Visualisation in R Part 1

Lab 6

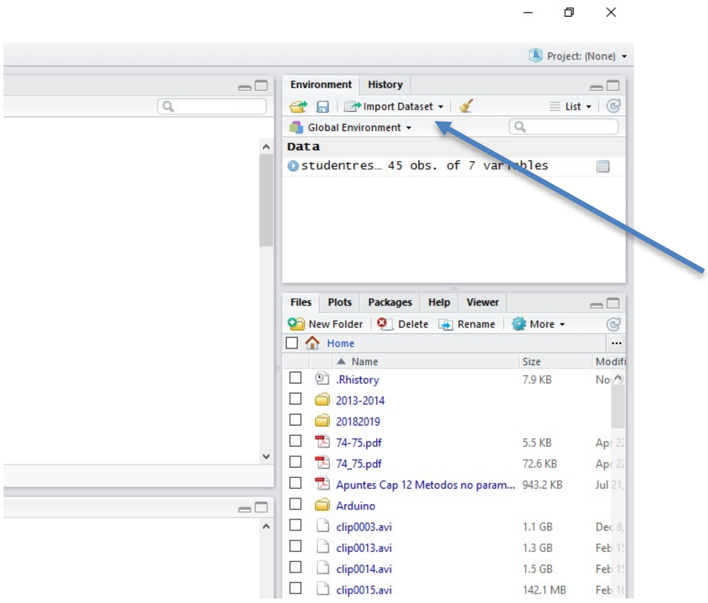
In this lab you will use ggplot and sqldf to analyse student marks in a very small file.

Rstudio Interface

- Data Pane is Top Left
- Environment and History Pane is Top Right
- Command Console is Bottom Left
- Visualisation, Package and File Pane is Bottom Right

Import the data

Click on Import Datasets and import your dataset into RStudio



```
Data
studentres… 45 obs. of 7 variables
```

	Name	Size	Modifi
□	.Rhistory	7.9 KB	No
□	2013-2014		
□	20182019		
□	74-75.pdf	5.5 KB	Apr 2
□	74_75.pdf	72.6 KB	Apr 2
□	Apuntes Cap 12 Metodos no param…	943.2 KB	Jul 21,
□	Arduino		
□	clip0003.avi	1.1 GB	Dec 8,
□	clip0013.avi	1.3 GB	Feb 1
□	clip0014.avi	1.5 GB	Feb 1
□	clip0015.avi	142.1 MB	Feb 1

Immediately the data is visible in the Data Pane on the top left.

In the console Pane, use the command:

```
View(studentresult)
```

This command can be used to quickly view a dataset that has been loaded into RStudio. Alternatively, you can load the dataset using read.table(… or read.csv(..

```
studentresult<-  read.csv( 'C:\\.your location..\\studentresult.csv'  ,  sep= ','  , header=T  )
```

R loads the data into a structure called a data frame. Data frames are essentially a collection of vectors and like standard data tables consist of different types of data primitives. To get more information about your data frame you can use:

```
str(studentresult)
```

To see different aspects of a data frame you can do the following, see number of columns, number of rows, and length:

ncol(studentresult)

nrow(studentresult)

length(studentresult)

To get the names of each column:

names(studentresult)

To get the dimensions (number of rows and cols):

dim(studentresult)

The main data structures in R are:

- **list** - Contains data elements of any type
- **vector** - Sequence of data elements of the same basic type
- **matrix** - Vector with additional dimensions
- **data frame** - Used to store data tables, list of vectors of equal length

Coercion is used to transform one type to another:

- **as.matrix** - Changed vector into a matrix
- **as.vector** - Changed into a vector
- **as.factor** - Changes number into categorical variable

The data primitives in R are:

- **numeric** - Numeric data (approximations of the real numbers, ℝ)
- **integer** - Integer data (whole numbers, ℤ)
- **factor** - Categorical data (simple classifications, like gender)
- **ordered** - Ordinal data (ordered classifications)
- **character** - Character data (strings)
- **raw** - Binary data

To remove a dataset from RStudio:

remove(studentresult)

To view a Summary table for this dataset:

summary(studentresult)

This returns a **SUMMARY** for every dimension in the dataset, including min, max,median, mean and 1st and 3rd quartile information(for numerical dimensions).

Are there any missing values for any of the variables within the dataframe?

