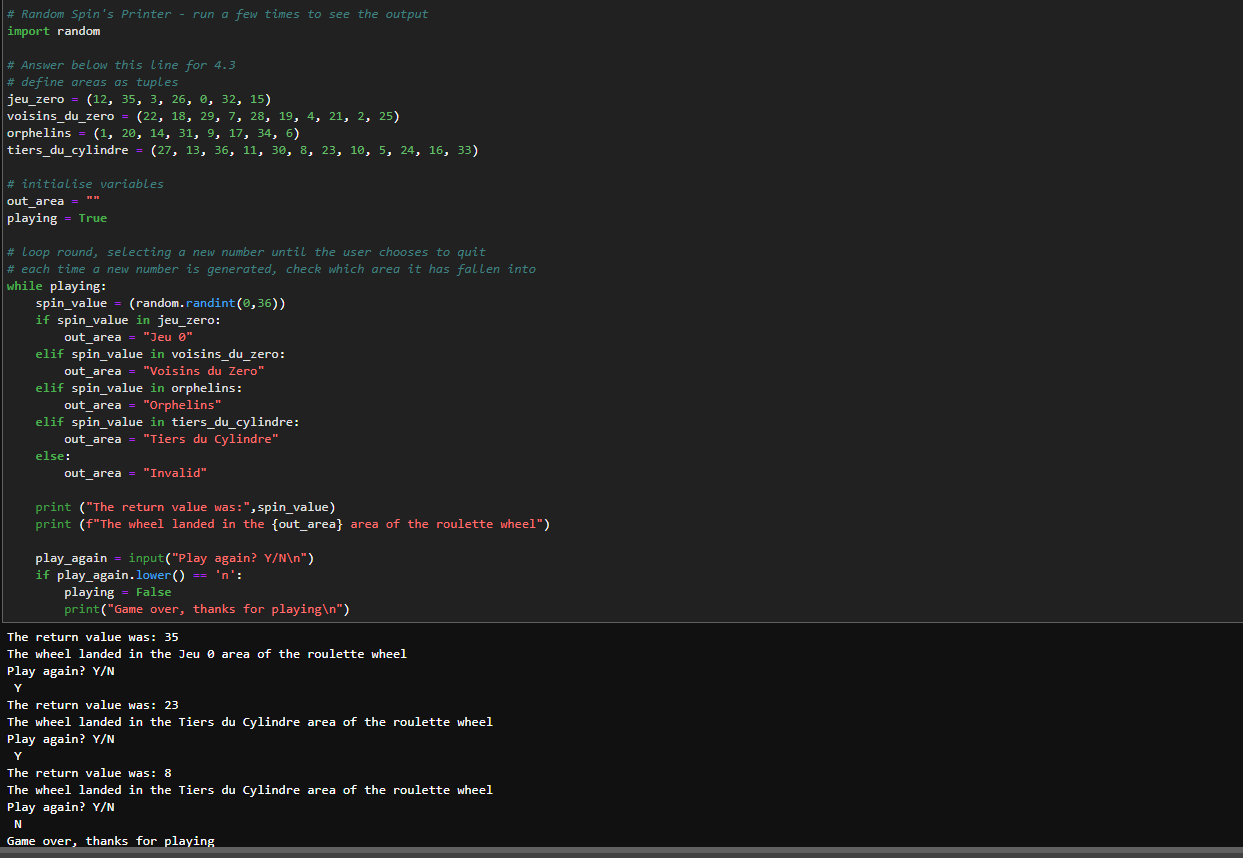
1. A screenshot of the code in the code cell and the output for question 4.1



1. A screenshot of the code in the code cell and the output for question 4.2



1. A screenshot of the code in the code cell and the output for question 4.3



## Weekly summary, complete the following:

This week I learned… about data visualisation and the different options for presenting data based on what the data is trying to present. In Python, we learned about extracting individual elements from lists and also program control – for loops in conjunction with if/else statements.

