WESTERN SYDNEY
UNIVERSITY

# Computing, Engineering & Mathematics

## ASSIGNMENT / REPORT COVER SHEET

**This sheet must be attached to all material being submitted for marking.**

| | | | |
|---|---|---|---|
| **Student name: Student number:** | Bhuban Pun<br>22105743 | **Student name:<br>Student number:** | |
| **Sections completed individually** | All Section | **Sections completed individually** | |
| **Unit name & number:** | Data Science<br>COMP 7006 | | |
| **Tutorial day and time:** | Friday, 11 AM to 1 PM | | |
| **Title of Assignment:** | Data Analysis Task | | |
| **Student Submitting the Assignment:** | Bhuban Pun | | |
| **Date submitted:** | 13th October 2023 | | |

### Student Declaration (must be signed)

**Declaration:**

- ❑ I hold a copy of this assignment if the original is lost or damaged.
- ❑ I hereby certify that no part of this assignment or product has been copied from any other student's work or from any other source except where due acknowledgement is made in the assignment.
- ❑ No part of the assignment/product has been written / produced for me by any other person except where collaboration has been authorized by the subject lecturer/tutor concerned.
- ❑ I am aware that this work may be reproduced and submitted to plagiarism detection software programs for the purpose of detecting possible plagiarism ***(which may retain a copy on its database for future plagiarism checking)*)**
- ❑ I hereby certify that no part of this assignment or product has been submitted by me in another (previous or current) assessment, except where appropriately referenced, and with prior permission from the Lecturer/Tutor/ Unit Co-Ordinator for this unit.

| | |
|---|---|
| **Student signature and date:** | Bhuban Pun<br>13th October 2023 |
| **Student signature and date** | |

**Note: An examiner or lecturer/tutor has the right to not mark this assignment if the above declaration has not been signed.**

# Data Analysis Task on Factors affecting Stock Market

Bhuban Pun

Computer, Data & Math Sciences, Western Sydney University
Sydney, Australia
study@westernsydney.edu.au

**Data analysis task on factors affecting stock market is a project guided with a purpose of finding economic factors which affect the movement of the stock market, basically the Index value. In the project, we used different machine learning techniques to develop a model which fits the best with our data set and helps to predict the stock market based on the economic variables. As our project is based on predicting the Australian Stock Exchange, historical economic data has been gathered from sources like Australian Stock Exchange and Reserve Bank of Australia. Mainly, linear regression model and decision tree method is used to build the model where decision tree model seems to perform better with less error. The stock market is affected by various factors, among which economic and financial factors have a larger impact in the long run. The most important part is the significance of the project. We can use the model we have designed to predict the stock market but to gain more significance in the model, more statistical methods can be used to increase the accuracy.**

## I. Introduction

In the modern world of finance and investment data analysis plays a vital role in decision making. Data is considered as one of the most important assets in this field. Traditional data analysis techniques might not be relevant in today’s world. To change over time, we should focus on new ways of analyzing data which would result in better efficiency and ultimately maximum returns on our investment.

Data Science has made possible a lot of things easier which we couldn’t have even imagined few decades ago. It is an interdisciplinary field, which fits in almost every sector or industries. The main reason is that there is history or statistics in every sector. We use those statistics or records to make decisions for the future performance of the specific entity.

In finance, there are different kinds of investment opportunities with securities being the most popular option around the world. Securities might be stock, bond, debenture, commodities, Mutual Funds etc. which are traded on the stock exchange. The stock market is the second largest financial market in the world after the foreign exchange market (Forex). Most of the country has their own stock market, which measures the performance of the listed companies. Australia has few stock exchanges among which Australian Stock Exchange (ASX) is the main exchange which have 2310 companies listed in it. All Ordinaries in ASX is a index which measures the price movement of the largest 500 companies. It has the ownership of 87% of the Australia’s security market (September 2023). Australian securities market is categorized into 11 sectors, 23 Industry Groups, 69 Industries and 158 Sub-Industries. Among 11 sectors, Materials sectors has the largest number of listed companies with 816, followed by financials sector with 509 companies. The utilities sector has the smallest number of listings with 21. Other sectors in ASX are Health Care, Energy, Consumer Discretionary, Consumer Staples, Communication Services, Real Estate, Industrials, and Information Technology.

There are many factors which affect the stock market, which might range from political factors to economic factors and financial factors, and from social factors to technological factors. Affected by almost every aspect, it becomes a challenge to take investment decisions which will create maximum returns. In this project I have collected data representing economic and financial aspects. The main objective of the project is to find the most relevant factors which affect the stock market.

To perform this project historical data are collected from Reserve Bank of Australia [1], Australian Stock Exchange [2], Yahoo Finance [3], Australian Bureau of Statistics [4], and Nepal Rastra Bank [5]. The collected data represents the economic condition of the country and performance of Australian stock market ASX. I have observed economic and financial factors have a major impact on the stock market. There are economic factors which have a greater or no impact in the market. This project is fully focused on finding the major economic factors affecting the stock market trend of Australia. In this project, we have used Rstudio to compute the statistical operation.

## II. Data Description Explore and Pre-Processing

### A. *Data set*

Data set a list of observations representing different variables in it. For the completion of this project data set has been developed by collecting data from different sources like, Reserve Bank of Australia [1], Australian Stock Exchange [2], Yahoo Finance [3], Australian Bureau of Statistics [4], and Nepal Rastra Bank [5]. To initiate this project, initially the main concept or statement of problem was developed, and the required raw data were collected to give direction to the project. Basically, data set contains historical performance of Australian economy, Australian stock exchange, interest rate of Australian Banks and inflation of Nepal. Monthly data has been collected from January 2003 to July 2023. However, as per the project's limitation of using only structured data. Date variables have been removed from the data set and updated to make it more structured. All the observations and variables are stored in a data set name "economyau".

The initial phase of data analysis task is to gather raw data from relevant sources and proceed for cleaning. The main reason for using data from multiple sources is to avoid variables with missing data and to gather correct data from valid sources. Data cleaning is a time-consuming process and requires the use of different tools. For this project tools like data scrapper, spreadsheets (Ms. Excel), are used and stored all the required data in a csv file "economy.au.csv".

### *B. Data and statistics*

The data set contains a total of 247 observations and 7 variables. All the listed variables in a data are numeric variables meaning they are quantitative in nature and no categorical variables are present. Variables of the data set "economyau" are as follows:

- Interest rate: It is the percentage of interest rates provided by Australian Banks on term deposit.
- GDP growth: It represents the economic growth of Australia represented in percentage change.
- Inflation: It the increment of consumer price index in percentage basis.
- Volume: It is the number of shares traded in Australian Stock Exchange which is represented in unit terms.
- FX rate: This variable indicates foreign exchange rate between Australian dollar in terms of US dollars.
- In: In is a variable representing inflation rate of Nepal. The values are presented in percentage terms.
- Index: This variable represents index values of All Ordinaries Index of Australian Stock Exchange.

In a data set, observations with variables GDP growth, Inflation and FX rate (foreign exchange rate) indicates major economic indicator and Interest rate as financial indicator of Australian economy. Volume and Index variable indicates movement of Australian Stock Exchange. In variable includes inflation of Nepal. The reason behind choosing this indicator is to examine the relation between Australian economy and its Equity Exchange with a developing economy Nepal. Detailed observations and variables about a data set can be extracted with a code available in appendix A1.

