FOSSEE Summer Fellowship Report

on

FLOSS - R

submitted by

Sakshee Phade (Pune Institute of Computer Technology)
M. Sai Anand (Kalasalingam Academy of Research and Education)

under the guidance of

Prof. Kannan M. Moudgalya
Chemical Engineering Department
IIT Bombay

Prof. Radhendushka Srivastava
Department of Mathematics
IIT Bombay

June 15, 2020

Acknowledgement

We want to express our sincere gratitude to Prof. Kannan M. Moudgalya, Department of Chemical Engineering, IIT Bombay for creating the FOSSEE Fellowship programme and providing students from all over India an opportunity to participate in it. We would equally like to thank our FLOSS mentor, Prof. Radhendushka Srivastava, Department of Mathematics, IIT Bombay, for his immense support, patience, motivation, knowledge and influence throughout this research project and for helping us on various statistical models. We would also like to express our gratitude to other members of the R FLOSS team, namely Mrs. Smita Wangikar and Mr. Digvijay Singh for their guidance and valuable inputs throughout the fellowship and also for providing us with an overview on data analysis and LaTeX. We are very grateful to be given such a fantastic opportunity to work on this exciting project. We would also like to thank the other fellows who got selected along with us, namely Ashwin Guptha and Amish Sharma for their support, intellectual discussions and enthusiasm.

Contents

Chapter 1

Introduction

In this report, we mention our contributions to open-source software (FLOSS), made in the duration of the FOSSEE Fellowship, starting from 20th April 2020 to 15th June 2020. Contributions were made using a Free-Libre/Open Source Software (FLOSS) known as "R" as a part of the FOSSEE Project by IIT Bombay and MHRD, Government of India. FOSSEE project is a part of the National Mission on Education through ICT. The thrust area is the adaptation and deployment of open-source simulation packages equivalent to proprietary software, funded by MHRD, based at the Indian Institute of Technology Bombay (IITB). Our contributions involved making Spoken Tutorial scripts and analysis of the difference between the predicted GDP and actual GDP of various nations.

Chapter 2

Spoken Tutorial

The Spoken Tutorial project aims to make spoken tutorials on Free and Open Source Software (FOSS) available in several Indian languages. The goal is to enable the use of Spoken Tutorials to teach in any Indian language to learners of all levels of expertise - Beginner, Intermediate or Advanced. Every tutorial has to go through a series of stages to ensure that it is perfect for its audience, which is crucial for achieving the goal of this project. We contributed to the creation of the tutorial scripts for "Introduction to Classification and Clustering," "Data Cleaning in R," "Linear Discriminant Analysis," "K- means Clustering," and "Hierarchical Clustering."

2.1 Introduction to Classification and Clustering

The tutorial script includes a brief introduction to Classification and Clustering in machine learning. Classification algorithms recognize the category of new inputs based on a training set of data. The organization of unlabeled data into similar groups is called "Clustering." The tutorial also includes information on the usage of operations like scaling and standard machine learning libraries in R.

2.2 Data Cleaning in R

Data cleaning involves transforming raw data obtained directly from the sources into ordered, consistent data that can be analyzed. Data cleaning improves data quality and in doing so, increases overall productivity. The tutorial demonstrates processes such as reading data from a text file (.txt), type conversion, translating characters into factors and checking data set for missing values. It includes "airquality" data set for the demonstration of data cleaning processes.

2.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a linear modeled dimension reduction technique. It is used to model the difference in groups present in a data set while retaining as much information as possible. The tutorial script includes "PimaIn-

dianDiabetes" data set for classifying the test for diabetes of the patients. The packages used were "mlbench" [1] and "MASS" [2].

2.4 K-means Clustering

The organization of unlabeled data into similar or homogeneous groups is known as Clustering. K-means is a clustering algorithm; it partitions or divides the given data into "k" clusters. Each cluster has a cluster center, known as the Centroid. The tutorial script involves "New York Uber service" data set for clustering the pickup locations using the k-means algorithm and "factoextra" [3] & "ggplot2" [4] packages.

2.5 Hierarchical Clustering

Hierarchical clustering is an unsupervised clustering algorithm that groups similar objects (clusters) having predominant ordering from top to bottom. The tutorial script includes "milk" data set for clustering animals based on the nutrients present in their milk and "flexclust" [5], "cluster" [6] & "factoextra" [3] packages.

Chapter 3

Analysis of difference between the predicted GDP and actual GDP

3.1 Abstract

The growth or decline of a country's GDP determines the health of its economy. Though it is measured and predicted by every country's government-backed statistical body, there are various financial organizations like IMF, OECD, Moody's, JP Morgan, etc. which also do the same by using its economists and statisticians. To substantiate their accuracy, we analyzed the data produced by the two financial organizations, i.e., IMF and OECD. We also clustered countries based on the mean square error between actual GDP and estimated GDP, which showed how accurately the GDP was predicted for each of them.

3.2 Introduction

The project's objective was to find whether the financial bodies over-estimated or under-estimated the projected GDP for all the examined countries. The data obtained from the IMF and OECD were explored and analyzed. Furthermore, we checked for stationarity in the data by visualizing the correlations. Performing tests like Ljung-Box and ARCH provided us confirmation on the presence of white noise in the series, which was an indication to run GARCH model on the data of all the countries. We found the best model by taking into consideration the mean squared error obtained from each model. We further applied K-means clustering based on the smallest prediction error.

3.3 Methodology

3.3.1 Data Collection

3.3.1.1 Actual GDP growth data

3.3.1.1.1 World Bank:

The Development Data Group of the World Bank [7] cooperates with several macros, financial and sector organization's databases. The GDP data is collected in local currency rate by World Bank's economists, using the information published by the individual country's statistical authorities or sourced from OECD.

Data source: **World Bank GDP data** [8]

Year	Australia	Canada	United States	India
1980	3.034068	2.155537	-0.2567519	6.735822
1981	3.337959	3.474126	2.5377187	6.006204
1982	3.328400	-3.187262	-1.8028745	3.475733
1983	-2.220458	2.601339	4.5839273	7.288893
1984	4.581270	5.908457	7.2366200	3.820738
1985	5.249241	4.737400	4.1696560	5.254299

Figure 3.1: World Bank GDP data

3.3.1.2 Predicted GDP growth data

IMF (International Monetary Fund) and OECD (Organization for Economic Co-operation and Development) publish their forecasted data for every biannual analysis of the major economic trends and prospects for the next two years through the World Economic Outlook Report [9] and OECD Economic Outlook [10].

