Exploratory Data Analysis

Michael Sinclair

8 October 2018

# Exploratory Data Analysis (EDA)

In today’s session we will be discussing EDA and working through an example using Anscombe’s quartet as a demonstration of why EDA is so important in the initial stages of data analysis.

## What is EDA

In statistics, exploratory data analysis **(EDA)** is an approach to analysing data sets to summarise their main characteristics, often with visual methods.

Exploratory data analysis was promoted by **John Tukey** to encourage statisticians to explore the data, and possibly formulate hypotheses that could lead to new data collection and experiments.

EDA is different from initial data analysis **(IDA)** which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as needed. EDA encompasses IDA.

### “a picture paints a thousand words”

*Frederick R. Barnard in Printer’s Ink (December, 1921)*

# Summary Statistics

Summary statistics help us to gain an understand of the shape of the data by exploring;

* Mean
* Median (middle value)
* Minimum value
* Maximum value
* 1st and 3rd Quartiles

Let’s start having a look at the summary statistics of Anscombe’s quartet.

### First we shall set the working directory to allow us to load in our data

setwd('//dsfin/corpcom/LBBDR/TNG/Session 2')

### The next step is to read in the data from our csv file

df <- read.csv('session2a.csv')

## Displaying summary statistics

If you would like to explore the main summary statistics of your data you can use the *summary( )* command providing your dataframe as the argument.

summary(df)

## X1 Y1 X2 Y2   
## Min. : 4.0 Min. : 4.260 Min. : 4.0 Min. :3.100   
## 1st Qu.: 6.5 1st Qu.: 6.315 1st Qu.: 6.5 1st Qu.:6.695   
## Median : 9.0 Median : 7.580 Median : 9.0 Median :8.140   
## Mean : 9.0 Mean : 7.501 Mean : 9.0 Mean :7.501   
## 3rd Qu.:11.5 3rd Qu.: 8.570 3rd Qu.:11.5 3rd Qu.:8.950   
## Max. :14.0 Max. :10.840 Max. :14.0 Max. :9.260   
## X3 Y3 X4 Y4   
## Min. : 4.0 Min. : 5.39 Min. : 8 Min. : 5.250   
## 1st Qu.: 6.5 1st Qu.: 6.25 1st Qu.: 8 1st Qu.: 6.170   
## Median : 9.0 Median : 7.11 Median : 8 Median : 7.040   
## Mean : 9.0 Mean : 7.50 Mean : 9 Mean : 7.501   
## 3rd Qu.:11.5 3rd Qu.: 7.98 3rd Qu.: 8 3rd Qu.: 8.190   
## Max. :14.0 Max. :12.74 Max. :19 Max. :12.500

The output from this command is the summary statistics for each *column* within your dataframe.

# Linear Model

A linear model is trying to predict that value of y given x, written as **y ~ x**. The results of the linear model are the y intercept, where the line crosses the y axis, and the slope of the line.

We can create a linear model using the *lm( )* function. We will give the model the equation of *y ~ x* i.e. pass the two variables that we will then go on to display on our x and y axis. Remember to present in the order y and then x.

We will save the linear model to a variable called model which will allow us to get additional data from it later.

#Linear models are to be written as y predicted by x  
model <- lm(Y2 ~ X2, data = df)  
model

##   
## Call:  
## lm(formula = Y2 ~ X2, data = df)  
##   
## Coefficients:  
## (Intercept) X2   
## 3.001 0.500

In our example the line crosses the y axis at 3.00 and the slope of the line is 0.500. This gives us the equation of the line as y = 3.00 + 0.500x. You will have used this in Excel when you have looked at a linear (straight) line of best fit for your data.

We can also extract the R-squared value, which is a statistical measure of how close the data are to the fitted regression line. This can be accessed using the r.squared part of the model summary as below.

The R-squared value shows us how well the linear model fits to the data with 1 being a perfect fit and 0 indicating that the model does not.

summary(model)$r.squared

## [1] 0.666242

# Correlation

The correlation between two variables indicates how one changes when the other one does. In other words when x increase if the correlation is positive then y increased, if the correlation is negative then y decreases and if the correlation is 0 then y remains constant.

There is a *cor( )* function in R which allows us to find the correlation between two variables.

cor(df$X1, df$Y1)

## [1] 0.8164205

#### Remember that correlation does not show causation

Remember that a high correlation does not mean that one variable causes the other.

An good example of this is around July each year the number of **cold drinks** sold increases at a similar rate to the sale of **ice-cream**. Based on this data we could assume that the sale of cold drinks is *causing* the sale of ice-creams, however there is another vaiable that we are not considering that is causing them both. That variable is **heat**, when the weather gets nicer (hotter) people buy more **ice-creams and cold drinks**.