In the lab we… ran a number of scripts to extract individual characters of ranges of characters from a string variable; also ran some scripts to take input data and process differently based on the values provided. Final challenge was to randomly generate a data value and handle the output based on the value.

I enjoyed… the challenge of working through the different possible outcomes and writing code to handle these.

I found it challenging when… trying to work out the indexing values - I have used other programming languages that index starting at 1 and where ranges are inclusive, so ensuring I am extracting the correct values can take some thought

## Computer Programming Lab 1: Title – Computer Programming 1

## Date Completed: 12/05/2022

The flow chart and decomposition for 2.2, 2.3 and 2.4

2.2 The next algorithm to create is one to check if a word contains a vowel ‘aeiou’:

Write out your algorithm in decomposition notation.

> Read input word, set ‘match’ flag to ‘False’

> Check each letter to see if it is in ‘aeiou’

>> read next letter of word

>> if no more letters, exit loop

>> else, check to see if letter is in ‘aeiou’

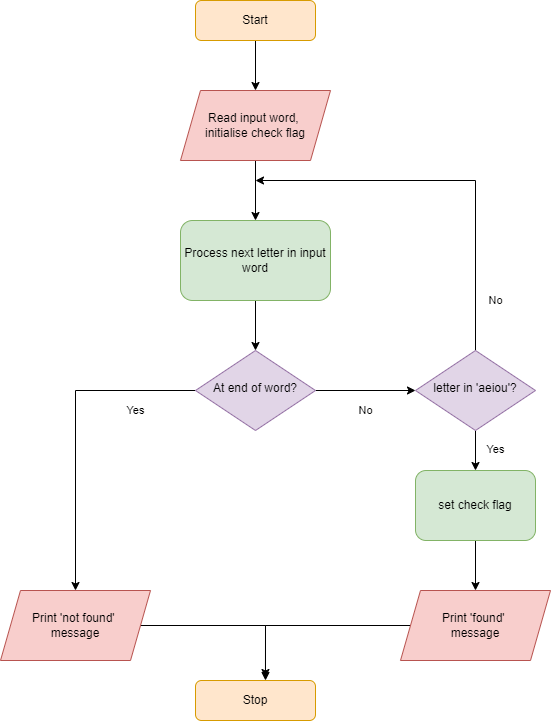
>>> if yes, set ‘match’ flag to ‘True’ and exit loop

> Print output message

>> if ‘match’ flag is ‘True’, print ‘match’ message

>> else print ‘no match’ message

Draw your algorithm as a flowchart.



2.3 This algorithm should be an enhancement of the previous, in 2.2 you created an algorithm to determine if a word contained a vowel, can you improve this and say how many vowels - if any - a word contains?

Write an algorithm that is an improvement of the algorithm in 2.2 in decomposition notation; this should now calculate the number of vowels in a word.

> Read input word, set counter to zero

> Check each letter to see if it is in ‘aeiou’

>> read next letter of word

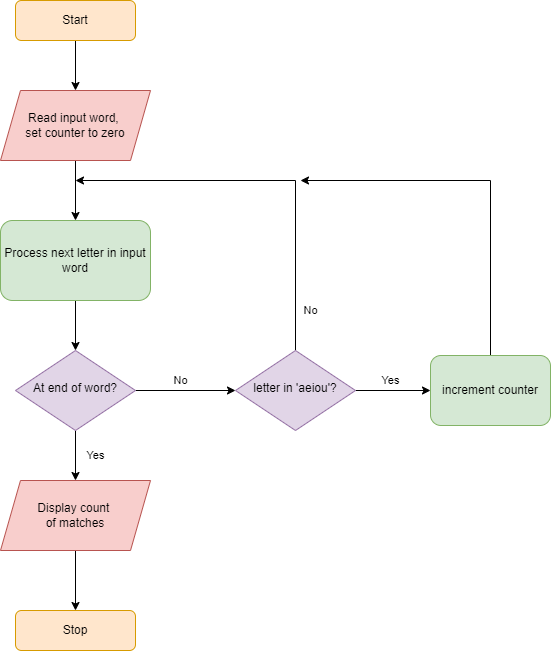
>> if no more letters, exit loop

>> else, check to see if letter is in ‘aeiou’

>>> if yes, increment counter

> Print output message containing count of matches

Write a flowchart to represent your algorithm.