3.3.1.2.1 IMF:

The IMF follows a sophisticated approach in forecasting its GDP; i.e., all the member countries generate projections on their own. The data obtained from various IMF-backed sources regarding every country's GDP forecasts are aggregated. It is then cross-checked with the predictions reported in the WEO. Since each country's teams generate forecasts individually, the methodology can vary from country to country depending upon various factors.

Data source: **IMF GDP data** [11]

Year	Australia	Brazil	Canada	C. African Republic	Germany
1980	2.9	9.2	2.2	-3.0	1.3
1981	4.1	-4.4	3.5	13.0	0.1
1982	0.1	0.6	-3.2	-3.6	-0.8
1983	-0.5	-3.4	2.6	-6.0	1.6
1984	6.3	5.3	5.9	9.9	2.8
1985	5.5	7.9	4.7	3.7	2.2

Figure 3.2: IMF's world economic outlook report GDP data

3.3.1.2.2 OECD:

OECD's GDP forecast is based on the economic climate of individual countries. It employs a combination of model-based analyses and expert judgement. The measurement of the indicator is done in growth rates compared to the previous year.

Data source: **OECD GDP data** [12]

Year	Australia	Canada	USA	India	Germany
1980	2.7153521	2.155536	-0.2567556	NA	NA
1981	4.1817777	3.477538	2.5377262	NA	NA
1982	-0.3570328	-3.188331	-1.8028780	NA	NA
1983	-0.1940294	2.601399	4.5839236	NA	NA
1984	6.5825593	5.909302	7.2366273	NA	NA
1985	5.0323649	4.737634	4.1696525	NA	NA

Figure 3.3: OECD's economic outlook GDP data

3.3.2 Data Exploration

Economically, the classification of nations is based on their development index, mainly on their GDP per capita index. The GDP per capita index is calculated by dividing a country's GDP by its population. To perform data exploration, we chose fifteen countries out of which five were developed (Australia, Canada, USA, Switzerland & Germany), five were developing (India, South Africa, Mexico, Brazil & Turkey) and five were under-developed (Sudan, Venezuela, Central African Rep., Mozambique & Niger).

"ggplot2" [4] package was used to plot the data obtained from IMF & OECD against actual GDP data. The function "ggarange()" from the package "ggpubr" [13], was used to arrange multiple plots together.

Following code chunk depicts the comparison between predicted GDP data by the IMF & OECD and actual GDP data for developed, developing and under-developed countries -

```
# Loading library " ggplot2 " for data visualization.
# Loading library " ggpubr " for combining the generated plots.
library(ggplot2)
library(ggpubr)
# Plot for developed countries.
Developed <- ggarrange(Australia,Canada,USA,Germany,Switzerland,
                       ncol = 3,
                       nrow = 2,
                       common.legend = TRUE,
                       legend = "bottom")
Developed
# Plot for developing countries.
Developing <- ggarrange(India,Brazil,South.Africa,Turkey,Mexico,
                        ncol = 3,
                        nrow = 2,
                        common.legend = TRUE,
                        legend = "bottom")
Developing
# Plot for under-developed countries.
Under.developed <- ggarrange(CAR,Venezuela,Mozambique,Niger,Sudan,
                             ncol = 3,
                             nrow = 2,
                             common.legend = TRUE,
                             legend = "bottom")
Under.developed
```

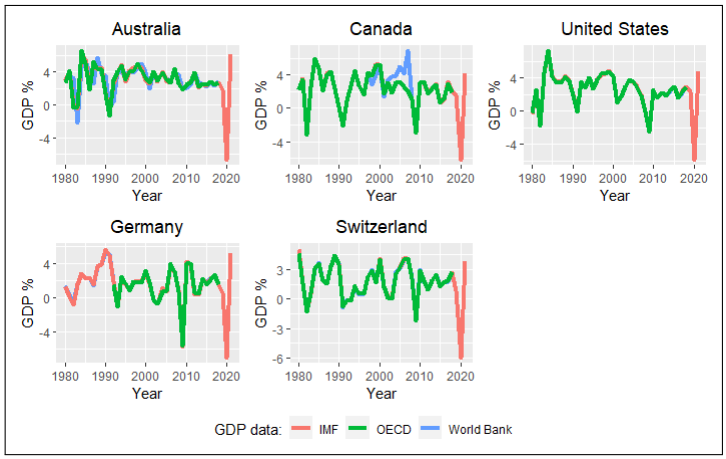


Figure 3.4: Comparison between predicted GDP data by the IMF & OECD and actual GDP data for developed countries

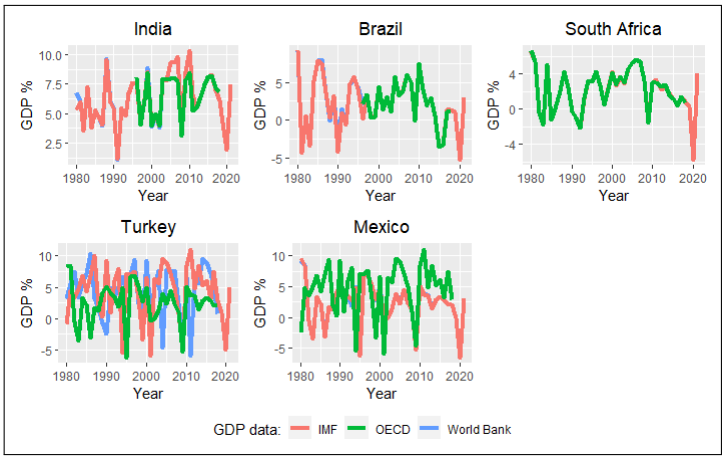


Figure 3.5: Comparison between predicted GDP data by the IMF & OECD and actual GDP data for developing countries