Every variable in a data set has a different value with different ranges. Among the 7 variables in a data set, Volume variable seems to have larger values than other variables followed by Index, remaining variables has mean value below 10. Statistics of the sample variables are as follows:

- Interest rate: Values of interest rate ranges from minimum 0.25 to maximum 0. 8.25 with an average value of 3.566.
- GDP growth: Its values range from -5.75 to 10.315 with a mean value of 2.728.
- Inflation: Its values range from -0.3 to 7.80 with a mean value of 2.651.
- Volume: Its values range from 2381000000 to 41760000000 with a mean value of 19010000000.
- FX rate: Its values range from 0.5884 to 1.0954 with a mean value of 0.8013.
- In: Its values range from -4.519 to 24.583 with a mean value of 9.971.
- Index: Its values range from 2778 to 7823 with a mean value of 5368.

Similarly, we can view the relationship between the variables with statistical methods like covariance and correlation. As values in a variable have different scale, we measure correlation among the variables in a data set to check the linear relationship. Upon evaluating correlation in a data set, we can observe that Index variable has positive correlation with 4 variables and negative correlation with 2 variables. Index variable has weak positive correlation with Volume, Inflation, In and GDP growth variables. The index has medium negative correlation with Interest rate and weak negative correlation with FX rate. Likewise, Interest rate and FX rate tends to have comparatively strong correlation. Detail summary statistics of variables can be obtained from a code in appendix A2.

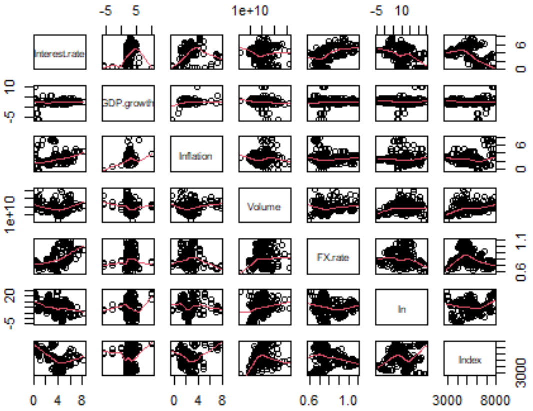


Figure A2.1: Correlation plot

## III. Aims and objectives

The aims and objectives of a project is a main driving factor of the project which gives direction to the project. The main aim and objectives of the project is to find answer to the below mentioned questions:

- What are the major economic factors affecting Australian Stock market?
- Which of the variables have maximum contribution in model building?
- Which of the machine learning model best fits to predict the Australian Stock Market?

The main object of the project is to identify the major factors affecting the Australian stock market and develop a machine learning model which can predict the market efficiently.

## IV. SUpervised learning

Machine learning is an important part of Data Science, which helps to build models bas on our data set nature. Machine learning is basically categorized into three parts, Supervised Learning, Unsupervised Learning and Reinforcement Learning.

Supervised Learning is a process of building models when target variables is known or identified. Supervised Learning includes regression models and classification models. For this project our data set economyau target variable is Index. We will be focusing on building supervised learning model which fits the best to predict the index values with less error.

### A. *Method 1*

This is the most crucial step part of the project. This part focus on building model which best fits our data set. In this phase we examine to find the best linear model from our variables. We have 7 variables in our data set, and by evaluating correlation coefficients we observed some of the variables have positive correlations and some have negative.

Since we have multiple variables and observations it becomes more efficient to use K-Fold cross validation approach to find the best linear model. K-Fold cross validation is a popular machine learning approach to identify linear model. K-Fold cross validation is a popular cross validation technique. It basically divided the data into k equal sized groups randomly in the first step. Then it builds the model on k-1 of the groups, leaving 1 group. Finally, it predicts the model with 1 group left out and repeats until all k groups have been left out. For k fold cross validation only k models are built on approx. (k -1 / k) * n data points. The main benefit of using this technique is it is computationally faster, and it can be less expensive as compared to Leave One Out Cross Validation (LOOCV) method.

In the first phase of this Method 1, we need to load a boot package from library in rstudio. Now, we divide our data set "economyau" into 2 parts. Data set is divided into the ratio of 70:30, meaning 70% of data set has be divided into training data set and remaining data set has been allocated for validation and testing purpose. Data set is divided using random sampling method (Appendix B1.1). Training data set is for building the model using different types of supervised learning models. Validation data set is separated to validate the model to choose the best model after predicting the variables and computing the model accuracy. The model with the lowest error rate is taken for the project.

After the completion of the first phase, we perform k fold cross validation, in this project we perform 10-k cross validation method to develop a linear regression model. We use the inbuild glm function in R Script scripter of Rstudio. We use the function to find best linear model where target variable is Index and other remaining 6 variables are predictors. We have built the model with target variable with respect to other variables. In total we have build 21 linear models based on our variables of training data set "train_economyau". All the linear models can be seen on appendix B1.2

Now, we perform 10-fold cross validation to test each model and to find the error of the models. An object "kfolderrors" has been assigned to store errors of all the 21 models. This method can be computed using the inbuild function cv.glm. The performed cross validation approach can be computed using the code in appendix B1.3 and a plot of the models and kfolderrors can be viewed with the help of code B1.4.

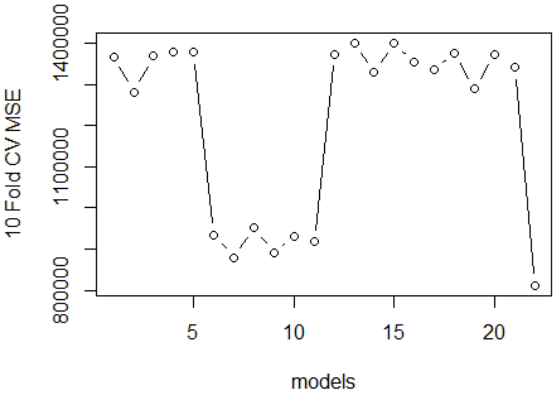


Fig. B1.4: MSE of linear models

The above plot is a MSE of all the models, models are plotted on the x-axis and MSE value on the y-axis. As per the 10-fold cross validation approach we can observed that the model 7 "m7" has the least error. Among 21 models we select m7 as the best linear fit for our data set. Model 7is a multiple linear regression model. Equation of model 7 is given by, Index = 6259.96 – 403.33 * Interest rate + 229.59 * Inflation (i)

Here, the Index is a target variable and coefficients of our equation are 6259.96 as intercept, –403.33 as slope of Interest rate and 229.59 as the slope of Inflation.

After we build a model and choose the best model, our next step is to predict the Index with the help of validation data set “valid_economyau”. We use the built predict function to predict the model. So, in this part we predict our target variable using our best model “m7” by using predictor variables available in validation data set. The model for prediction can be found in appendix B1.5.

It is always important to check the performance of the model. This is a post analysis of the model. There are different approaches to check the model accuracy. For a regression problem we can use Residual Sum of Squares (RSS) or Mean Square Error (MSE). Since, our target variable is a numeric type, it’s better to choose Mean Square Error (MSE) method to choose the error of the model. MSE is simply a mean of squared difference between actual value and predicted value. Error value can be higher or lower based on the values of target variables. In our model MSE is 966586.4 which is given by a code in appendix B1.6. Standard or unit values of error can be computed by putting square root on mean square error. This helps to interpret the error in a better way. In our model square root of MSE is 983.1513 which indicates the error of the model m7 while predicting our target variable Index. Error can be more useful when performing comparative analysis.

