**NAME**

**COLLEGE NUMBER**

**Abstract**

The application of data analyses systems span across nearly almost all industries today. Data science is the gold of the future, individuals and organisation are taking the advantages derived from the use and exploitation of the principle of the data science. Finance, Healthcare and the education industries are currently investing heavily as they tap into the full benefits and capabilities of the data science industry. Engineers with a technical background have also gone ahead to develop software tools and programs that are focused on making the whole analysis process faster by developing algorithmic tools and software programs that automatically drive statistical inferences at the click of a button.

Further, functions like sales and marketing, health sectors and the financial departments are now employing the use of big data to help predict the next trend or point that their forecast might land in. Examples here include future diseases discoveries and explorations, the next direction that the stock market might take and whether or not to invest on the particular shares and bonds, given past and current market prevailing factors. To achieve a given level of expected output and prediction, the choice has to be made on the correct dataset, the model and the classification method to apply.

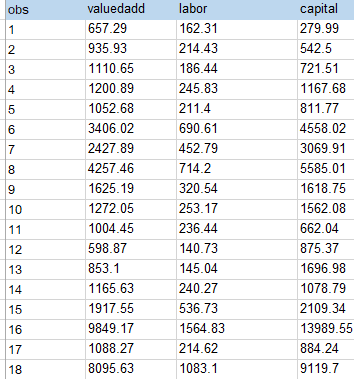
**Introduction**

One final objective of the data engineer/data scientist is to finally be able to produce analysis that speaks to anyone. The data analysis process should contain a clear scenario of where the objectives of the analysis is coming from, the problem it’s trying to solve and the future of such analysis. Managers and top management consistently rely on the analysis of these data to make useful and meaningful derivations. One way to achieving this is by using R programming language which is a statistical tool as well as data visualisation pack.

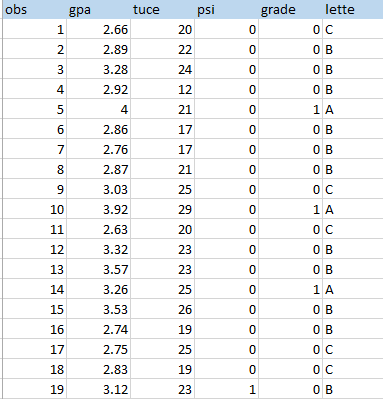
The R studio environment gives the user the full potential to grab the analytical capabilities of this tool. R as a programming language on the hand has benne used for by individual and correlations over the years to deliver business needs and meet statistics analysis that area meaningful. Additional, another significance of the language is that it’s highly supported, easy to learn and can be has a number of packages that support quick analysis and visualisations.

**Dataset description**

There are two datasets provided for this particular project, all the datasets are delivered in common separated files. A quick seep in to be the first dataset that is a production dataset reveals the following contents:



Whereas the snip into the second dataset reveals the following:



In order to begin the analysis, we shall load the two datasets unto o the Rstudio IDE, beginning with the first set:

##

> library(readr)

Warning message:

package ‘readr’ was built under R version 4.1.1

> production <- read\_csv("class/hello/papers/7th/Econometric decisions/production.csv")

Rows: 27 Columns: 4

-- Column specification --------------------------------------------------------------------------

Delimiter: ","

dbl (4): obs, valuedadd, labor, capital

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show\_col\_types = FALSE` to quiet this message.

> View(production)

> > str(production)

spec\_tbl\_df [27 x 4] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)

 $ obs      : num [1:27] 1 2 3 4 5 6 7 8 9 10 ...

 $ valuedadd: num [1:27] 657 936 1111 1201 1053 ...

 $ labor    : num [1:27] 162 214 186 246 211 ...

 $ capital  : num [1:27] 280 542 722 1168 812 ...

 - attr(\*, "spec")=

  .. cols(

  ..   obs = col\_double(),

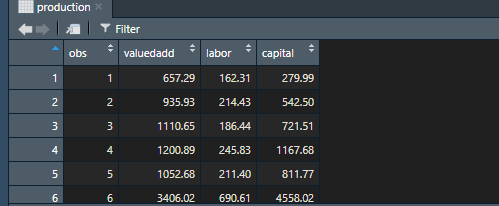
  ..   valuedadd = col\_double(),

  ..   labor = col\_double(),

  ..   capital = col\_double()

  .. )

 - attr(\*, "problems")=<externalptr>



1. Cobb-Douglas production function

The Cobb-Douglas production function usually is used in economics to evaluate and estimate the how production is affected by other factors of production such as laor, price and the demand.

library(micEcon)