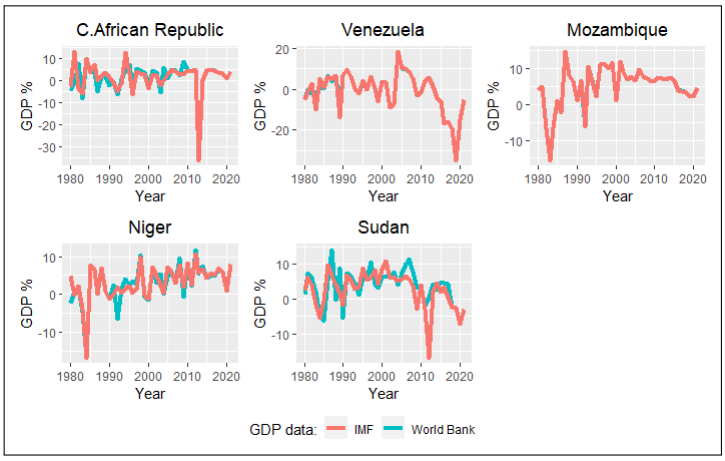


Figure 3.6: Comparison between predicted GDP data by the IMF & OECD and actual GDP data for under-developed countries

On observing figures 3.4, 3.5 and 3.6, we perceived that there existed some differences between the actual and predicted data points. To make the dispute between predicted and actual data more prominent, we plotted the square of their difference. Further data analysis was performed on the square of difference to make patterns broader and more comfortable to visualize. The "select()" function from "dplyr" [14] package was used to manipulate and restructure the data.

The following code chunk finds and plots the square of the difference between the predicted (IMF & OECD) and actual data -

```
# Loading library " dplyr " for data manipulation.
library(dplyr)
# Data reformatting for finding the difference of actual and predicted data.
actualNew <- as.data.frame(actual)
```

```
actualNew$Central.African.Republic <- NULL
actualNew$Venezuela <- NULL
actualNew$Sudan <- NULL
actualNew$Niger <- NULL
actualNew$Mozambique <- NULL
actualNew$Years <- NULL
predict1New <- as.data.frame(predict1)
predict1New <- select(predict1New, Australia, Canada, United.States, India, Germany, Switzerland, Brazil, South.Africa, Turkey, Mexico, Central.African.Republic
                      , Venezuela, Mozambique, Niger, Sudan)
actual$Years <- NULL
predict1New <- as.data.frame(predict1New)
predict1New <- predict1New[-c(40, 41, 42), ]
# Obtaining the difference.
difference1 = predict1New - actual
row.names(difference1) <- seq(1980,2018)
difference2 = val - actualNew
Yr <- seq(1980,2018)
# Square of differences.
diff1.sq <- difference1^2 # IMF
diff2.sq <- difference2^2 # OECD
# Built-in function to plot the square of differences.
plotCountry <- function(country, difference, countryName) {
  Yr <- seq(1980, 2018)
  plotvar <- ggplot() + geom_line(data = difference, aes(x = Yr, y = as.numeric(as.matrix(country)), color="Red"), size = 1) +
    ggtitle(countryName) + labs(x = " ", y = " ")
  return(plotvar)}
# Arranging the plots for IMF.
Diff.compare <- ggarrange(Australia, India, CAR, Canada, Brazil, Venezuela,  USA, South.Africa, Mozambique, Germany, Turkey, Niger, Switzerland, Mexico,
                          Sudan,nrow = 5, ncol = 3, common.legend = TRUE, align = "v")
# Graph aesthetics.
annotate_figure(Diff.compare,
                bottom = text_grob("Years"),
                left = text_grob("Difference", rot = 90),
                fig.lab = "Comparison of differences - IMF",)
# Arranging the plots for OECD.
Diff.compare <- ggarrange(Australia, India, Canada, Brazil,  USA, South.Africa, Germany, Turkey, Switzerland, Mexico,
                          nrow = 5, ncol = 2, common.legend = TRUE, align = "v")
# Graph aesthetics.
annotate_figure(Diff.compare,
                bottom = text_grob("Years"),
                left = text_grob("Difference", rot = 90),
                fig.lab = "Comparison of differences - OECD",)
```

Figures 3.7 and 3.8 depict the square of differences between the predicted and actual data.