We can also assess the model by checking whether the assumptions of the linear model are fulfilled or not. We plot the selected model “m7” to examine the model’s assumptions. Though six plots are available, by default rstudio provides four plots. Before we move forward we want to be sure about three of the major assumptions, Linearity, Normality of errors and Consistency of variance (Homoscedasticity).

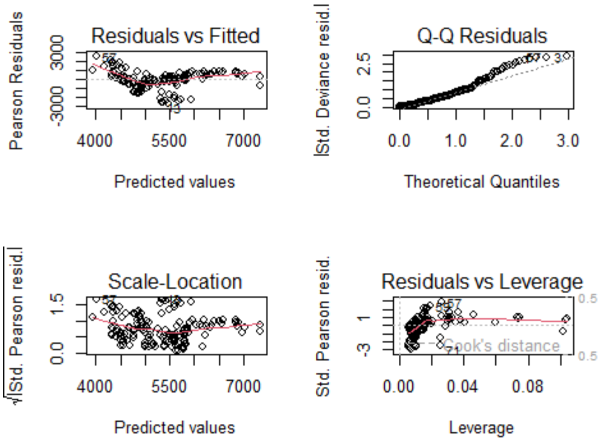


Fig. B1.7 Model 7 plot

- If the model is linear the first plot Residual vs Fitted plot would look like a clear sky with no formation of patterns. If any patterns are observed, assumption is said to be violated. In the figure, assumption of linearity has been violated.
- If the model aligns with normality assumption, the plot 2 Q-Q residual plot should be straight meaning all the values should fit in a straight line (have normal distribution). In the figure, assumption of normality has been violated.
- Plot 3 Scale Location helps to identify the homoscedasticity, meaning the graph is good for detecting non-constancy of variance. To meet the assumption, values should fit around the horizontal line with no formation of patterns. Here in the figure, the assumption of homoscedasticity is violated.
- The plot 4 Residuals vs Leverage helps to identify any possible patterns in the standardized residuals as a function of the leverage. It also helps to identify Cook’s distance.

When assumptions of the model are violated, accuracy about the prediction might be affected.

### *B. Method 2*

There are various approaches of machine learning to identify the best model to make inference about the data. Using a single model to fit the model might bring challenges in prediction and might even result poor efficiency. To identify the model which fits the best we have used a linear regression model. We can also compute models using other statistical techniques. In this Method 2 we will be using classification method to build the model for our data set. We will be using decision tree method to identify the better model than previous model.

Basically, we use regression models like linear regression and polynomial approach to create a model and we use classification models like Decision tree, Support Vector Machines, Logistic regression for classification problems. Decision tree is useful for both regression and classification type of problems which use visualization approach to define the model. It is one of the most popular models in classification method. Decision tree or tree-based method simply involves segmenting the predictor variable into a few simple regions. This method is easy to explain, and most people can easily understand. It has three parts, root node, branches or decision nodes and terminal nodes or leaf nodes. The main upper variable is called root node, nodes splitting after root node and node above terminal nodes are called branches or internal nodes. The last nodes are called terminal nodes. The main objective of this model is to capture the most important variables and reduce the noise.

For the computation of a decision tree model, we use the same training data set and validation data set. Firstly, we need to load a tree package from library in rstudio. Once package has been loaded, we perform calculation step using tree function with our training data set "train_economy". We then evaluate a tree model using summary function available in appendix C1.1. The output of the regression tree indicates that there are 10 terminal nodes. Detailed output of the decision tree model can be seen from the figure below and code can be found in appendix C1.2.

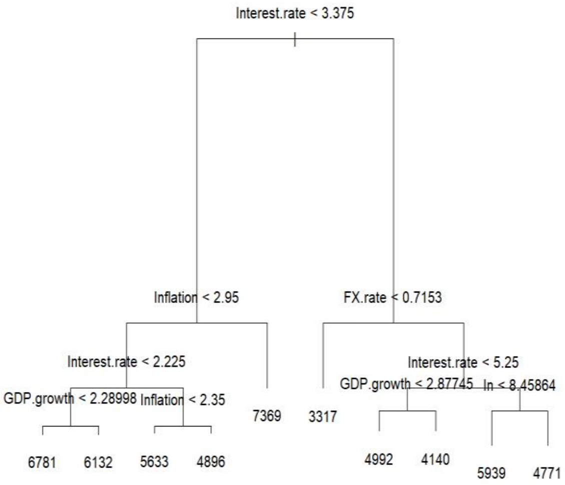


Fig. C1.2: Decision Tree

After the formation of the model now time is to check how the model is performing. We can do this by calculating a Mean Square Error (MSE) after predicting our target variable with help of our validation data set. After calculating MSE we see that Mean Square Error is 625685, and square root of MSE is 791.0025. Detail about how the model is performing can be found using the code in appendix C1.3.

We have completed model building using tree regression, but we can perform pruning to find the better model with less error. So, in this part we prune the previous regression tree model using cross validation method. Pruning in decision tree model is basically cutting down the branches which are not significant to the model. This function is performed when trees are too big & complex, which might lead to overfitting and ultimately bad prediction. A model with smaller branches and nodes results lower variance and better interpretation. So, the main strategy of the model is to design large tree and then perform pruning to fit best variables for the model. We perform pruning using cross validation to get the best model. After pruning, we plot the result choosing size and dev vector. The result of pruning can be observed using code in appendix C1.4.

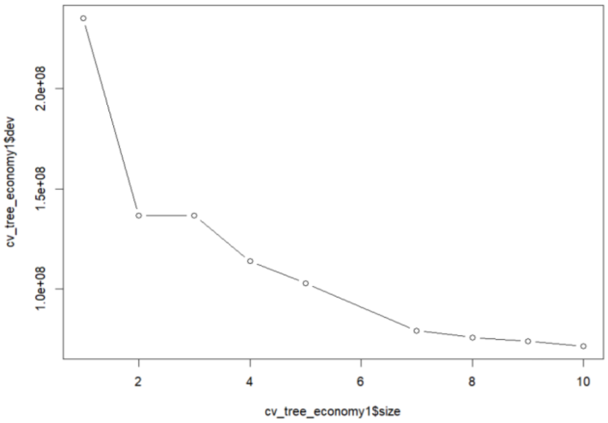


Fig. C1.4: Cross validation pruning

From the above plot we can see that from pruning using cross validation approach, the best model is given by the size 10, which is the number of terminal nodes in a tree. This indicates that there is no improvement in a model. The result of tree after pruning is shown in the figure below.

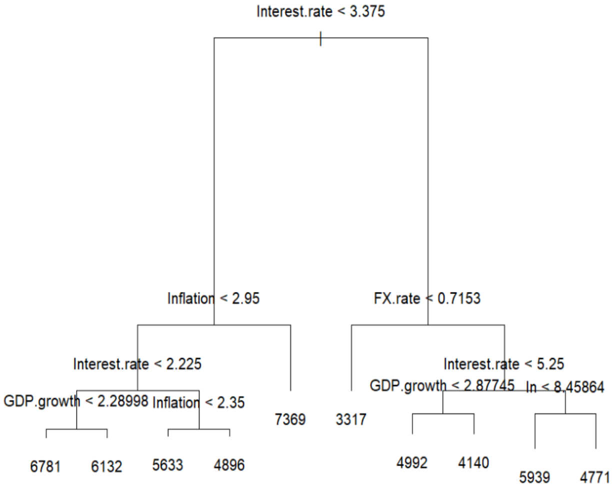


Fig. C1.5: Decision Tree after pruning

To check the accuracy of the model, we calculate Mean Square Error (MSE) after predicting the model using predictors of validation data sets. After calculating MSE we see that Mean Square Error is 625685, and square root of MSE is 791.0025. Details about how the model is performing can be found using the code in appendix C1.6.