Often analysts will look at the first ten rows and last ten rows to get a better understanding of the data:

head(studentresult)   **tail(studentresult)**

Visualisation

For Visualisation we will use a package called GGPLOT2 developed by Hadley Wickham. To install GGPLOT2 you can do either of the following:

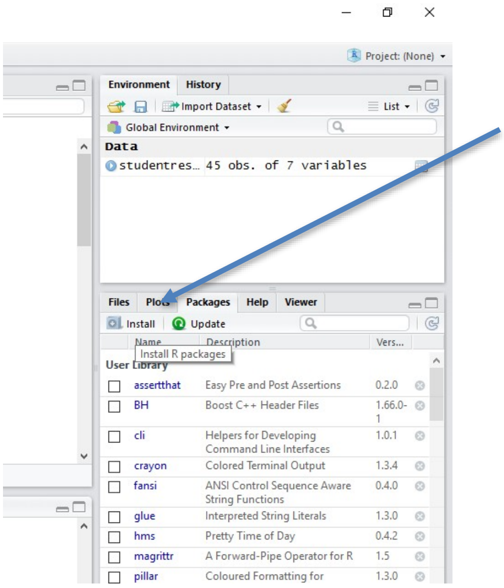
Using the Console

```
install.packages(“ggplot2”)

library(ggplot2)
```

Using RStudio

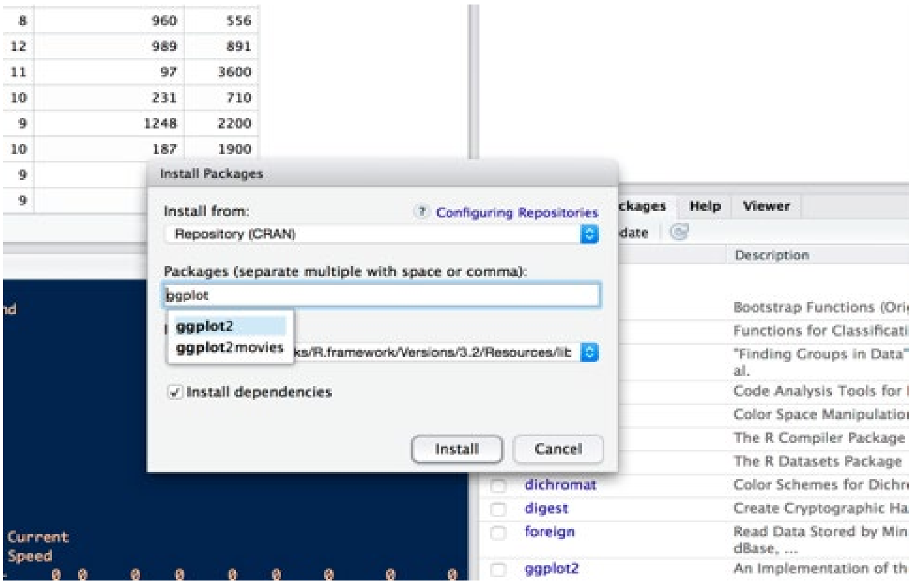
Click on the Install button and type ggplot2:



Data
studentres… 45 obs. of 7 variables

	Name	Description	Vers...
□	assertthat	Easy Pre and Post Assertions	0.2.0
□	BH	Boost C++ Header Files	1.66.0-1
□	cli	Helpers for Developing Command Line Interfaces	1.0.1
□	crayon	Colored Terminal Output	1.3.4
□	fansi	ANSI Control Sequence Aware String Functions	0.4.0
□	glue	Interpreted String Literals	1.3.0
□	hms	Pretty Time of Day	0.4.2
□	magrittr	A Forward-Pipe Operator for R	1.5
□	pillar	Coloured Formatting for	1.3.0

Then click ggplot2 in the packages:



8	960	556
12	989	891
11	97	3600
10	231	710
9	1248	2200
10	187	1900
9		
9		

Install Packages

Install from: Repository (CRAN)

Packages (separate multiple with space or comma): ggplot

ggplot2

ggplot2movies

ks/R.framework/Versions/3.2/Resources/lit

☑ Install dependencies

Once you have ggplot2 installed we can create our first visualisation. For more information on using ggplot2 see: http://www.medstat.nl/images/ handout_ggplot2.pdf.

For this lab we will use a package called sqldf - which stands for sql for data frames. This enables us to use standard SQL functions on a data frame.

```
install.packages("sqldf")

library(sqldf)
```

sqldf allows us to run sql queries across data frames enabling us to do things like group by and sophisticated date queries. One of the most popular queries for data is a group by.

Basic Exercises

Ex1: What are the average results in written exams across all subjects and all years per student?