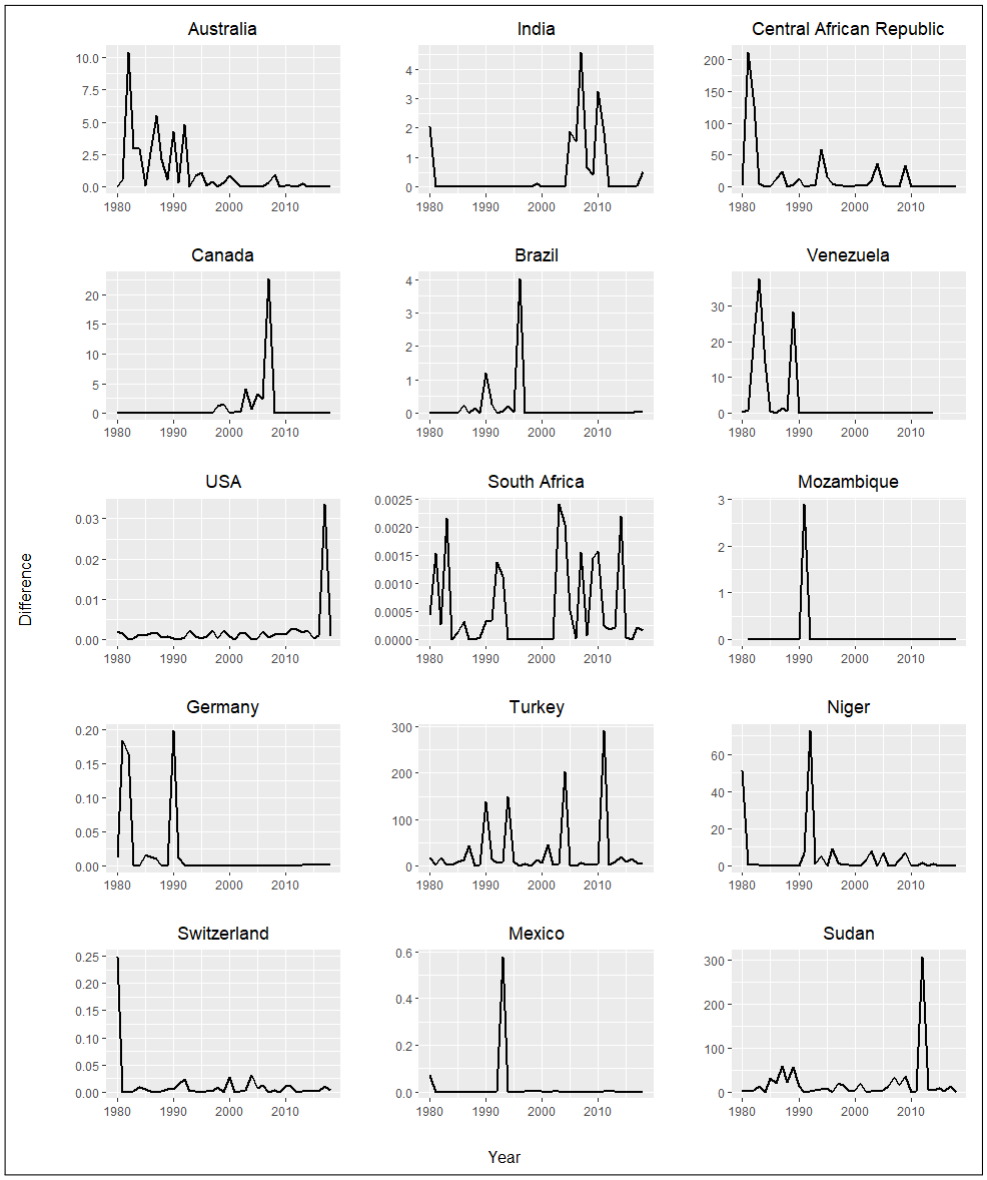


Figure 3.7: Square of difference between the predicted IMF data and actual data

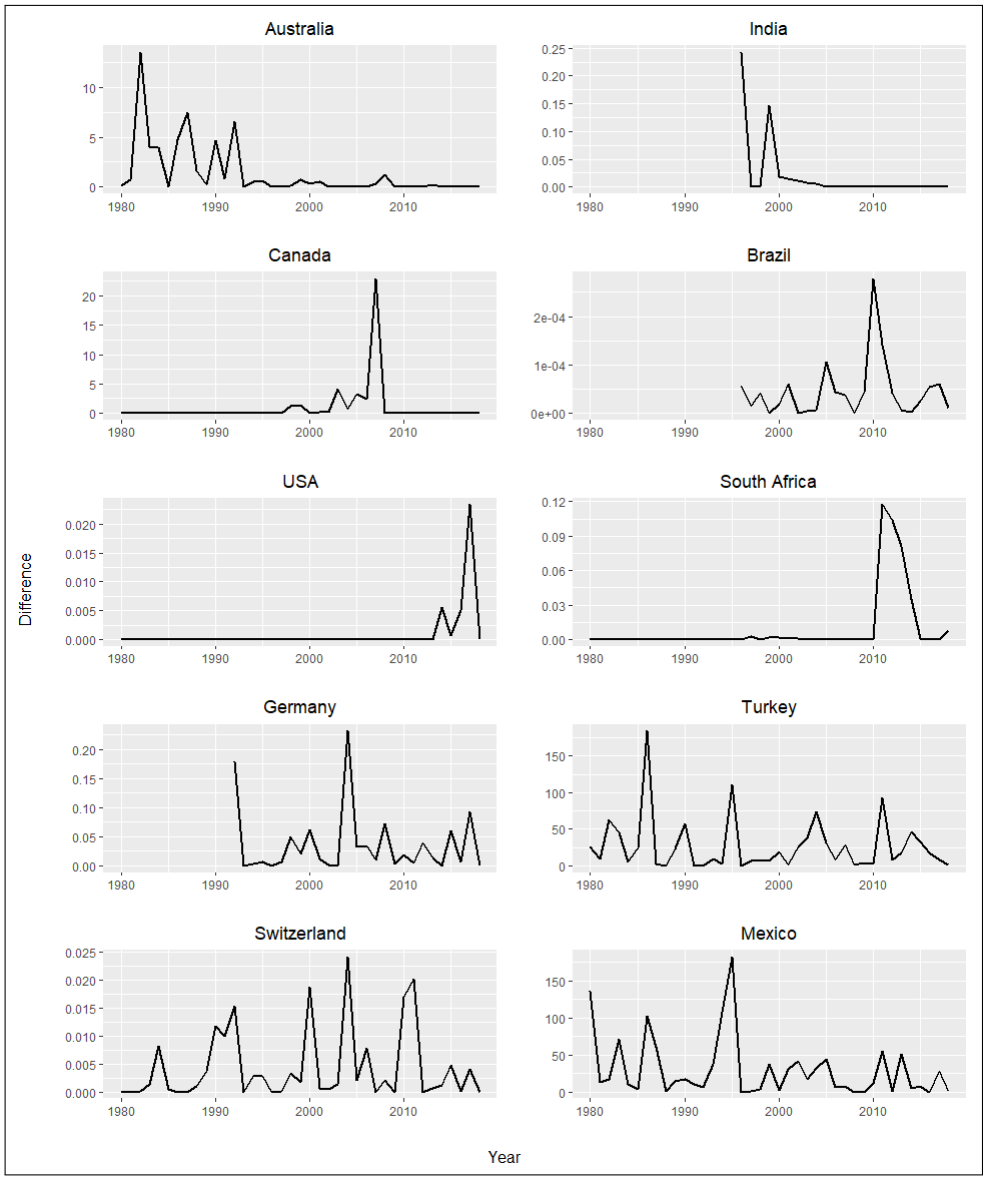


Figure 3.8: Square of difference between the predicted OECD data and actual data

3.3.3 Data Analysis

3.3.3.1 Correlograms

Correlograms are plots that summarize the strength of a relationship graphically, with observation in a time series with observations at previous time steps.

3.3.3.1.1 Autocorrelation function (ACF):

In a time series, autocorrelation describes the linear relationship between lagged values. It is the correlation between pairs of values present in data at a certain length. Since our data was a white noise series, it was expected that at least 95%

of the spikes generated lie within the bound limit "X", where

$$X = \frac{2}{T} \tag{3.1}$$

Here, "T" represents the length of the time series.

3.3.3.1.2 Partial autocorrelation function (PACF):

A partial autocorrelation is a type of correlation which is conditional. PACF is the correlation between observations in a time series and the previous observations at lag "k." It is different from a regular correlation in one aspect, i.e., it takes control of all the intermediate values present in the time series. It is also useful in finding the order of the autoregression process.