From the figure above, it can be interpreted that,

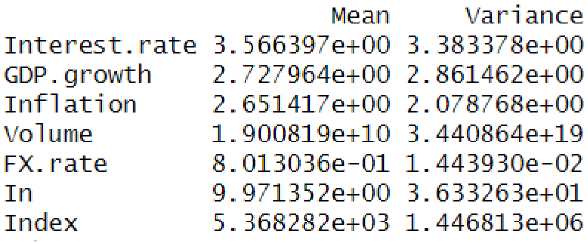
- Interest rate is the most important predictor in determining the Index. Inflation, FX rate and GDP growth is also significant in determining Index.

- Index value is maximum when Interest rate is below 2.225, Inflation is below 2.95 and GDP growth is below 2.28998.
- Approximately, equal Index (to the maximum) can be obtained when Interest rate is greater than 3.375 and FX rate is smaller than 0.7153.

## V. UNSUPERVISED LEARNING

Unsupervised learning is a machine learning technique used in data science. Data set is considered as unsupervised when it doesn't have target variable. In this learning, variables in a data set tries to find relationship and patterns on its own. Some of the most popular unsupervised learning techniques are, Dimensionality Reduction method, Clustering method etc. For this project, we use the Dimensionality Reduction method where we use Principal Component Analysis (PCA) to reduce dimensions of the variable. Most data sets are of higher dimensions like large number of observations and variables. We use PCA method to find a small number of summary variables that still capture the nature of the data. The main aim is to find a low dimensional representation of data that defines as much information as possible. We use PCA technique in this project for dimension reduction. It identifies a new variable that is a linear combination of the original variable, and which has maximum variance.

For the computation of this model, we first explore the data set including some statistical summary like mean and variance. To perform PCA method, we need to have numeric variables in our data set. Details about the data set including mean and variance is given by code in appendix D1.1.



```
                       Mean     Variance
Interest.rate 3.566397e+00 3.383378e+00
GDP.growth    2.727964e+00 2.861462e+00
Inflation     2.651417e+00 2.078768e+00
Volume        1.900819e+10 3.440864e+19
FX.rate       8.013036e-01 1.443930e-02
In            9.971352e+00 3.633263e+01
Index         5.368282e+03 1.446813e+06
```

Table D1.1: Mean and variance

From the output we can see that variables in data set have different mean and variance values. Volume has the largest mean value which is followed by Index. Similarly, FX rate has the smallest mean value. Likewise, Volume has the largest variance followed by Index and In. FX rate, Inflation, Interest rate and GDP growth variance is comparative smaller. Given the variability in variance, we need to scale the data set while performing PCA function. We perform PCA function in R using prcomp function which is mentioned in appendix D1.2

From the output we can print out "sdev", "rotation", "center", "scale" and "x". The "rotation" matrix of the output provides the principal component loadings. Each column contains the corresponding principal component loading vectors. The "x" values are the result of the principal component models. Similarly, "sdev" gives the standard deviation of principal components, "center" gives the mean value of the original variables and "scale" gives the standard deviation of the original variables. The first two principal components using loading parameters is given by,

PC1= -0.633535058 * Interest rate - 0.158182448 * GDP growth - 0.005297316 * Inflation + 0.138203609 * Volume - 0.445472350 * FX rate + 0.352450268 * In + 0.481473300 * Index (ii)

PC2= 0.13832877 * Interest rate + 0.57241391 * GDP growth + 0.71420762 * Inflation + 0.21557725 * Volume + 0.04690872 * FX rate + 0.08937961 * In + 0.29402764 * Index (iii)

We can also compute the proportion of variance explained by the principal components using a summary function. The first principal component explains about 30% of the variation in the data, the second PC explains about 20% of the variation in the data, the next principal component explains about 18% of the variation in the data set, and so forth. Similarly, the first and second PCs together explain about 50% of the variation in the data set. The first, second, third, fourth and fifth principal components explain about 92% of the variation in the, and so on. The summary of the cumulative proportion is shown in the plot below. Summary of the cumulative proportion and plot can be computed using the chunk code in appendix D1.3.

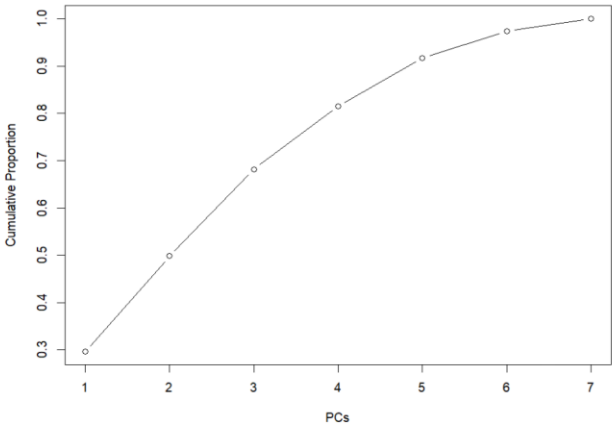


Fig. D1.3: PCs cumulative variance proportion

For better understanding we biplot to out principal component model. We use biplot function to make visualization with scale argument. The scale=0 argument to biplot function ensures that the arrows are scaled to represent the loadings; other values for scale give slightly different biplots with different interpretations.

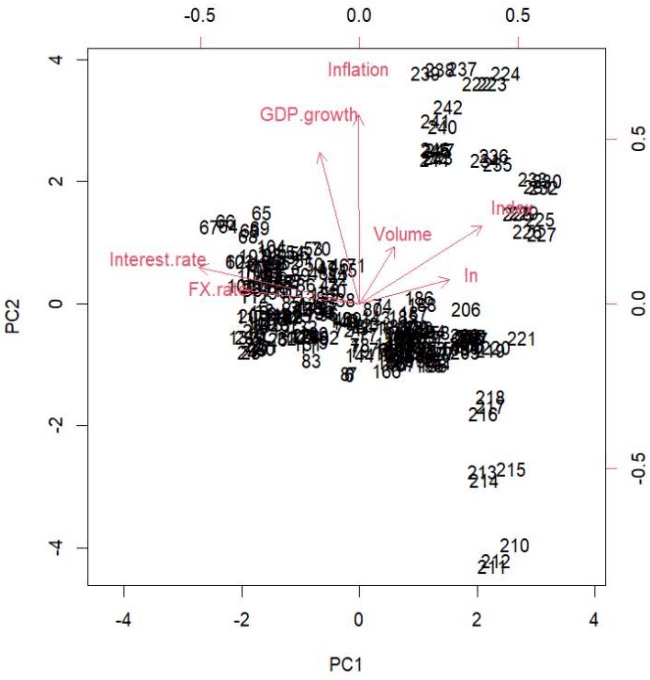


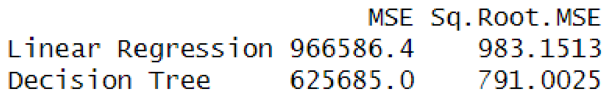
Fig. C1.4: Biplot

From the above plot we can see that Index, Volume and In has high correlation, Inflation and GDP growth has high correlation, and Interest rate and FX rate have high correlation. We can also see that Index, Volume and In has high contribution to PC1 where other variables have lower contribution. Similarly, Inflation and GDP growth has higher contribution to PC2.

