

Data Visualisation

Lab Recap Week 10

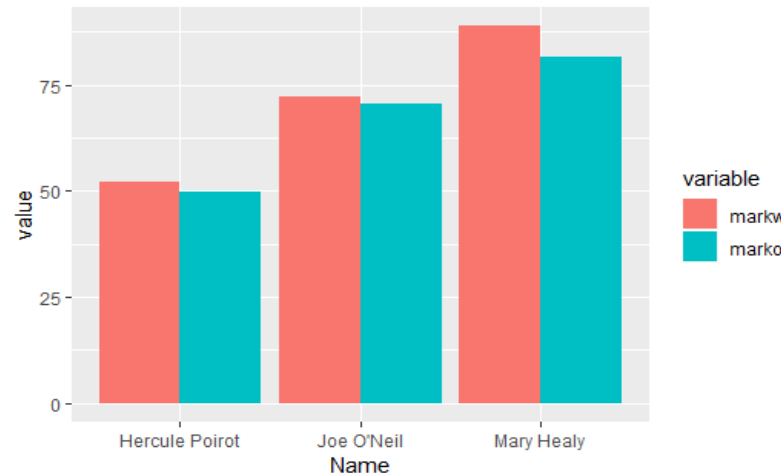
Dr. Cathy Ennis

Using ggplot in R

- Long Tables
- Bar charts
- Histograms
- Scatter plot and Bubble charts
- Line charts
- Smoothing

Long Tables in R

- Looking at last week's studentresults file, we may want to plot the results for the written and oral exams in two bars per student



- In order to plot those results in a bar chart with written and oral in separate bars, we need to convert the table into what is called long table format

Long Tables in R

Original Table

	Name	markw	marko
1	Hercule Poirot	52.06667	49.73333
2	Joe O'Neil	72.20000	70.40000
3	Mary Healy	88.86667	81.40000

Long Table

	Name	variable	value
1	Hercule Poirot	markw	52.06667
2	Joe O'Neil	markw	72.20000
3	Mary Healy	markw	88.86667
4	Hercule Poirot	marko	49.73333
5	Joe O'Neil	marko	70.40000
6	Mary Healy	marko	81.40000

Long Tables in R

- We can create the long table using the function melt from the reshape2 package:

```
library(reshape2)
```

```
resultsW_O2 <- sqldf ( "select Name, avg(Mark_Written) as markw,  
avg(Mark_Oral) as marko from studentresult group by Name")
```

```
head(resultsW_O2)
```

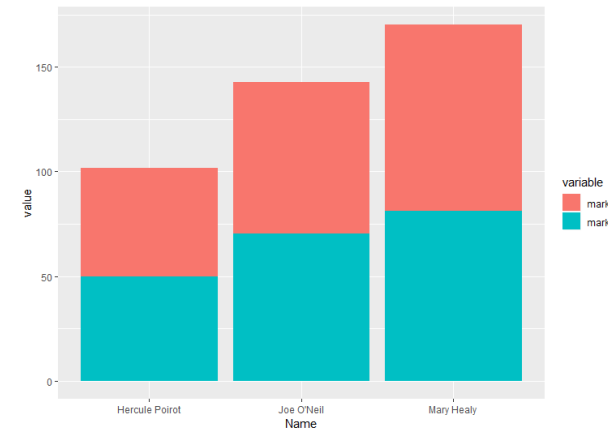
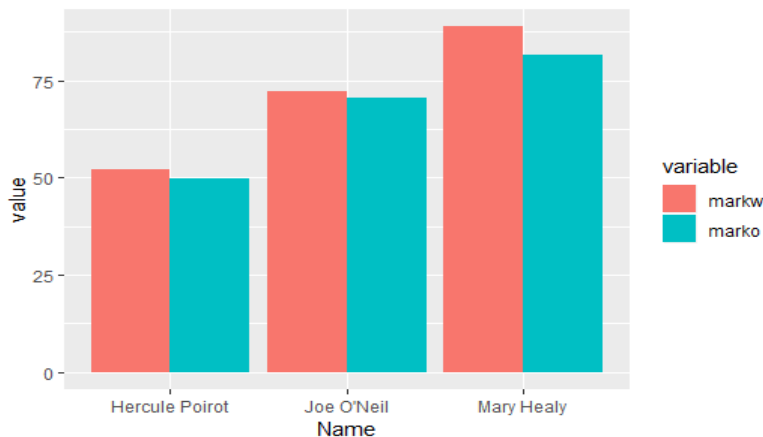
```
resultsW_Om<-melt(resultsW_O2,id.vars = c('Name'))
```

```
head(resultsW_Om)
```