In R, the functions "acf()" and "pacf()" from the package "stats" [15] make it easy to find the ACF and PACF of a time series. We created a user-defined function containing the pre-defined "acf()" and "pacf()" functions to plot the square of differences between the actual and predicted (IMF & OECD) GDP values.

```
# ACF
plotACF <- function(country, country.name) {
  country <- na.omit(country)
  Acf.var1 <- acf(country, lag.max = 20, type = "correlation", plot = FALSE)
  plot(Acf.var1, main = country.name)
}
# PACF
plotPACF <- function(country, country.name) {
  country <- na.omit(country)
  Pacf.var1 <- pacf(country, lag.max = 20, type = "correlation", plot = FALSE)
  plot(Pacf.var1, main = country.name)
}
```

3.3.3.2 ARCH Model

A time series is said to be heteroskedastic when the variance is not constant in time but changes regularly, i.e., an increase in variance with the trend. If a time series exhibits periods of increased variation, then the series shows volatility and is called conditional heteroskedastic. ARCH or Auto-Regressive Conditional Heteroskedastic is a volatility model for the variance of the times series which showcases the change in the conditional variance.

ARCH of order "1", i.e., "ARCH(1)" is defined as,

$$\epsilon_t = \omega_t \times \sqrt{\alpha_0 + \alpha_1 \epsilon_{t-1}^2} \tag{3.2}$$

where "ϵ_t" is a time series, "ω_t" is white noise with mean "0" & "α_0" and "α_1" are model parameters.

"garch()" function from the package "tseries" [16] can be used to fit the ARCH model over given data. It returns various parameters like "Estimate", "Std. Error",

"t-value", "Pr(>|t|)", "Ljung-Box X-squared", "Ljung-Box p.value" referring to the standard estimate error, t-value, significance of t-test, p-value of Ljung-Box test and X-squared value of Ljung-Box test respectively, which determines the significance of the model.

The following code snippet was executed to obtain the ARCH model parameters from the data under consideration -

```
# ARCH model over the square of difference.
Arch1 <- function(country) {
  country <- na.omit(country)
  model <- quiet(garch(x = country,order = c(0,1),trace = F))
  summ <- summary(model)
  t <- as.data.frame(summ$coef[2,])
  col1 <- c(rownames(t), "Ljung-Box X-squared", "Ljung-Box p.value")
  col2 <- c(t[,1], as.numeric(summ$l.b.test$statistic)
            , as.numeric(summ$l.b.test$p.value))
  df <- data.frame("Coeff"=col1, "Value"=col2)
  colnames(df) <- c("Value")
  return(df)
}
# To stop the summarized output generated from function execution.
quiet <- function(x) {
  sink(tempfile())
  on.exit(sink())
  invisible(force(x))
}
# Applying the user-defined ARCH model function over the data.
imf <- lapply(na.omit(diff1.sq), Arch1)
oecd <- lapply(na.omit(diff2.sq), Arch1)
```

The results were then stored into a table using the functions "grid.arrange()" & "tableGrob()" from the packages "grid" [15] & "gridExtra" [17] respectively.

We observed that OECD had fifty member countries whereas IMF had one hundred eighty-nine member countries. Since IMF had more countries, we chose to work on the data associated with IMF. After data cleaning, the number of countries got reduced to one hundred eleven and further data analysis was implemented on those countries.

3.3.3.3 ARMA Model

ARMA or Autoregressive Moving Average is a volatility model for the mean of a times series which is used to showcase the change in conditional mean. ARMA is a union of "AR" and "MA" models. It describes a time series in terms of two polynomials, where the first polynomial denotes the Autoregression (AR) and the second polynomial denotes the Moving Average (MA). It is often referred to as the "ARMA(p,q)" model, where "p" and "q" denote the order of the autoregressive polynomial and the order of the moving average polynomial.

It is defined as,

$$X_t = c + \omega_t + \sum_{i=1}^{p} \varphi_i X_{t-i} + \sum_{i=1}^{q} \theta_i \omega_{t-i} \tag{3.3}$$

where "X_t" is a time series, "c" is a constant, "ω" is white noise, "φ_i" is the autoregressive model's parameter and "θ_i" is the moving average model's parameter.

3.3.3.4 GARCH Model

Bollerslev developed generalized Auto Regressive Conditional Heteroskedasticity (GARCH) in 1986. It is a statistical model where the volatility or the variance depends on the previous residual squared observations or the past variances of a time series. GARCH fits the autoregressive model, which yields the best fit. It returns the irregularity of the error term, i.e., heteroskedasticity & significance value, and describes both the conditional mean and conditional variance. Due to which the best order of fit depends on the optimal order based on "ARCH", "AR" and "MA", i.e., the ARMA model.

"GARCH(1,1)" is defined as,

$$\epsilon_t = \omega_t \times \sqrt{h_t} \tag{3.4}$$

where "ϵ_t" is a series, "ω_t" is white noise and "h_t" is a volatility or conditional variance. Also,

$$h_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-1}^2 + \sum_{j=1}^{q} \beta_j h_{t-j} \tag{3.5}$$

where "α_i" and "β_j" are the model parameters.

3.3.3.5 Best order of fit

We have used the GARCH model because we needed to figure out an approximation of the model which these agencies have used to forecast the GDP. It is assumed that the same model must have been used to predict GDP for all countries. In a quest to find an approximate model, we focused on the mean squared prediction error, which was obtained by adding the squared differences between the actual values and fitted values. A fitted value is a statistical model's prediction of the mean response value when factor values are taken as the input.

Mean squared prediction error is defined as,

$$\sum_{i=1980}^{2018} (y_i - \widehat{y_i}) \tag{3.6}$$

where "y_i" is the actual difference value and "$\widehat{y_i}$" is the fitted value obtained from our GARCH-ARMA model.

To create a univariate GARCH specification object before fitting, we must pass the parameters concerned with ARCH, GARCH, AR and MA orders. The smallest mean squared prediction error indicates the optimum combination of these orders. We calculated the mean squared prediction error for combinations ranging from (0,0,0,0) to (3,3,3,3) and obtained the smallest error among them. The best order of fit was recorded for each country. After getting the best model for every country,

we grouped countries based on the order of fit and mean squared prediction error using clustering.