2.4 The next algorithm we want to create is an algorithm to check if a number can be divided by 3 without a remainder. The algorithm should then check if the number can be divided evenly by 4. Then the algorithm should check if the number can be divided evenly by both 3 and 4.

Write out your algorithm in decomposition notation.

> Read input number, initialise booleans to ‘False’

> Check number is divisible by 3

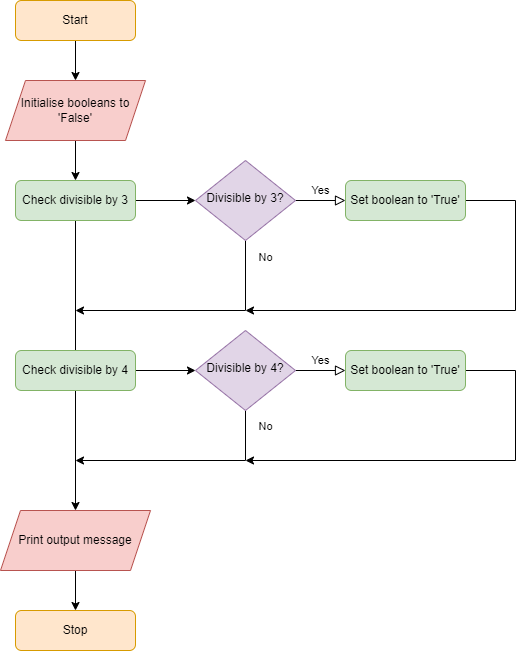
>> if modulus of input number divided by 3 is equal to zero, set ‘divisible by 3’ boolean to ‘True’

> Check number is divisible by 4

>> if modulus of input number divided by 4 is equal to zero, set ‘divisible by 4’ boolean to ‘True’

> Print output message containing ‘true’ or ‘false’ result for both booleans

Draw your algorithm as a flowchart.



## Weekly summary, complete the following:

This week I learned … about the ways of presenting an algorithm, both using decomposition notation and flowcharts.

In the lab we … took a number of programming challenges and presented these as algorithms in both formats

I enjoyed… the process of taking lines of code and working out the steps involved in producing an algorithm. Normally I would try writing lines of code first rather than working things out in a document, so it is a different way of programming.

I found it challenging when… trying to follow the rules for producing a flowchart in particular, i.e. where to put a decision, whether something is an input/output or a step, how to get two outputs from a decision etc. Found it much easier to write the decomposition notation.

An explanation of two different techniques for representing algorithms, how can we express an algorithm, why would we use these techniques?

- The two different techniques are decomposition notation and flowcharts. Decomposition uses text to declare the individual steps in producing the end result, and breaks these down into substeps using indentation to indicate a substep. Flowchart represents the steps in the algorithm as a chart using standard notation for presenting input/output, decisions, processes and start/stop. The first technique is more informal and probably used at a very early stage when trying to work out how to present all the steps involved in completing a task, and is easier to alter, e.g. to add or remove steps. The second type is a more formal document using standard notation and is probably used at a later stage in the design process, i.e. after the more informal earler stage of working out what steps are required. It is more difficult to alter and is used to present a plan for completing a task to others such as coders who may not be involved in the planning stage.

An explanation of the following three concepts:

- **algorithmic efficiency**: Algorithmic efficiency is a calculation of the amount of programming resources required by an algorithm to complete all the required steps. It uses mathematical formulae to estimate the amount of resources such as processing power, memory and time to establish whether an algorithm is likely to perform within a required set of constraints when run as code. It may help determine which of a number of possible algorithms are preferred, depending on which resource is considered more valuable.