For other examples visit [Tyler Vigen’s Spurious Correaltions Web Page](http://www.tylervigen.com/spurious-correlations).

## What can we say about Anscombe’s Quartet?

From the summary statistics we can say that each of the four sets of data have

* Mean of x = 0
* Mean of y = 7.50 (to 2 decimal places)
* Linear Regression line ( y = 3.00 + 0.500x )
* Correlation between x and y = 0.816 (to 3 decimal places)

# Libraries

In today’s session we shall be extending the functionality of R by introducing the *grid* library.

Libraries allow us to expand the functionality or commands available in R to carry out additional tasks.

## Install new libraries

Libraries can be installed using *code* or using the packages menu on in the bottom right panel of the screen.

The package we are going to use today will already be on your machines as R comes with this pre-installed.

## Load a library

We will use *code* to load libraries in order to ensure that your code runs everytime you use it and that you do not forget to load the correct libraries for your analysis.

library(grid)

# EDA

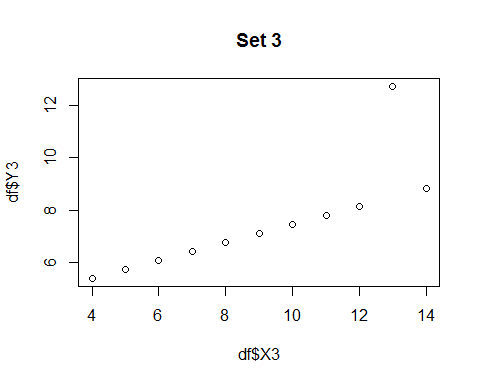
### …make both calculations and graphs. Both sorts of output should be studied; each will contribute to understanding.

F. J. Anscombe, 1973 (and echoed in nearly all talks about data visualization…)

The base plot function in R will produce a scatter plot of our data. The following is a plot of one of the sets of data (set 3) in Anscombe’s quartet.

### Scatter plot

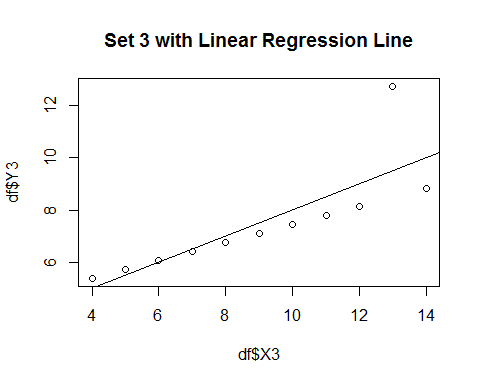
plot(df$X3, df$Y3, main='Set 3')



### Adding linear model

We can then add our linear model to the above plot. We can do this by creating a linear model and passing to the *abline( )* function which requires the arguments of where the line crosses the y axis and the slope of the line, which are the outputs of out liner model.

#create the scatter plot  
plot(df$X3, df$Y3, main="Set 3 with Linear Regression Line")  
  
#add the linear model line to the plot  
abline(model)



## Tuning the parameters of an R base plot

We will be touching on more advanced graphical techniques in future sessions. For now we will look at how we can change the symbology of a base scatter plot to allow the data to stand out more.

**arguments**

* col -> sets the **outline colour** of the points
* bg -> sets the **fill colour** of the points
* cex -> sets the **size** of the points, scaled relative to default (1). 0.5 is 50% smaller and 1.5 is 50% larger
* pch -> sets the **symbol** of the points, more points are available [here](http://www.sthda.com/sthda/RDoc/figure/graphs/r-plot-pch-symbols-points-in-r.png)

Additional help is available via the Statmethods site [here](https://www.statmethods.net/advgraphs/parameters.html)

We will also be using the *par( )* function to allow all four charts to be displayed together.

# Anscombe’s Quartet

#--- add linear model -----  
par(mfrow=c(2,2), oma=c(0,0,1,0), cex.main = 2)  
  
plot(df$X1, df$Y1, main="Set 1", col='black', bg = 'orange', xlim = c(4,20), ylim = c(2,14), cex=3, pch=21)  
abline(model, col='red')  
  
plot(df$X2, df$Y2, main="Set 2", col='black', bg = 'orange', xlim = c(4,20), ylim = c(2,14), cex=3, pch=21)  
abline(model, col='red')  
  
plot(df$X3, df$Y3, main="Set 3", col='black', bg = 'orange', xlim = c(4,20), ylim = c(2,14), cex=3, pch=21)  
abline(model, col='red')  
  
plot(df$X4, df$Y4, main="Set 4", col='black', bg = 'orange', xlim = c(4,20), ylim = c(2,14), cex=3, pch=21)  
abline(model, col='red')  
  
title("Anscombe's Quartet", outer = TRUE)