Long Tables in R

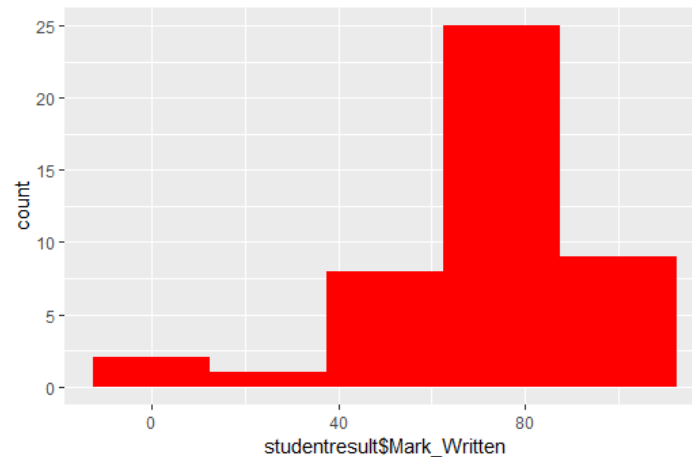
- Now, we can use ggplot to plot the different values, using value for y axis and variable for fill colour, as well as position dodge to place written and oral bars beside each other instead of stacked.

```
ggplot(resultsW_Om, aes(x = Name, y = value, fill = variable)) +  
geom_bar(stat="identity", position = 'dodge')
```



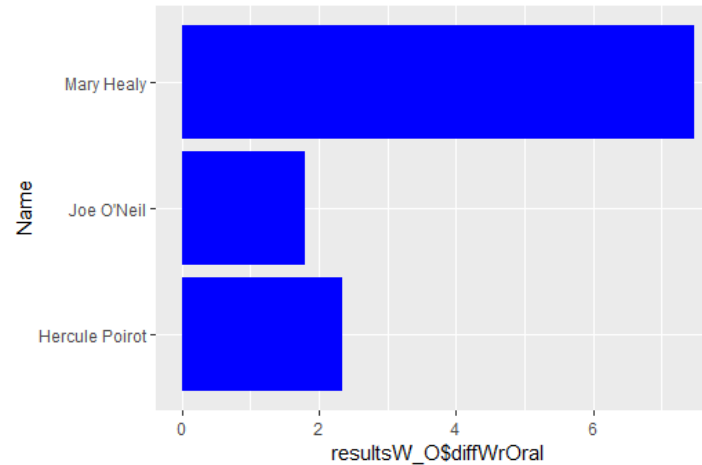
Visualisations using ggplot Histograms

- Plots one variable counting the instances that fall in each bin. Bin size can be adjusted with the command `binwidth`.



```
ggplot(data=studentresult, aes(studentresult$Mark_Written)) +  
geom_histogram(fill='red',binwidth = 25)
```

Visualisations using ggplot Horizontal Bar chart

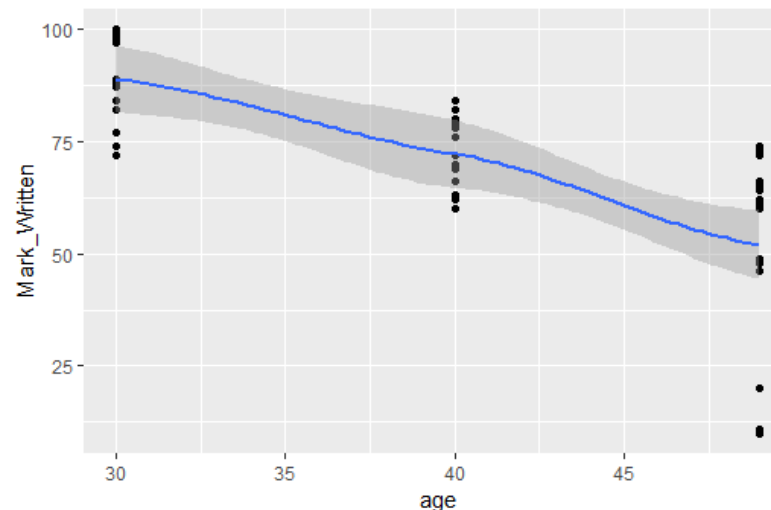


```
ggplot(resultsW_O , aes(x = Name, y = resultsW_O$diffWrOral) ) +  
geom_bar(stat="identity",fill='blue')+coord_flip()
```