To get the maximum likelihood estimation of the GARCH model, we used the function "ugarchspec()" to create the GARCH model and function "ugarchfit()" to fit the model, from package "rugarch" [18].

```
# Finding the best parameter order for model prediction.
# Model definition.
Model <- function(Country,Order)
{ # Inputs
  m <- Order[1]
  n <- Order[2]
  p <- Order[3]
  q <- Order[4]
  m <- as.numeric(m)
  n <- as.numeric(n)
  p <- as.numeric(p)
  q <- as.numeric(q)
  # Model
  Garch <- suppressWarnings(ugarchspec(variance.model = list(garchOrder = c(m,n)),
      mean.model = list(armaOrder=c(p,q))))
  Fit <- suppressWarnings(ugarchfit(Garch, Country))
  # Results
  FV <- Fit@fit$fitted.values
  if(!is.null(FV)) {
    pred.error.sq <- (Country - FV)^2
    sum.pred.error.sq <- sum(pred.error.sq)
    return(sum.pred.error.sq)
  }
  else {
    return(NULL)
  }

}
# All possible input combinations.
List <- with(expand.grid(0:3,0:3,0:3,0:3), paste(Var1, Var2, Var3, Var4))
Combinations <- matrix(0L,nrow = length(List),ncol = 4)
for(i in 1:length(List))
{
  Combinations[i,] <- as.integer(unlist(strsplit(List[i]," ")))
}
# Removing unnecessary objects.
rm(i,List)
try <- Model(Difference[,111], c(1,1,1,1))
```

3.3.3.6 Clustering

Countries can be divided into clusters based on their best order of fit and the smallest mean squared prediction error. We used the K-means clustering algorithm for this purpose. It assigns data points to a cluster in a way that the sum of the average squared distance between the cluster's centroid and the data points is at the minimum. The smaller the variation within clusters, the more homogeneous the data within them.

After observing an un-clustered plot of the errors for each country, we noticed that the errors could be roughly divided into three clusters. Therefore, we defined the target number as three, which was the number of centroids we needed in our data set. These centroids were used as the beginning points for their respective cluster, and then the clustering algorithm performed repetitive calculations to optimize the

position of the centroids. Thus countries having order and error in the same range were combined into the same cluster.

The purpose of clustering was to group countries fitting well over similar models and each having an almost same error. After observing the clusters, we found out the names of the countries which had a negligible error, medium error or high error. Further, a list is tabulated based on the errors.

The code snippet depicting the plot of the final three clusters is as follows -

```
# Creates dataframe with optimum order and mean square error for every country.
orders_and_errors <- cbind(orders.all, smallest.pred.error)
clus <- kmeans(na.omit(orders_and_errors), 3, nstart = 25) # K-means clustering
na.index <- which(!is.na(smallest.pred.error)) # Stores indices of countries with "NA" values.
na.index
clus$cluster
d <- as.data.frame(na.omit(smallest.pred.error))
# Cluster plot.
ggplot(data = d, aes(x=colnames(na.omit(Difference[na.index])), y=na.omit(smallest.pred.error), color=clus$cluster)) + geom_point() +
  scale_x_discrete(guide = guide_axis(n.dodge = 6)) + xlab("Countries") + ylab("Prediction errors") + + coord_flip()
```

3.4 Results

3.4.1 Correlograms

The correlogram is a visualization technique which shows the correlation of the data that changes over time. Since we had 39 data points, therefore "T" was given the value "39" and the bounds were obtained at $2/39 = \pm$ **0.32** (from equation 3.1).