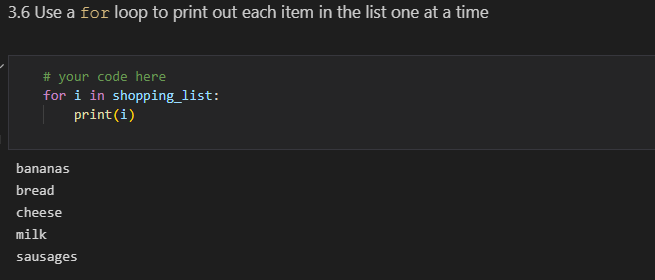
- **syntax**: Syntax is the set of rules which determine how words and symbols are interpreted by a computer language and combined into a program. Each language has its own set of rules unique to it which determine how code is interpreted and expression or line of code must be presented in accordance with these rules, otherwise the program will not be understood by the language software and will present an error when a user tries to run or compile it.

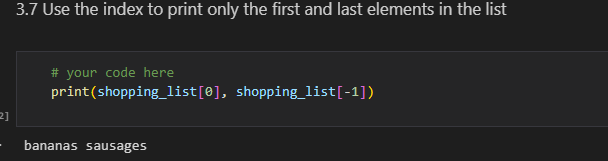
- **semantics**: In programming, semantics refers to the meaning of the code within a program. While the code may be syntactically correct (i.e. it follows the required syntax rules for the programming language, and is accepted by the software as executable code), it may still be semantically incorrect if it does not consider whether each element of the code makes sense and carries out all the required steps in the correct order. When a semantic error occurs, normally the code will execute successfully but will not give the expected result to the user.

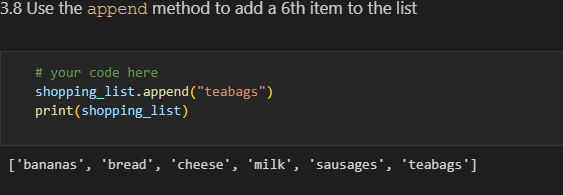
## Computer Programming Lab 2: Title – Computer Programming 2