Visualisations using ggplot

Fitting lines and smoothing methods

- Lines and curves can be fitted into scatter plots to help in the identification of patterns.
- Options
- method: LOESS, LM
- se: confidence interval display(TRUE,FALSE)



Visualisations using ggplot

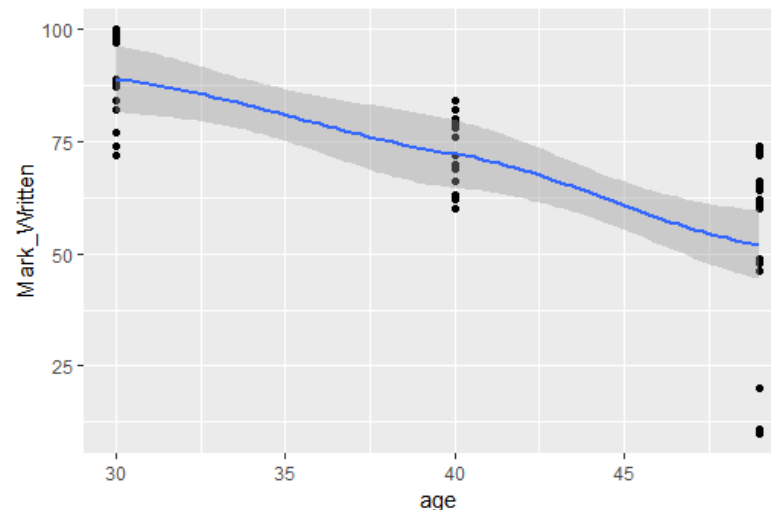
Fitting lines and smoothing methods

- Lines and curves can be fitted into scatter plots to help in the identification of patterns.

```
v<-ggplot(studentresult, aes(x = age, y =Mark_Written ) )
```

```
+geom_point()
```

```
v+geom_smooth(method='loess')
```