To run a simple group by query in R and SQLDF, run the following command:

```
markswritten <- sqldf('select Name, avg(Mark_Written) as Written_marks from studentresult group by Name')

View(markswritten)
```

Name	Written_marks
Hercule Poirot	52.06667
Joe O'Neil	72.20000
Mary Healy	88.86667

This shows the average marks in their written exams per student.

Without sqldf, using aggregate:

```
markswritten2<-aggregate(data=studentresult, Mark_Written~ Name,mean)
```

Ex2: Impute missing values

The first thing you should notice with the data is that there are several missing values or *NA* visible for the student Mary Healy’s Oral Marks. Let’s impute those missing values. There are several ways to do this, but one approach is to average her scores over the previous two years and then use that average as her score for the missing year. To do that we:

Version 1 with sqldf

```
avgmark <- sqldf("select AVG(Mark_Oral) from studentresult where Name = 'Mary Healy'
AND Mark_Oral is not ‘NA’")
```

Version 2 without sqldf

```
avgmark2<- mean( studentresult$Mark_Oral [  !is.na( studentresult$Mark_Oral) &
studentresult$Name == 'Mary Healy'  ]  )
```

This calculates the average for Mary Healyy where there are no Nas. Now we want to replace those NA with the average from the query:

```
studentresult$Mark_Oral <- ifelse(is.na(studentresult$Mark_Oral), as.numeric(avgmark),
                                        studentresult$Mark_Oral)
```

Graphing Student Results

ggplot is built on the idea that you build every graph from the same components, a data set, a set of geoms and a coordinate system. Layers of information can be added to customize your visualisation. You may need to install ggplot at this point.

ggplot structure

```
myplot <- ggplot(data= yourdataset, aes(x=yourx, y = youry))    # this begins your plot by adding
the data
```

Examples of extra layers

```
myplot +geom_point()   # this adds a geometry to your plot (scatter plot in the example)

myplot +geom_point(aes(color=dimension)     # geom layer can be customized

 myplot +geom_bar()+scale_fill_brewer(palette=’Reds’)   # customizing the colour palette
myplot +geom_bar(color=dimension) + scale_fill_manual(values=c(‘blue’,’red’))
```

#customize colours manually

```
 myplot+coord_map(projection=”ortho”, orientation= c(41,-74,0))   # map projection
myplot + theme_classic()    # applies a predefined theme to the plot
```

```
 myplot + labs(title=”graph title”, x=”xaxis title”, y=”yaxistitle”) myplot +
facet_wrap(dimension)   # creates small multiples based on dimension
```

See function help and ggplot cheatsheet for extra layer options

Graphing Exercises

Ex1. What are the average results in written exams across all subjects and all years per student?

We already calculated the result of this query earlier. These values are saved in a data frame called markswritten:

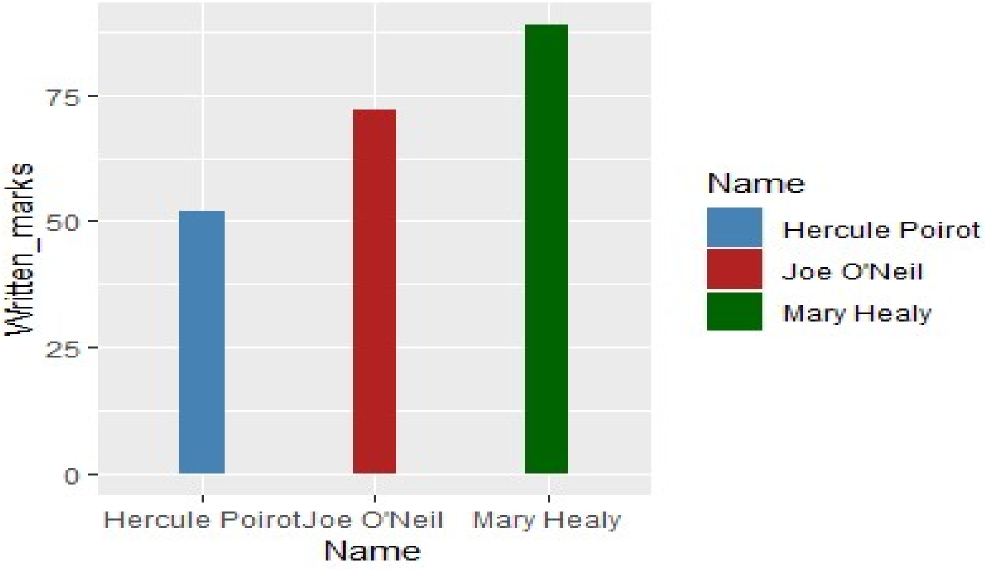
Name	Written_marks
Hercule Poirot	52.06667
Joe O'Neil	72.20000
Mary Healy	88.86667

Ex2: Plot the average marks per student for written submissions.