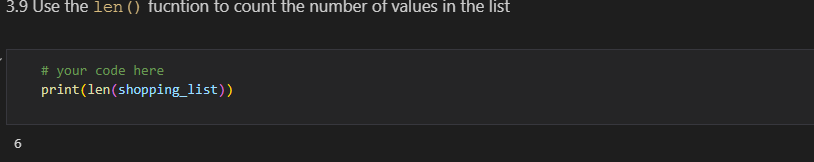
## Date Completed: 19/05/2022

- Your answers for questions 3.6,3.7,3.8,3.9

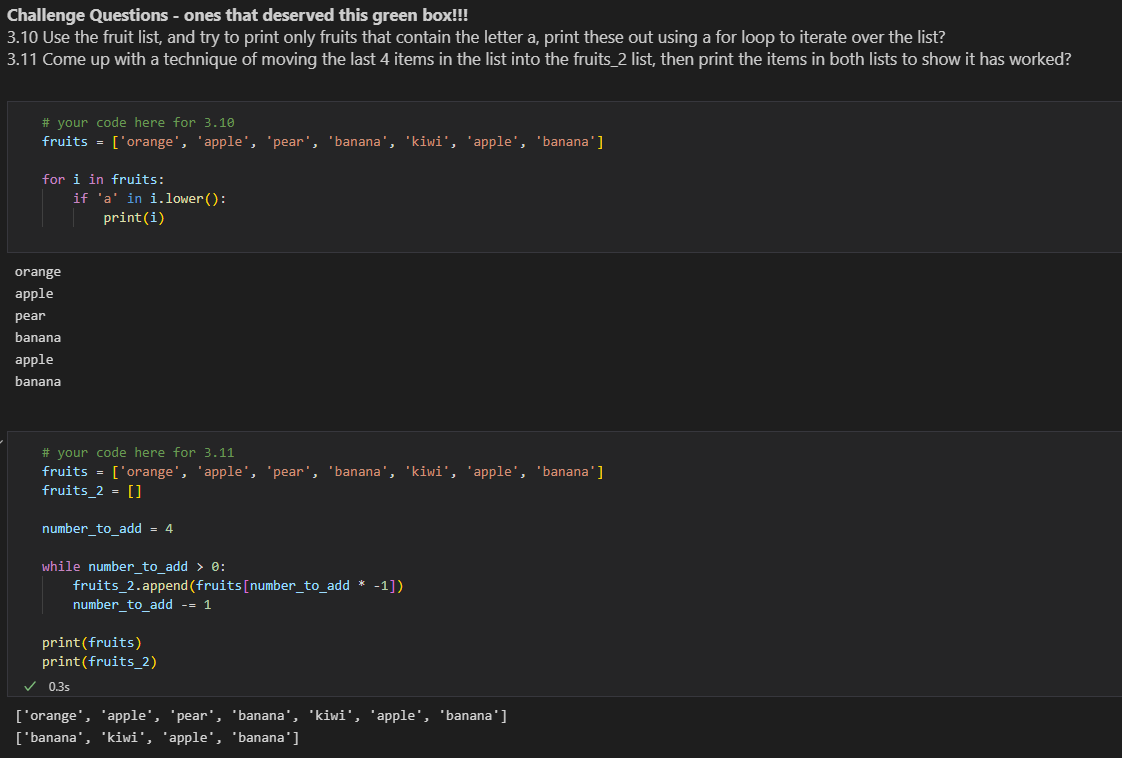








- Your answers for the challenge questions 3.10 and 3.11



- An explanation of the following programming concepts:

* **Iteration**: this refers to completing a task a number of times, for example executing a piece of code a specific number of time or executing it until a certain condition is identified.
* **data types:** these are a way of identifying the ‘type’ of an element of data. The different data types have their own characteristics and are used differently by the programming language, for example numeric data types can have mathematical calculations performed on them while character data types cannot.
* **operators:** operators are symbols used by the programming language to identify what action to take with variables and values in a program, such as assigning a value or performing a mathematical operation.

- An explanation of the order of evaluation in python, in what order are arithmetic operators evaluated:

In Python, operators are evaluated using the following order:

Order of operation - PEMDAS

1. Parentheses ( )

2. Exponent \*\*

3. Multiplication \*

4. Division / // %

5. Addition +

6. Subtraction -

After the PEMDAS order goes left to right. To override an order, an expression should be enclosed in parentheses.

- An explanation of the purpose of internal documentation (comments) in a program:

The purpose of comments in a program are to aid readability and understanding of the code. A comment is an explanation or annotation in the source code, readable only to the programmer. They are added to make the source code easier for humans to understand, and are generally ignored by compilers and interpreters.

## Weekly summary, complete the following:

This week I learned … more basic elements of the Python language such as data types, operators and lists.

In the lab we … generated and ran code to carry out a few challenges using loops and list elements

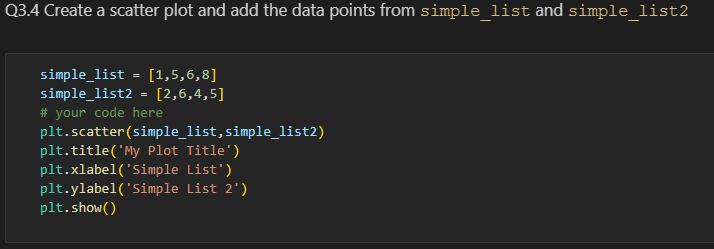
I enjoyed… the challenge of trying to work out some of the more difficult tasks, such as combining conditionals with loops to only access certain list elements

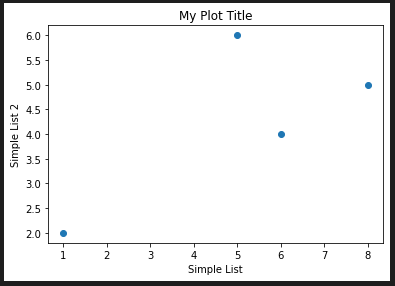
I found it challenging when… trying to work out the most efficient way of accessing some of the list elements, particularly when accessing elements at the end of a list rather than at the start.