> productionOutput <- production$valuedadd

> QuantityInputs <-production$capital

> productionTime <- c(1:20)

estResult <-  translogEst( "valueadd", c( "labor","capital", "productionTime"),

                           production, linear = TRUE )

In R, we can formulate the Cobb Douglas function by calling in the library Micecon that has a number of built in packages that support economic analysis. In this case the first step is to identify the variables that need to be used. We know very well the factors that would affect production and as such, the output in this case is the dependent variable that change based on the independent variables of labour, time and capital. So we first initialise the depends variable which in this stance is the $valueadd, then we initialise two more variables that is $captial and $labor, but in order to get a smooth estimation, we need to do this over some period of time, so we add the last variables which is estResult that shall consume the translogEst built in micEcon package to give us the full estimate of these factors

Further, we can test for the multi -linearity of this dataset using the linearity matrix in R. For that, a built function called the corrplot () has a number of in-built functions to help do this.

 ##DETERMINING MULTICOLINEARITY

 > library(corrplot)

corrplot 0.92 loaded

Warning message:

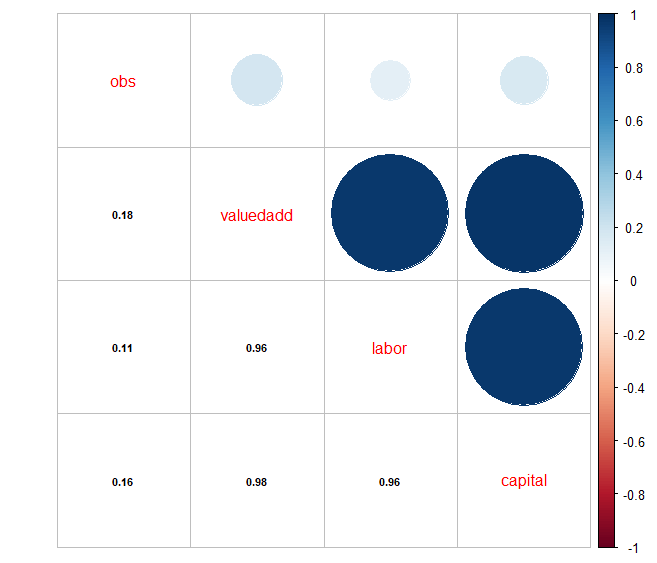
package ‘corrplot’ was built under R version 4.1.2

> cor1 = cor(production)

> corrplot.mixed(cor1, lower.col = “black”, number.cex = .7)

> corrplot.mixed(cor1, lower.col ="black", number.cex = .7)

>



The above illustration shows that there is a high correlation between the value addition and the labour inputs, also the mufti linear relationships indicate a high relationship between the capital and the value addition of the business. The above cases can be illustrated to mean that output is directly proportional to the labour and the capital invested.

**Model estimation and correlation**

##GETTING CORRELATION COFFICIENTS

> library(tidy verse)

> cor(production$valuedadd, production$capital)

[1] 0.9753576

> library(ggplot2)

> ggplot(production) +

+     aes(x = capital, y = valueadd) +

+     geom\_point(colour = "red") +

+     theme\_minimal()

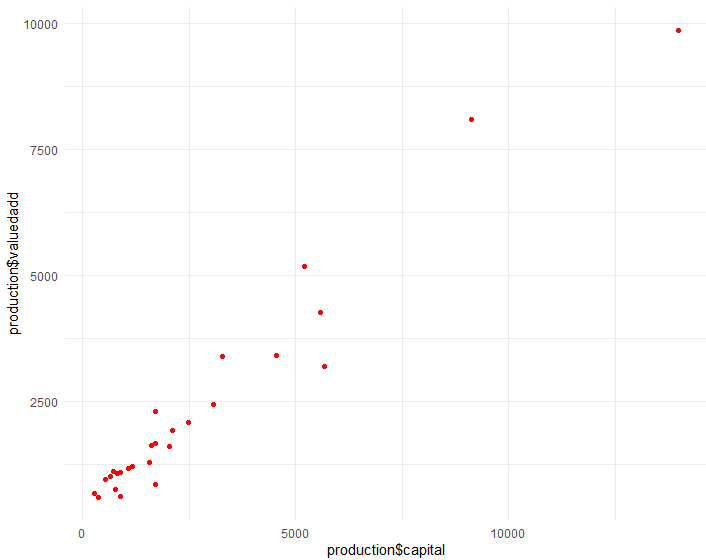
> ggplot(production) +

+     aes(x = production$capital, y = production$valuedadd) +

+     geom\_point(colour = "red") +

+     theme\_minimal()