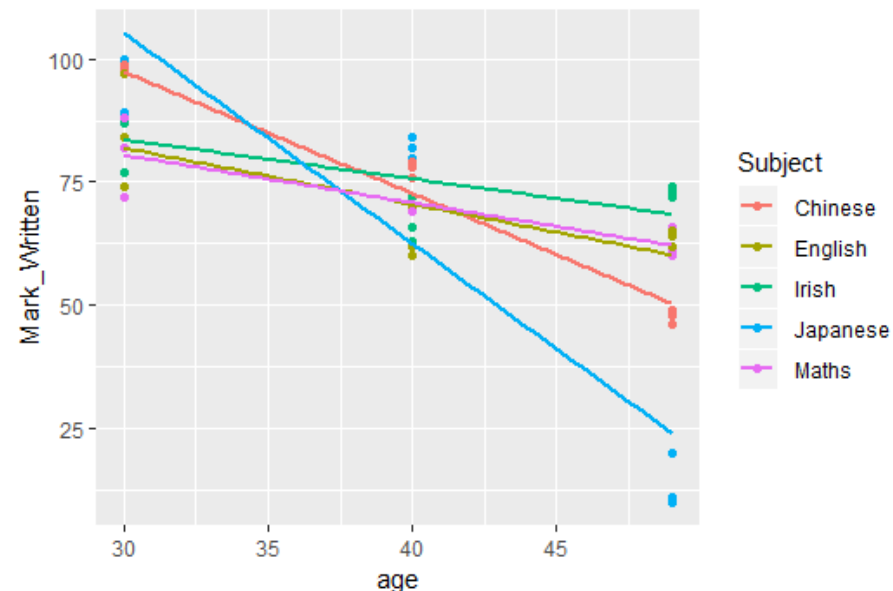


Visualisations using ggplot

Fitting lines and smoothing methods

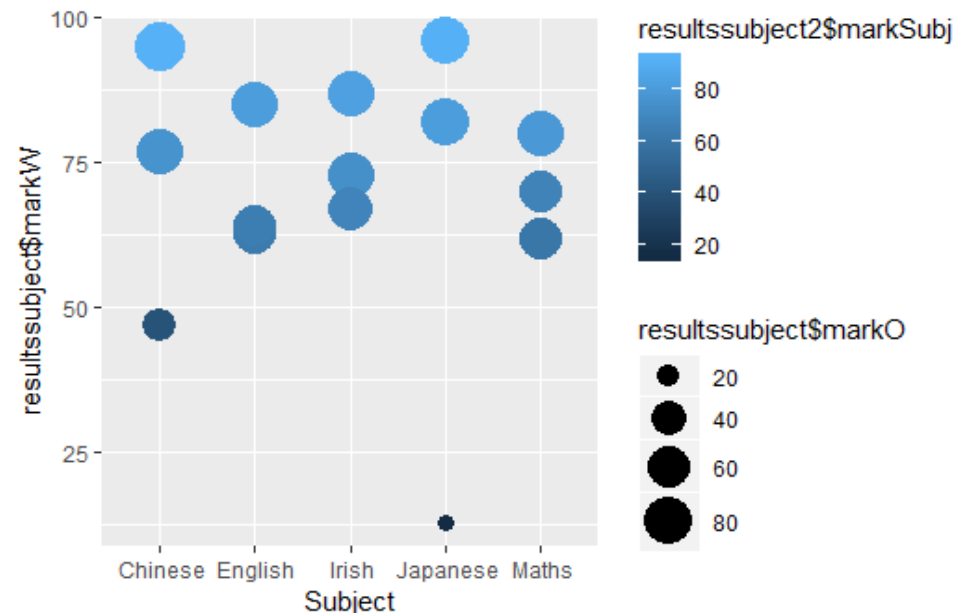
- Fitting separate lines for different data series (per subject)

```
v<-ggplot(studentresult, aes(x = age, y =Mark_Written ,color=Subject) )  
+geom_point()  
v+geom_smooth(method='lm',se=FALSE)
```



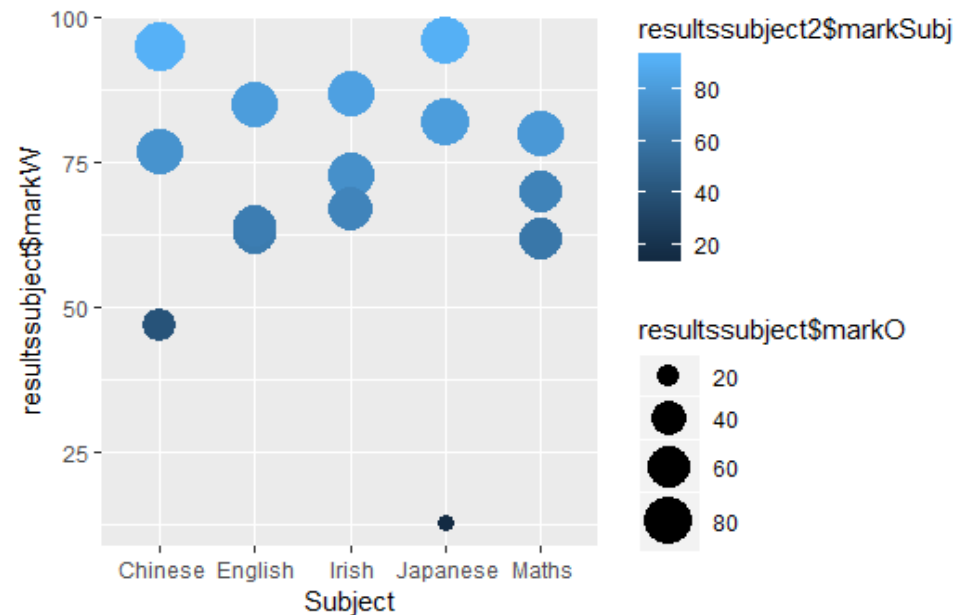
Visualisations using ggplot Bubble Charts

- Bubble charts are created using ggplot by creating a scatter plot and adding the third dimension in the aesthetic option size.
- A fourth dimension can be added using the colour fill.



Visualisations using ggplot Bubble Charts

```
ggplot(resultssubject, aes(x = Subject, y = resultssubject$markW)) +  
  geom_point(aes(size = resultssubject$markO,  
    color=resultssubject2$markSubj)) + scale_size(range = c(3, 9))
```



Question for Lab

- Find one message in our studentresult data
- Create the best visualization to show the message

Assignment 2 – 30%

- You are a Data Scientist. You are tasked with conducting some exploratory analysis. Your goal is to find “insights” in the data and present those findings to your colleagues.
-
1. Select, Clean and Wrangle a Dataset – 4%
 2. Decide on a story (user story) – 2%
 3. Using R, create three visualisations 21%
 4. Show previous iterations or alternatives 3%

Thanks To

- Marisa Llorens-Salvador, John McAuley, Colman McMahon and Brian Mac Namee for an earlier version of these lecture notes