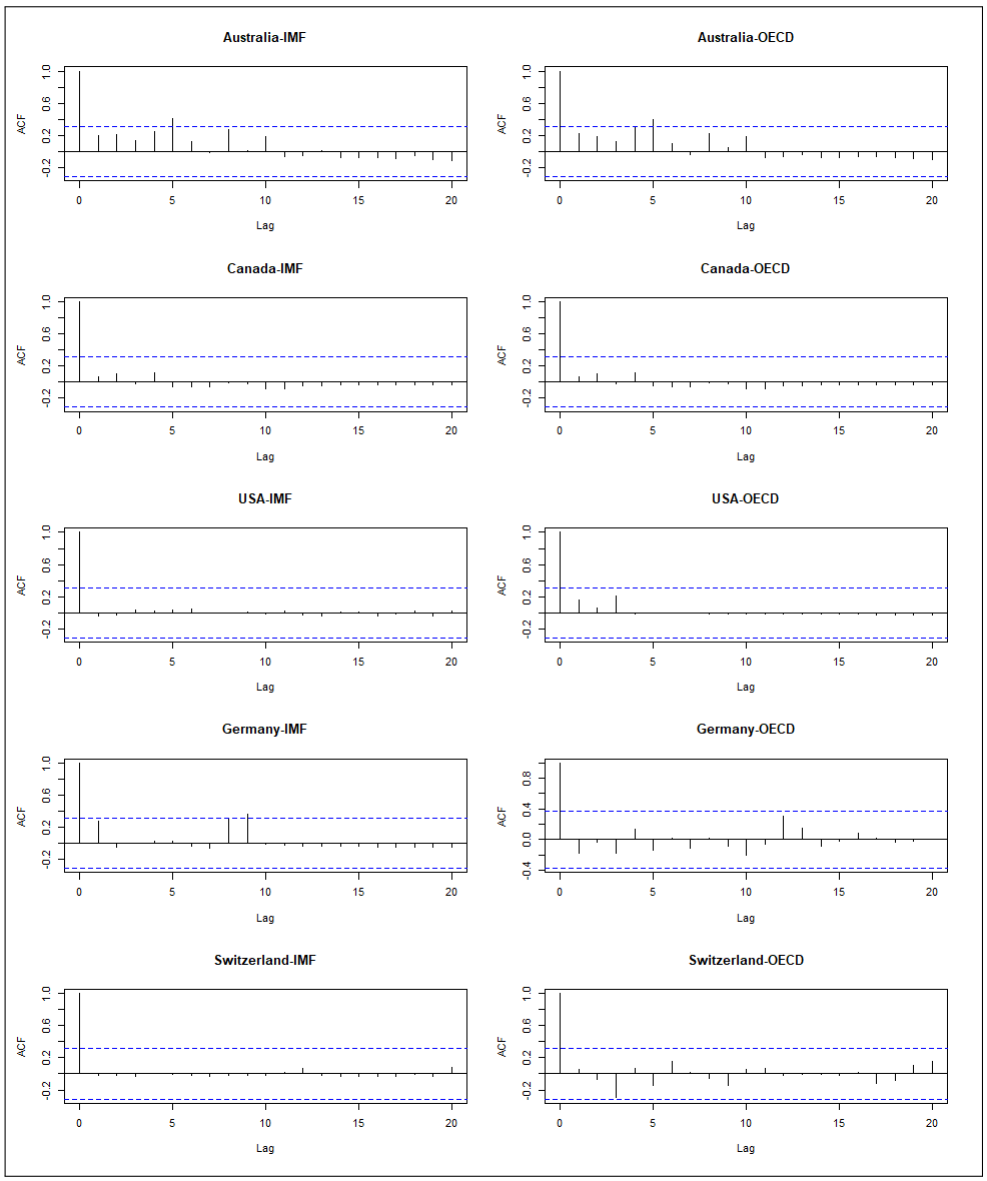


Figure 3.9: ACF plot for developed countries

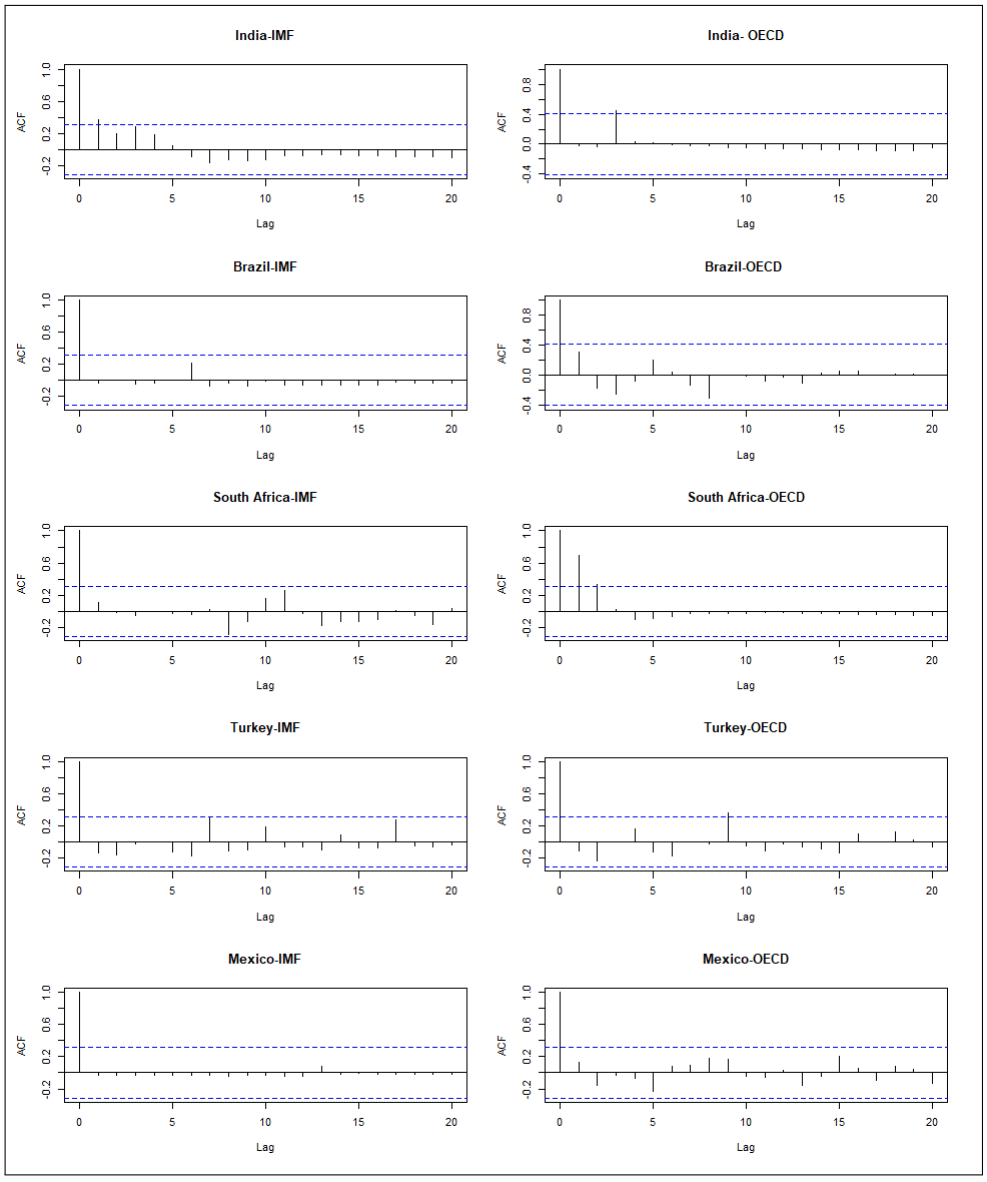


Figure 3.10: ACF plot for developing countries