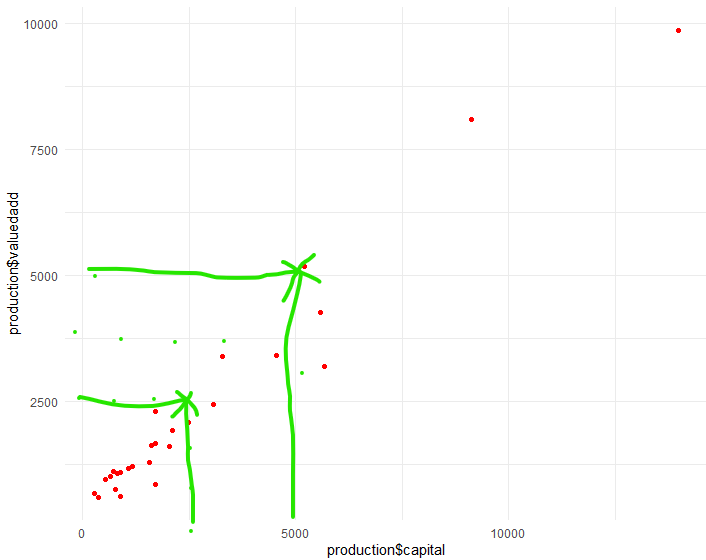
>

****

In the above we have done a coefficient correlation of the value addition against the capital invested and the coefficient value returned is 0.9, this is very strong, showing that there is a big positive relationship between the two observed variables

**Constant returns to scale**

The theory of constant returns in scale indicates that as the input variables in the factors of production like labour and capital increase, so does the value addition, in practice we can say that, if for instance we double the capital then the output will also double since these two are closely related. In the case above, if we take the initial value of 2500 income at a value of 2500 and decide to double it, we shall get the value of returns shall also double as per the below:



**Question two**

**Data inspection**

A quick observation of the data is done by loading it on the R environment and inspecting its structure

##GRADE DATA

> View(grade)

> str(grade)

spec\_tbl\_df [32 x 6] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)

 $ obs  : num [1:32] 1 2 3 4 5 6 7 8 9 10 ...

 $ gpa  : num [1:32] 2.66 2.89 3.28 2.92 4 2.86 2.76 2.87 3.03 3.92 ...

 $ tuce : num [1:32] 20 22 24 12 21 17 17 21 25 29 ...

 $ psi  : num [1:32] 0 0 0 0 0 0 0 0 0 0 ...

 $ grade: num [1:32] 0 0 0 0 1 0 0 0 0 1 ...

 $ lette: chr [1:32] "C" "B" "B" "B" ...

 - attr(\*, "spec")=

  .. cols(

  ..   obs = col\_double(),

  ..   gpa = col\_double(),

  ..   tuce = col\_double(),

  ..   psi = col\_double(),

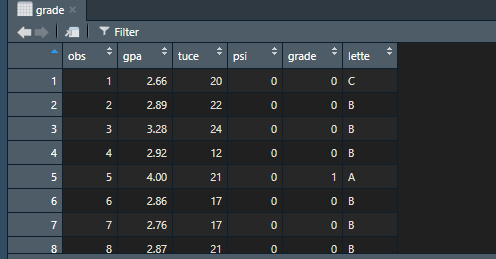
  ..   grade = col\_double(),

  ..   lette = col\_character()

  .. )

 - attr(\*, "problems")=<externalptr>

>



1. **Logistic regression model**

#CREATE TRAINIGN AND TEST DATA

> set1 <- grade$gpa

> set2 <- grade$tuce

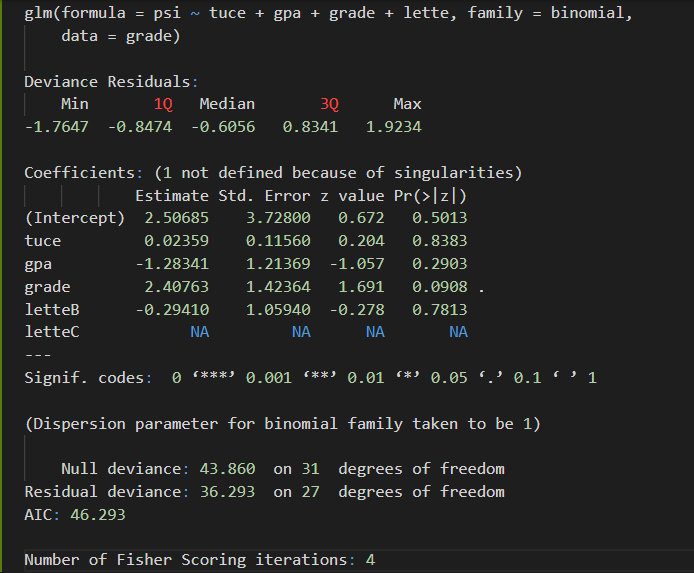
> set.seed(1000)