## VI. Model comparison

Based on the nature of data we can employ various machine learning techniques to design and develop the best model which results better efficiency and higher accuracy. We performed various statistical operations in this project. Model building was an important part of the project where we used supervised learning and unsupervised learning to build machine learning algorithms. Basically, we used linear regression method and decision tree method from supervised learnings and dimensionality reduction (Principal Component Analysis) method from unsupervised learning.

In supervised learning model, among six predictor variables, multiple linear regression model has taken 2 predictor variables, Interest rate and Inflation. While in a decision tree model Interest rate is considered as the most important predictor in determining the Index which is followed by Inflation, GDP growth and FX rate. Similarly, after the model was developed, we performed prediction with the list of our validation data set. It’s always vital to test the accuracy of the model. We have used Means Squared Error (MSE) of both the models to choose the best fitting model.



```
                        MSE  Sq.Root.MSE
Linear Regression 966586.4     983.1513
Decision Tree     625685.0     791.0025
```

Table E1.1 Model Comparison

While comparing two models from supervised learnings. We computed multiple linear regression model with Mean Square Error of 966586.4 and square root of MSE with 983.1513. Similarly, decision tree model is computed with Mean Square Error of 625685 and square root of MSE with 791.0025. From the error values presented in the above table, the decision tree model has less error as compared to that of linear regression model. Hence, the best model which fits our data set in this project is a decision tree model.

## VII. RESULT AND RECOMMENDATIONS

This is a conclusion part of the project. Data exploration, cleaning, model building and testing are the most important part of any data science project. We have successfully performed all these operations and different results have been generated. Some of the important results of the project are as follows:

- Both multiple linear regression and decision tree model can be used to design and build the model for our data set.
- Decision tree model seems to perform better with less accuracy as compared to decision tree model.
- Out of six predictor variables, multiple linear regression model defines 2 predictor variables, Interest rate and inflation as significant. While in a decision tree model Interest rate, Inflation, GDP growth and FX rate are considered important.
- Interest rate, Inflation and GDP growth seems to be a major economic factor influencing Australian stock market.

Some of the important recommendations are as follows:

- There are various ways to fit model based on the nature of data. In this project we have used limited number of statistical methods to build the model.

- Time series data are not considered for this project which limits the efficiency of the machine learning model.

## REFERENCES

[1] Reserve Bank of Australia (2019). Statistics. [online] Reserve Bank of Australia. Available at: https://www.rba.gov.au/statistics/.

[2] ASX (2016). Home - Australian Securities Exchange - ASX. [online] Asx.com.au. Available at: https://www.asx.com.au/.

[3] Yahoo Finance (2023). Business, Investments, Stocks & Quotes - Yahoo Finance. [online] Yahoo Finance. Available at: https://au.finance.yahoo.com/.

[4] Australian Bureau of Statistics (2021). Australian Bureau of Statistics, Australian Government. [online] Abs.gov.au. Available at: https://www.abs.gov.au/.

[5] Nrb.org.np. (2012). Nepal Rastra Bank - Central Bank of Nepal. [online] Available at: https://www.nrb.org.np/.

## APPENDIX

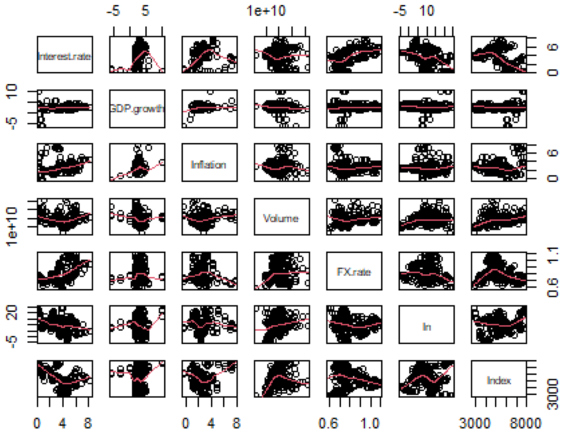
A1: Description

```
economyau <- read.csv("economy.au.csv")
str(economyau)

## 'data.frame':    247 obs. of  7 variables:
##  $ Interest.rate: num  3.95 3.85 3.85 3.85 3.85 3.55 3.6 3.75
3.8 4 ...
##  $ GDP.growth   : num  3.64 3.64 3.06 3.06 3.06 ...
##  $ Inflation    : num  2.9 2.9 3.3 3.3 3.3 2.6 2.6 2.6 2.6 2.6 ...
##  $ Volume       : num  2.38e+09 1.08e+10 1.08e+10
9.07e+09 1.16e+10 ...
##  $ FX.rate      : num  0.588 0.605 0.604 0.623 0.652 ...
##  $ In           : num  23.4 23.6 24.6 20.7 19.9 ...
##  $ Index        : num  2778 2849 2849 2971 2980 ...

head(economyau)

##   Interest.rate GDP.growth Inflation      Volume FX.rate
In  Index
## 1          3.95   3.639716       2.9  2380650400  0.5884
23.36957 2778.4
## 2          3.85   3.639716       2.9 10761888900  0.6054
23.55556 2848.6
## 3          3.85   3.059119       3.3 10761888900  0.6036
24.58297 2848.6
## 4          3.85   3.059119       3.3  9067411400  0.6230
20.65124 2970.9
## 5          3.85   3.059119       3.3 11602052000  0.6522
19.89619 2979.8
## 6          3.55   1.739292       2.6 12136302200  0.6674
16.02399 2998.9
```