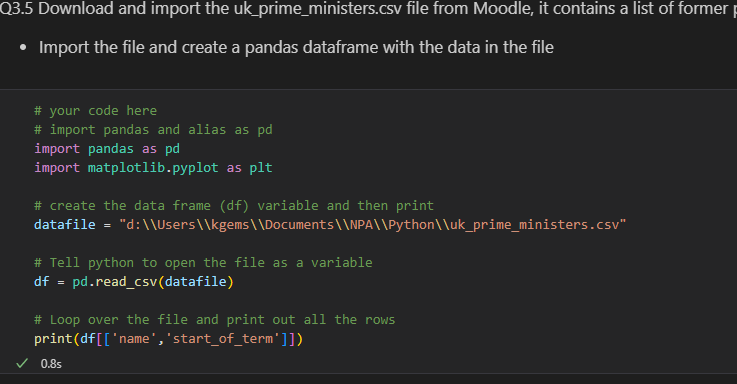
## Computer Programming Lab 3 Title – Computer Programming 3

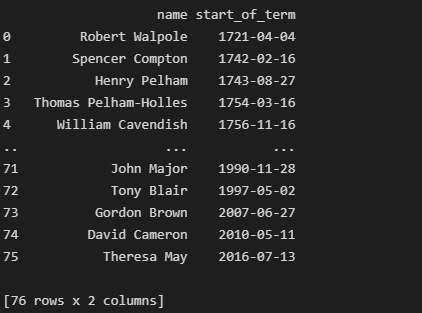
## Date Completed: 26/05/2022

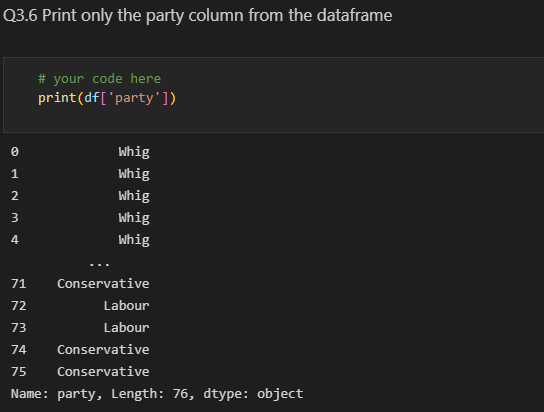
## - A screenshot of the code and the output from Q3.4-3.7