> logitmodel <- glm(set1,set2)

#MODEL

> glm.fits=glm(psi∼ tuce + gpa + grade + lette, data=grade,family =binomial)

> summary(glm.fits)



#GETTING THE COEFFICIENT FITS

> coef(glm.fits)

(Intercept)        tuce         gpa       grade      letteB      letteC

 2.50684627  0.02359214 -1.28340785  2.40762555 -0.29409538          NA

>

**And then the summary**

> summary(glm.fits)$coef

               Estimate Std. Error    z value   Pr(>|z|)

(Intercept)  2.50684627   3.728001  0.6724371 0.50130550

tuce         0.02359214   0.115601  0.2040825 0.83828902

gpa         -1.28340785   1.213690 -1.0574431 0.29030942

grade        2.40762555   1.423644  1.6911705 0.09080424

letteB      -0.29409538   1.059397 -0.2776064 0.78131455

>

c. **Getting the coefficient correlation**

COEFFICIENTS

> library(tidyverse)

> cor(grade$gpa, grade$grade)

[1] 0.4971474

>

ggplot(grade) + aes(x = grade$tuce, y = grade$gpa) + geom\_point(colour = "red") +theme\_minimal()

From the dataset, the PCI and the grade variables are considered binary whereas the unary variables in this case are obs and tuce

#PREDICTING GRADE

> fit\_1 <- lm(gpa ~ lette, data = gpa\_a)

> summary(fit\_1)

Call:

lm(formula = gpa ~ lette, data = gpa\_a)

Residuals:

     Min       1Q   Median       3Q      Max

-1.04273 -0.21250  0.02313  0.25156  0.56727

Coefficients:

            Estimate Std. Error t value Pr(>|t|)

(Intercept)   3.4327     0.1191  28.827  < 2e-16 \*\*\*

letteB       -0.3527     0.1618  -2.180 0.037519 \*

letteC       -0.6890     0.1835  -3.754 0.000776 \*\*\*

---

Signif. codes:  0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.395 on 29 degrees of freedom

Multiple R-squared:  0.3301,    Adjusted R-squared:  0.2839

F-statistic: 7.144 on 2 and 29 DF,  p-value: 0.003002

 ggplot(data=gpa\_a, aes(fit\_1$residuals)) +

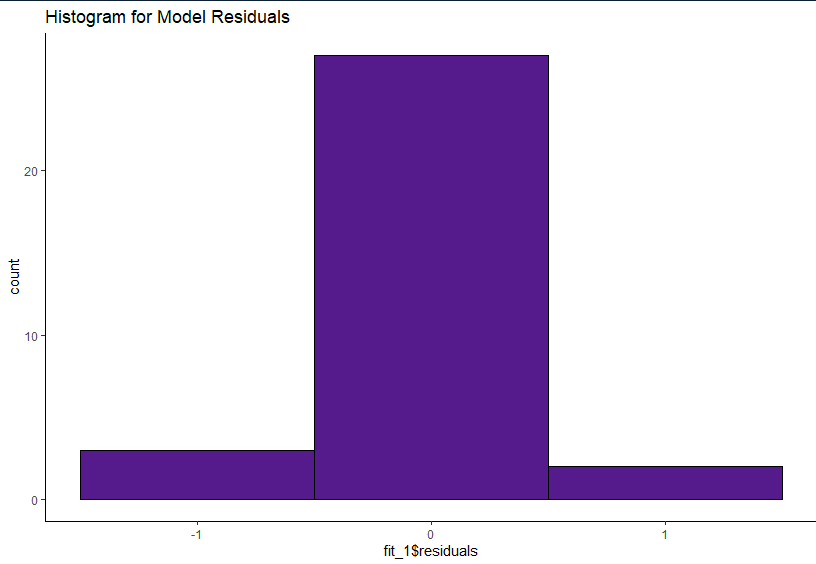
+     geom\_histogram(binwidth = 1, color = "black", fill = "purple4") +

+     theme(panel.background = element\_rect(fill = "white"),

+           axis.line.x=element\_line(),

+           axis.line.y=element\_line()) +

+     ggtitle("Histogram for Model Residuals")



#PREDICT GPA

predict(fit\_1, data.frame(gpa = 3.5))