A2: Data set

```
summary(economyau)

##  Interest.rate     GDP.growth       Inflation         Volume
##  Min.   :0.250   Min.   :-5.750   Min.   :-0.300   Min.
:2.381e+09
##  1st Qu.:2.250   1st Qu.: 2.198   1st Qu.: 1.800   1st
Qu.:1.521e+10
##  Median :3.550   Median : 2.639   Median : 2.400   Median
:1.824e+10
##  Mean   :3.566   Mean   : 2.728   Mean   : 2.651   Mean
:1.901e+10
##  3rd Qu.:4.850   3rd Qu.: 3.182   3rd Qu.: 3.050   3rd
Qu.:2.183e+10
##  Max.   :8.250   Max.   :10.315   Max.   : 7.800   Max.
:4.176e+10
##     FX.rate            In              Index
##  Min.   :0.5884   Min.   :-4.519   Min.   :2778
##  1st Qu.:0.7147   1st Qu.: 6.844   1st Qu.:4504
##  Median :0.7644   Median :10.088   Median :5353
##  Mean   :0.8013   Mean   : 9.971   Mean   :5368
##  3rd Qu.:0.8904   3rd Qu.:13.203   3rd Qu.:6128
##  Max.   :1.0954   Max.   :24.583   Max.   :7823

cov(economyau)

##               Interest.rate    GDP.growth     Inflation        Volume
## Interest.rate  3.383378e+00  6.175224e-01  4.683202e-01 -
5.861804e+08
## GDP.growth     6.175224e-01  2.861462e+00  8.037509e-01
-1.728985e+09
## Inflation      4.683202e-01  8.037509e-01  2.078768e+00
1.163720e+09
## Volume        -5.861804e+08 -1.728985e+09  1.163720e+09
3.440864e+19
## FX.rate        1.218646e-01  1.733789e-02 -1.975283e-02
1.536942e+08
## In            -4.009445e+00  4.483538e-01 -1.591866e-01
7.143229e+09
## Index         -1.239067e+03  1.538702e+01  3.419962e+02
1.664326e+12
##                     FX.rate            In         Index
## Interest.rate  1.218646e-01 -4.009445e+00 -1.239067e+03
## GDP.growth     1.733789e-02  4.483538e-01  1.538702e+01
## Inflation     -1.975283e-02 -1.591866e-01  3.419962e+02
## Volume         1.536942e+08  7.143229e+09  1.664326e+12
## FX.rate        1.443930e-02 -1.089419e-01 -2.847922e+01
## In            -1.089419e-01  3.633263e+01  9.418721e+02
## Index         -2.847922e+01  9.418721e+02  1.446813e+06
```

```
cor(economyau)

##              Interest.rate   GDP.growth   Inflation       Volume
FX.rate
## Interest.rate    1.00000000  0.198464771  0.17658936 -
0.05432781  0.55135234
## GDP.growth       0.19846477  1.000000000  0.32955237 -
0.17424621  0.08529608
## Inflation        0.17658936  0.329552374  1.00000000
0.13759782 -0.11401269
## Volume          -0.05432781 -0.174246210  0.13759782
1.00000000  0.21804719
## FX.rate          0.55135234  0.085296077 -0.11401269
0.21804719  1.00000000
## In              -0.36162673  0.043972224 -0.01831703
0.20202838 -0.15040887
## Index           -0.56003262  0.007562306  0.19720233
0.23588394 -0.19703745
##                        In        Index
## Interest.rate -0.36162673 -0.560032623
## GDP.growth     0.04397222  0.007562306
## Inflation     -0.01831703  0.197202334
## Volume         0.20202838  0.235883940
## FX.rate       -0.15040887 -0.197037451
## In             1.00000000  0.129908419
## Index          0.12990842  1.000000000

pairs(economyau, panel = panel.smooth)
```



B1.1 Sampling and Data allocation

```
set.seed(77)
tr_economy <-
sample(1:nrow(economyau),round(nrow(economyau)*0.70,
0))
train_economyau <- economyau[tr_economy, ]
valid_economyau <- economyau[-tr_economy, ]
actual_economyau <- valid_economyau$Index
```

B1.2 Linear Models using glm function and k-fold for cross validation method

```
library(boot)
m1 <- glm(Index~In, data = train_economyau)
m2 <- glm(Index~FX.rate, data = train_economyau)
m3 <- glm(Index~Volume, data = train_economyau)
m4 <- glm(Index~Inflation, data = train_economyau)
m5 <- glm(Index~GDP.growth, data = train_economyau)
m6 <- glm(Index~Interest.rate, data = train_economyau)
m7 <- glm(Index~Interest.rate+Inflation, data =
train_economyau)
m8 <- glm(Index~Interest.rate+GDP.growth, data =
train_economyau)
m9 <- glm(Index~Interest.rate+Volume, data =
train_economyau)
m10 <- glm(Index~Interest.rate+FX.rate, data =
train_economyau)
m11 <- glm(Index~Interest.rate+In, data = train_economyau)
m12 <- glm(Index~GDP.growth+Inflation, data =
train_economyau)
m13 <- glm(Index~GDP.growth+Volume, data =
train_economyau)
m14 <- glm(Index~GDP.growth+FX.rate, data =
train_economyau)
m15 <- glm(Index~GDP.growth+In, data = train_economyau)
m16 <- glm(Index~Inflation+Volume, data =
train_economyau)
m17 <- glm(Index~Inflation+FX.rate, data = train_economyau)
m18 <- glm(Index~Inflation+In, data = train_economyau)
m19 <- glm(Index~Volume+FX.rate, data = train_economyau)
m20 <- glm(Index~Volume+In, data = train_economyau)
m21 <- glm(Index~FX.rate+In, data = train_economyau)
m22 <- glm(Index~., data = train_economyau)
```

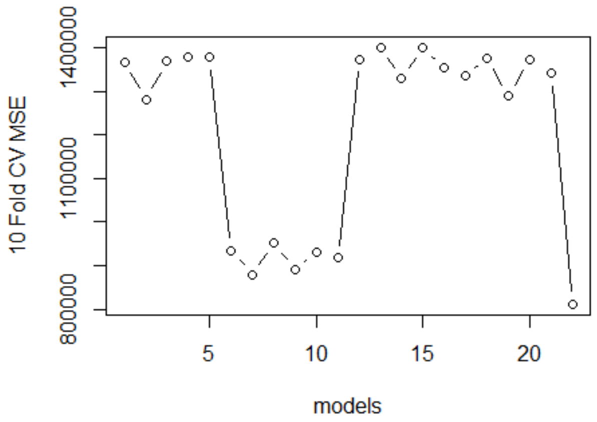
B1.3

```
models <- 1:22
kfolderrors <- numeric(22)
kfolderrors[1] <- cv.glm(train_economyau, m1,
K=10)$delta[1]
kfolderrors[2] <- cv.glm(train_economyau, m2,
K=10)$delta[1]
kfolderrors[3] <- cv.glm(train_economyau, m3,
K=10)$delta[1]
kfolderrors[4] <- cv.glm(train_economyau, m4,
K=10)$delta[1]
kfolderrors[5] <- cv.glm(train_economyau, m5,
K=10)$delta[1]
kfolderrors[6] <- cv.glm(train_economyau, m6,
K=10)$delta[1]
kfolderrors[7] <- cv.glm(train_economyau, m7,
K=10)$delta[1]
kfolderrors[8] <- cv.glm(train_economyau, m8,
K=10)$delta[1]
kfolderrors[9] <- cv.glm(train_economyau, m9,
K=10)$delta[1]
```

```
kfolderrors[10] <- cv.glm(train_economyau, m10, K=10)$delta[1]
kfolderrors[11] <- cv.glm(train_economyau, m11, K=10)$delta[1]
kfolderrors[12] <- cv.glm(train_economyau, m12, K=10)$delta[1]
kfolderrors[13] <- cv.glm(train_economyau, m13, K=10)$delta[1]
kfolderrors[14] <- cv.glm(train_economyau, m14, K=10)$delta[1]
kfolderrors[15] <- cv.glm(train_economyau, m15, K=10)$delta[1]
kfolderrors[16] <- cv.glm(train_economyau, m16, K=10)$delta[1]
kfolderrors[17] <- cv.glm(train_economyau, m17, K=10)$delta[1]
kfolderrors[18] <- cv.glm(train_economyau, m18, K=10)$delta[1]
kfolderrors[19] <- cv.glm(train_economyau, m19, K=10)$delta[1]
kfolderrors[20] <- cv.glm(train_economyau, m20, K=10)$delta[1]
kfolderrors[21] <- cv.glm(train_economyau, m21, K=10)$delta[1]
kfolderrors[22] <- cv.glm(train_economyau, m22, K=10)$delta[1]
kfolderrors

##  [1] 1367170.2 1281453.2 1368441.4 1380238.8 1377803.2  934966.3  880251.0
##  [8]  951644.3  892325.8  932012.7  918196.0 1373001.4 1400543.8 1330788.2
## [15] 1399246.5 1354369.8 1334441.9 1376894.4 1289785.4 1373111.9 1341065.3
## [22]  811559.1
```