To plot the data as a bar graph using the data contained in markswriten:

```
namesAvgmarkplot<-ggplot(data=markswritten, aes(x= Name, y=written_marks)) +
geom_bar(stat="identity",aes(fill=Name), width = 0.25)
```

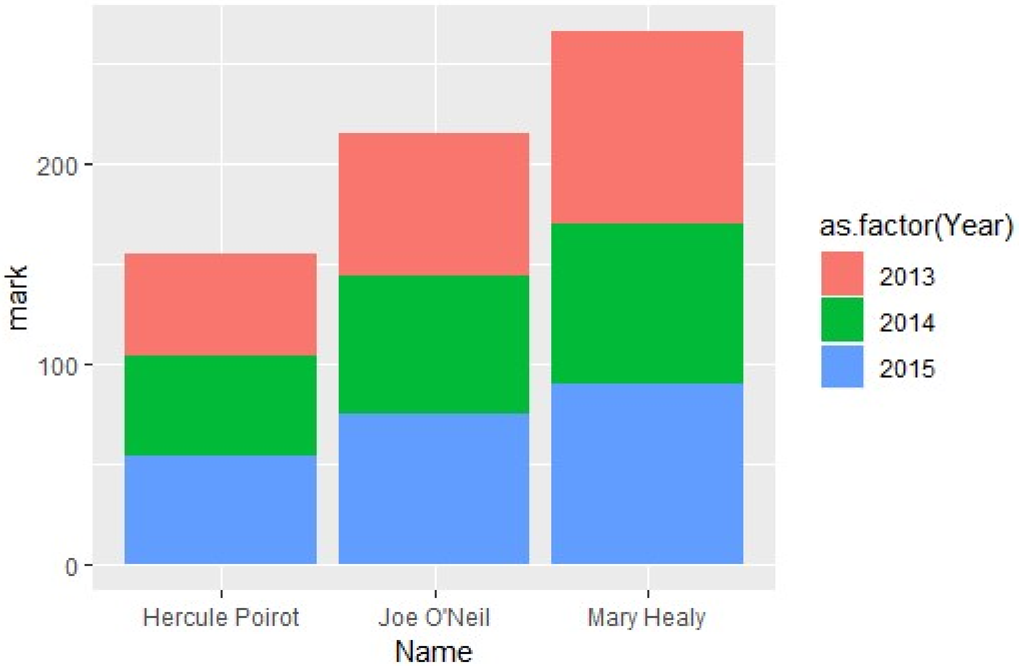
```
namesAvgmarkplot+ scale_fill_manual(values=c("Hercule Poirot"="steelblue", "Joe
O'Neil"="firebrick", "Mary Healy"="darkgreen"))
```



Ex3: What are the average results in the written exams per student and year?

Looking at the result set we see how each person did over each year.

```
mby <- sqldf(“select year, name, sum(Mark_Written)/5 as mark from studentresult group by year,
name”) View(mby) ggplot(data=mby, aes(x= Name, y=mark,fill=as.factor(Year))) +
geom_bar(stat="identity")
```



Creating custom functions

Let’s add a function to calculate the student’s age:

```
getAge <- function(d){              now <-
as.Date(Sys.Date())       then <- as.Date(d,
format="%d-%m-%Y")  result <- now - then
            return(round(as.numeric(result/365)))
}
```

As r is vectorised, you can calculate the age of each student by passing the DOB vector:

```
studentresult$age <- getAge(studentresult$DOB) View(studentresult)
```

References

http://www.cookbook-r.com/Graphs/Bar_and_line_graphs_(ggplot2)/

http://docs.ggplot2.org/current/

Exercises

Please answer the following questions. Use a visualisation to support the answer for each question. Experiment with the different types of charts and options ggplot offers. Please place the visualisations underneath each question.

1. What are the total marks (oral plus written divided by two) for each student for each subject? (2 marks)

2. What is the relationship between age and mark? (2 marks)

3. Did any students do better on their written compared with their oral (or vice versa)? (2 marks)

4. What subject obtained the best results on average? (2 marks)

5. What are the average results in oral exams across all subjects and all years per student? (2 marks)