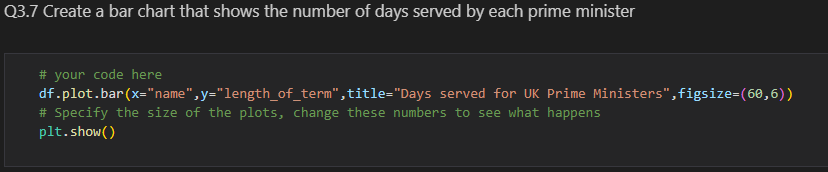


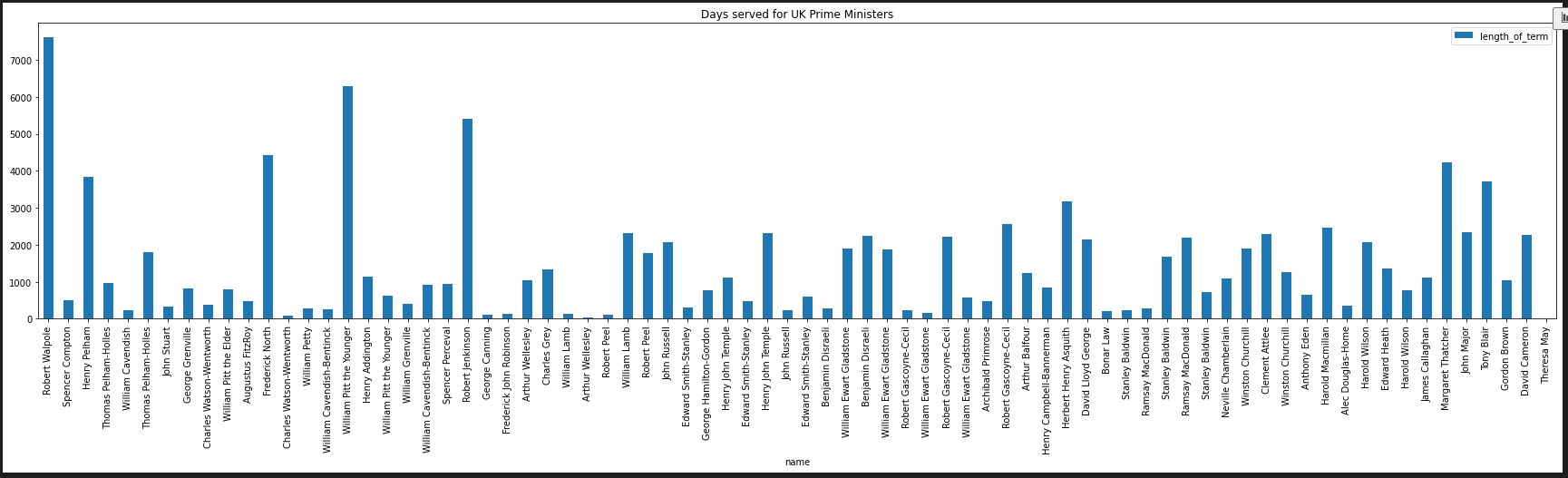






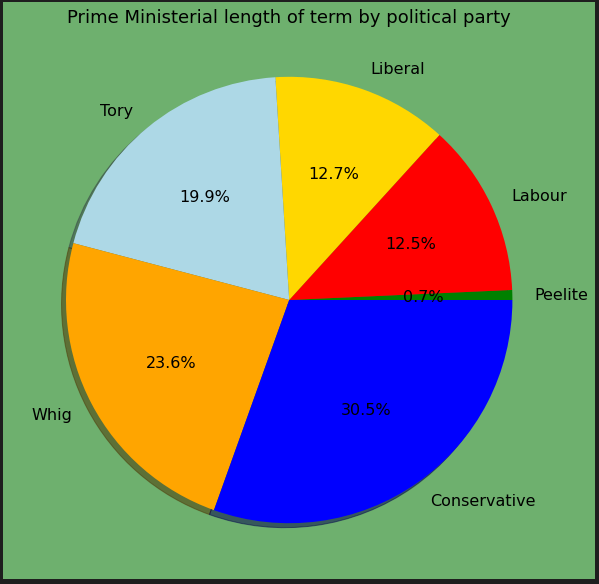






- A screenshot of the code and the output from Q3.8 challenge question





Weekly summary, complete the following:

This week I learned … about the libraries available to be imported into Python to present data in a visual way. Also learned about using dataframes and the operators available such as sorting and extracting data to use as input to the visual presentation libraries.

In the lab we … wrote and ran a number of pieces of code to present data in different forms, such as line graphs, bar charts, scatter graphs etc.

I enjoyed… seeing the complex output that can be generated from a few lines of code using imported libraries

I found it challenging when… trying to work out how to use the different parameters to resolve issues with the output, for example, getting sizing right, setting colours and fonts, etc.