B1.4

```
plot(models, kfolderrors, type="b", xlab = "models", ylab = "10 Fold CV MSE")
```



```
summary(m7)

## 
## Call:
## glm(formula = Index ~ Interest.rate + Inflation, data = train_economyau)
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    6259.96     190.04   32.94  < 2e-16 ***
## Interest.rate  -403.33      39.74  -10.15  < 2e-16 ***
## Inflation       229.59      53.14    4.32 2.65e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 834369.7)
## 
##     Null deviance: 231176469  on 172  degrees of freedom
## Residual deviance: 141842853  on 170  degrees of freedom
## AIC: 2854.7
## 
## Number of Fisher Scoring iterations: 2
```

B1.5

```
predict_m7 <- predict(m7, newdata = valid_economyau, type = "response")
```

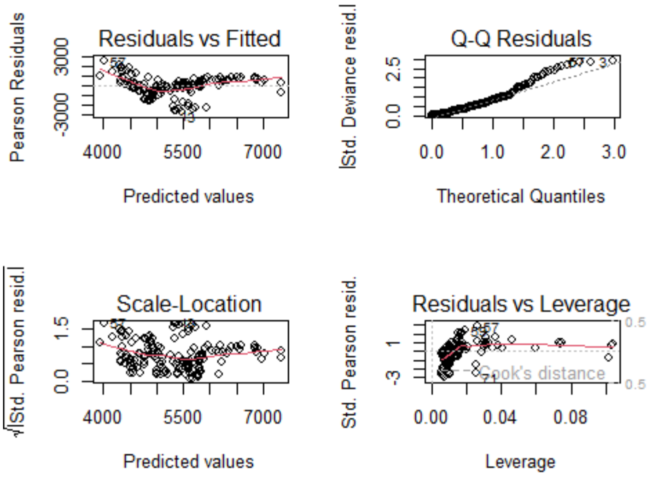
B1.6

```
MSE_m1 <- mean((actual_economyau - predict_m7)^2)
MSE_m1

## [1] 966586.4

sqMSE1 <- sqrt(MSE_m1)
sqMSE1
```

```
## [1] 983.1513
```

B1.7 model checking

```
par(mfrow = c(2, 2))
plot(m7)
```



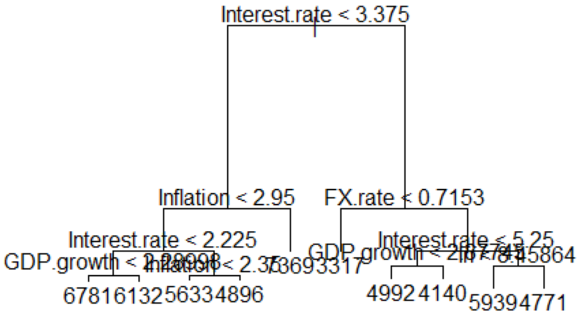
```
summary(m1)

##
## Call:
## glm(formula = Index ~ In, data = train_economyau)
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5258.82     182.30  28.846   <2e-16 ***
## In             10.01      15.75   0.635    0.526
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be
1348724)
##
##     Null deviance: 231176469  on 172  degrees of freedom
## Residual deviance: 230631878  on 171  degrees of freedom
## AIC: 2936.8
##
## Number of Fisher Scoring iterations: 2
```

C1.1 Decision Tree

```
library(tree)

## Warning: package 'tree' was built under R version 4.3.1

set.seed(77)
tr_economy <- sample(1:nrow(economyau),
round(nrow(economyau)*0.70, 0))
train_economyau <- economyau[tr_economy, ]
valid_economyau <- economyau[-tr_economy, ]
actual_economyau <- valid_economyau$Index
tree_economy1 <- tree(Index~., data = train_economyau)
summary(tree_economy1)

##
## Regression tree:
## tree(formula = Index ~ ., data = train_economyau)
## Variables actually used in tree construction:
## [1] "Interest.rate" "Inflation"     "GDP.growth"    "FX.rate"
## [5] "In"
## Number of terminal nodes:  10
## Residual mean deviance:  165300 = 26950000 / 163
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -1745.00  -218.60    62.51     0.00   228.80  1003.00

tree_economy1

## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 173 231200000 5360
##    2) Interest.rate < 3.375 78  56080000 6227
##      4) Inflation < 2.95 63  27580000 5955
##        8) Interest.rate < 2.225 30   5332000 6457
##         16) GDP.growth < 2.28998 15   1594000 6781 *
##         17) GDP.growth > 2.28998 15    575000 6132 *
##        9) Interest.rate > 2.225 33   7838000 5499
##         18) Inflation < 2.35 27   1422000 5633 *
##         19) Inflation > 2.35 6   3749000 4896 *
##      5) Inflation > 2.95 15   4289000 7369 *
##    3) Interest.rate > 3.375 95  68400000 4649
##      6) FX.rate < 0.7153 12   1656000 3317 *
##      7) FX.rate > 0.7153 83  42390000 4841
##       14) Interest.rate < 5.25 41  14780000 4514
##         28) GDP.growth < 2.87745 18   1927000 4992 *
##         29) GDP.growth > 2.87745 23   5519000 4140 *
##       15) Interest.rate > 5.25 42  18940000 5161
##         30) In < 8.45864 14   2647000 5939 *
##         31) In > 8.45864 28   3569000 4771 *
```

C1.2

```
par(mfrow=c(1,1))
plot(tree_economy1)
text(tree_economy1, pretty = 0)
```



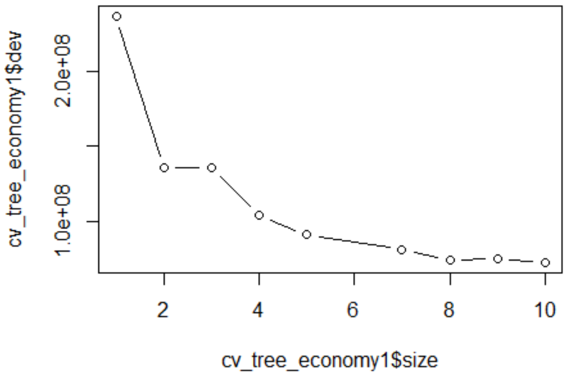
C1.3

```
predict_tree_economy1 <- predict(tree_economy1, newdata =
valid_economyau)
MSE_tree_economy1 <- mean((actual_economyau -
predict_tree_economy1)^2)
MSE_tree_economy1

## [1] 625685

sqMSE2 <- sqrt(MSE_tree_economy1)
sqMSE2

## [1] 791.0025
```

C1.4 purning using cross validation

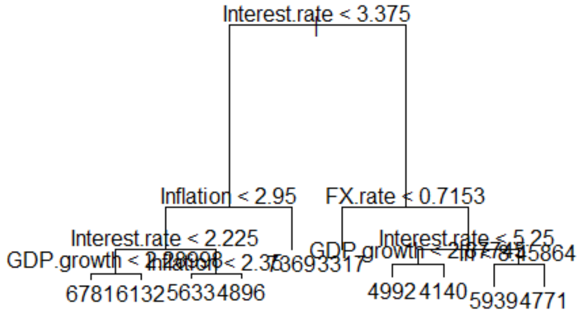
```
cv_tree_economy1 <- cv.tree(tree_economy1, FUN =
prune.tree)
cv_tree_economy1

## $size
## [1] 10  9  8  7  5  4  3  2  1
##
## $dev
## [1]  72192917  74622100  73627318  80915055  90656515
103831506 135711627
## [8] 135711627 237270093
##
## $k
## [1]      -Inf   2666573   3163487   7330413  10699386
14408898  24206943
## [8]  24348983 106703829
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

```
plot(cv_tree_economy1$size, cv_tree_economy1$dev,
type="b")
```



C1.5

```
pruned_tree_economy1 <- prune.tree(tree_economy1, best =
10)
pruned_tree_economy1

## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 173 231200000 5360
##    2) Interest.rate < 3.375 78  56080000 6227
##      4) Inflation < 2.95 63  27580000 5955
##        8) Interest.rate < 2.225 30   5332000 6457
##         16) GDP.growth < 2.28998 15   1594000 6781 *
##         17) GDP.growth > 2.28998 15    575000 6132 *
##        9) Interest.rate > 2.225 33   7838000 5499
##         18) Inflation < 2.35 27   1422000 5633 *
##         19) Inflation > 2.35 6   3749000 4896 *
##      5) Inflation > 2.95 15   4289000 7369 *
##    3) Interest.rate > 3.375 95  68400000 4649
##      6) FX.rate < 0.7153 12   1656000 3317 *
##      7) FX.rate > 0.7153 83  42390000 4841
##       14) Interest.rate < 5.25 41  14780000 4514
##         28) GDP.growth < 2.87745 18   1927000 4992 *
##         29) GDP.growth > 2.87745 23   5519000 4140 *
##       15) Interest.rate > 5.25 42  18940000 5161
##         30) In < 8.45864 14   2647000 5939 *
##         31) In > 8.45864 28   3569000 4771 *

par(mfrow=c(1,1))
plot(pruned_tree_economy1)
text(pruned_tree_economy1, pretty = 0)
```



C1.6 Test model accuracy

```
predict_tree_economy2 <- predict(pruned_tree_economy1,
newdata = valid_economyau)
MSE_tree_economy2 <- mean((actual_economyau -
predict_tree_economy2)^2)
MSE_tree_economy2

## [1] 625685

sqrt(MSE_tree_economy2)

## [1] 791.0025
```

D1.1 Unsupervised learning- Principal Component Analysis

```
Mean <- sapply(economyau, mean)
Variance <- sapply(economyau, var)
mv <- data.frame(Mean, Variance)
mv

##                      Mean     Variance
## Interest.rate 3.566397e+00 3.383378e+00
## GDP.growth    2.727964e+00 2.861462e+00
## Inflation     2.651417e+00 2.078768e+00
## Volume        1.900819e+10 3.440864e+19
## FX.rate       8.013036e-01 1.443930e-02
## In            9.971352e+00 3.633263e+01
## Index         5.368282e+03 1.446813e+06
```

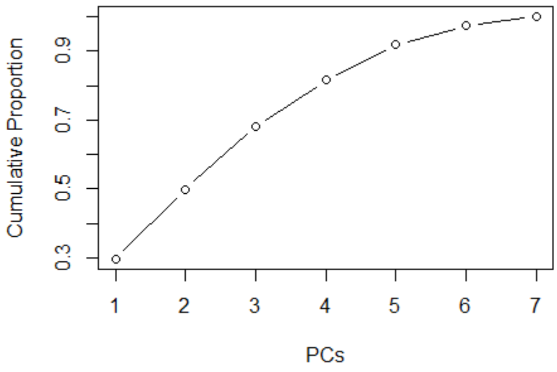
D1.2

```
pca1 <- prcomp(economyau, scale. = TRUE)
pca1

## Standard deviations (1, .., p=7):
## [1] 1.4413052 1.1879666 1.1329710 0.9654473 0.8472883
0.6263898 0.4305826
##
## Rotation (n x k) = (7 x 7):
```

```
##                        PC1        PC2         PC3           PC4        PC5
## Interest.rate -0.633535058 0.13832877 -0.08443970 -
8.151994e-05  0.1934766
## GDP.growth    -0.158182448 0.57241391  0.32767034 -
3.731845e-01 -0.4320678
## Inflation     -0.005297316 0.71420762  0.08882752
2.400256e-01  0.4667951
## Volume        0.138203609 0.21557725 -0.75881039
6.749259e-02  0.1990373
## FX.rate       -0.445472350 0.04690872 -0.50551168 -
1.052220e-01 -0.4698377
## In            0.352450268 0.08937961 -0.17621790 -
8.121253e-01  0.1883991
## Index         0.481473300 0.29402764 -0.12338453
3.576962e-01 -0.5119853
##                     PC6          PC7
## Interest.rate  0.1391790 -0.718028147
## GDP.growth    -0.4625751  0.005266813
## Inflation      0.3225234  0.320090222
## Volume        -0.5588889 -0.045882020
## FX.rate        0.3806899  0.408739421
## In             0.3426911 -0.155751031
## Index          0.2965018 -0.434187324
```

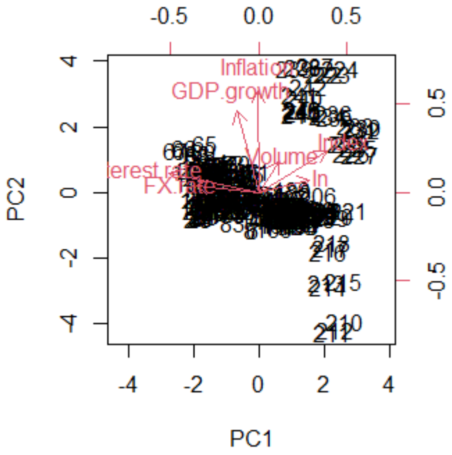
D1.3

```
summary(pca1)

## Importance of components:
##                           PC1    PC2    PC3    PC4    PC5     PC6
PC7
## Standard deviation     1.4413 1.1880 1.1330 0.9654 0.8473
0.62639 0.43058
## Proportion of Variance 0.2968 0.2016 0.1834 0.1332 0.1026
0.05605 0.02649
## Cumulative Proportion  0.2968 0.4984 0.6817 0.8149
0.9175 0.97351 1.00000

pca1_var <- pca1$sdev^2
pca1_var_pro <- pca1_var / sum(pca1_var)
pca1_var_cum <- cumsum(pca1_var_pro)
plot(1:7, cumsum(pca1_var_pro), type = "b", xlab = "PCs",
ylab = "Cumulative Proportion")
```



D1.4

```
biplot(pca1, scale = 0)
```



E1.1 Model Comparision

```
MSE <- c(MSE_m1, MSE_tree_economy1)
Sq.Root.MSE <- c(sqMSE1, sqMSE2)
Compare <- data.frame(MSE, Sq.Root.MSE)
rownames(Compare) <- c("Linear Regression", "Decision
Tree")
Compare

##                        MSE Sq.Root.MSE
## Linear Regression 966586.4    983.1513
## Decision Tree     625685.0    791.0025
```