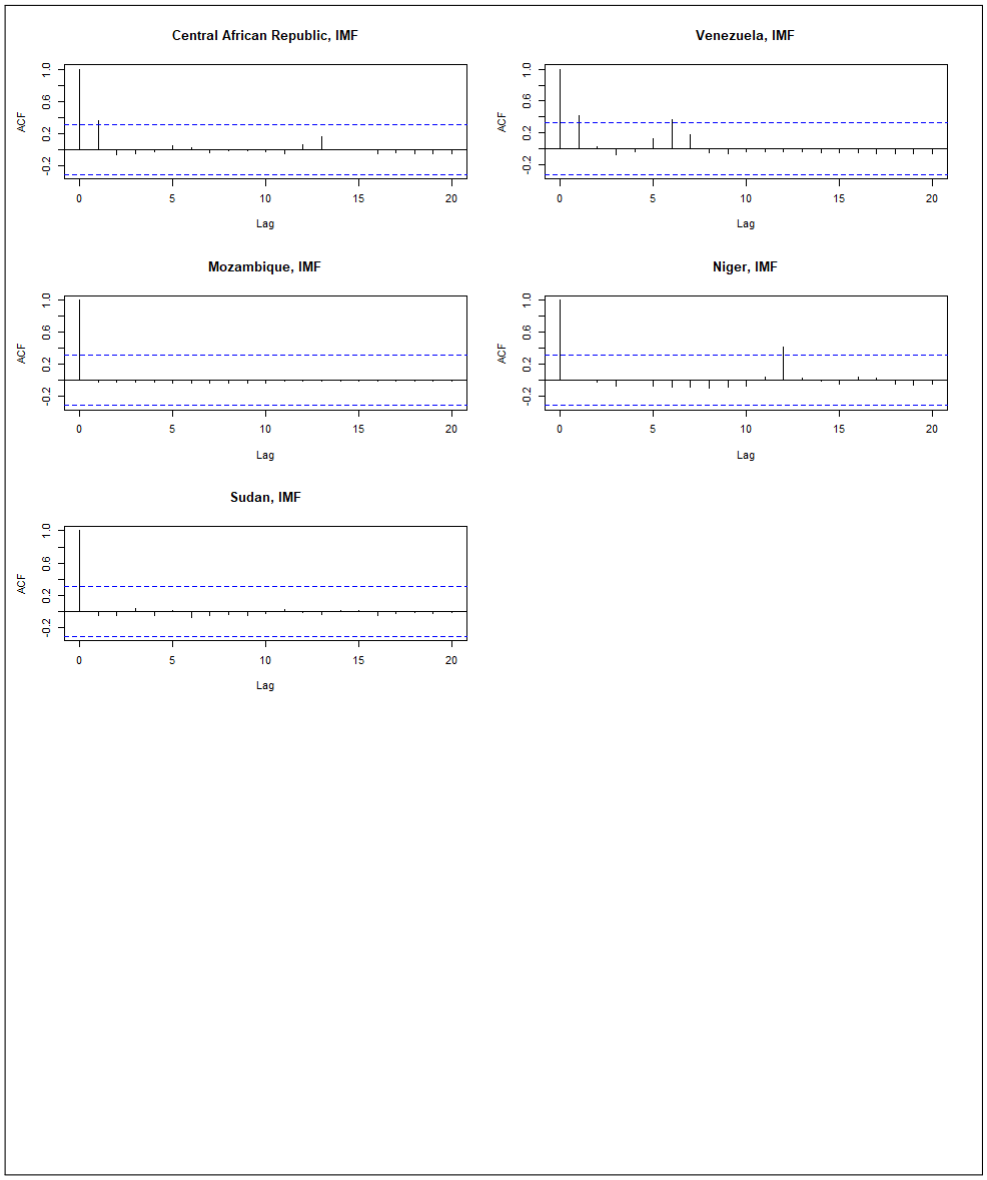


Figure 3.11: ACF plot for under-developed countries

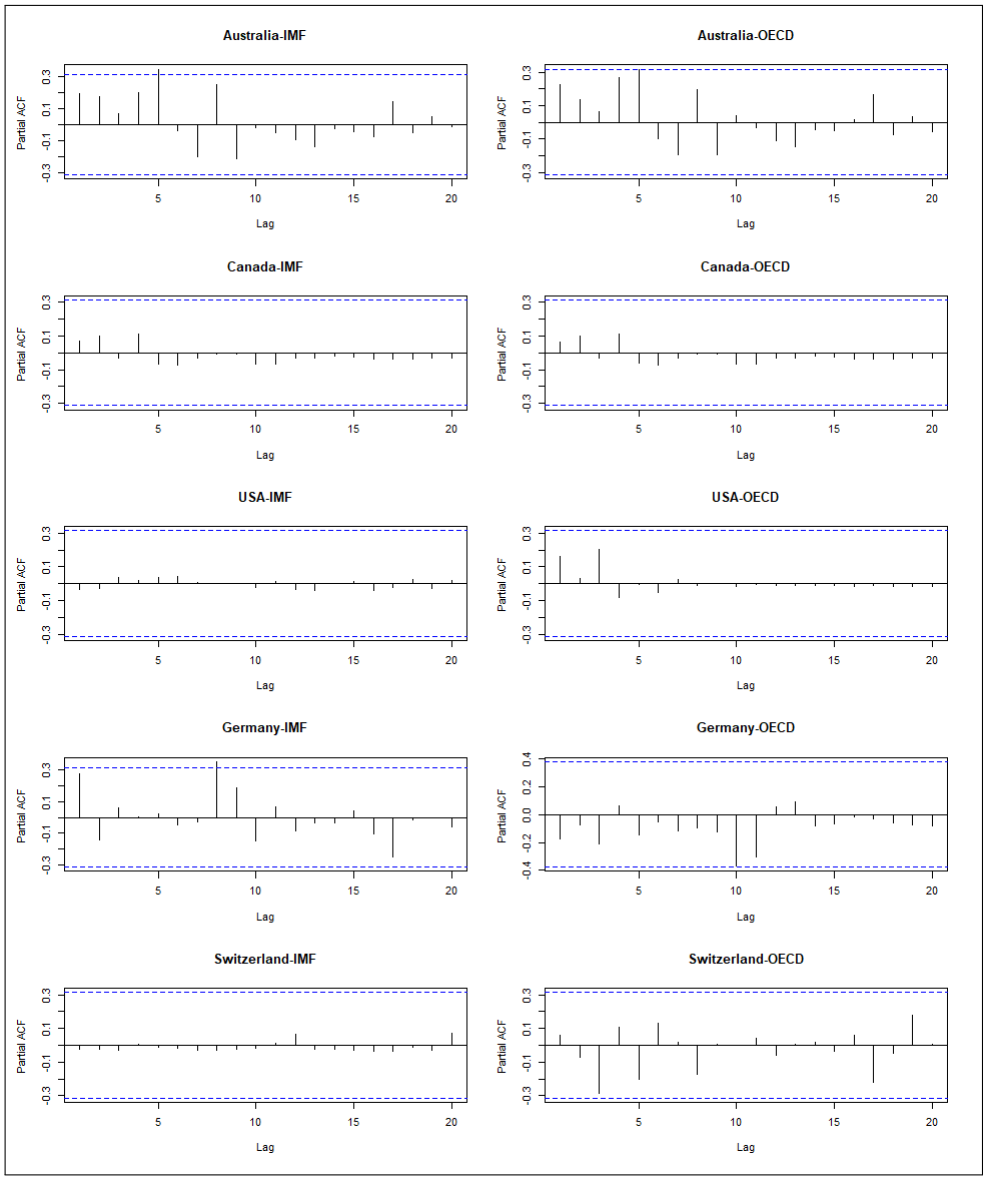


Figure 3.12: PACF plot for developed countries

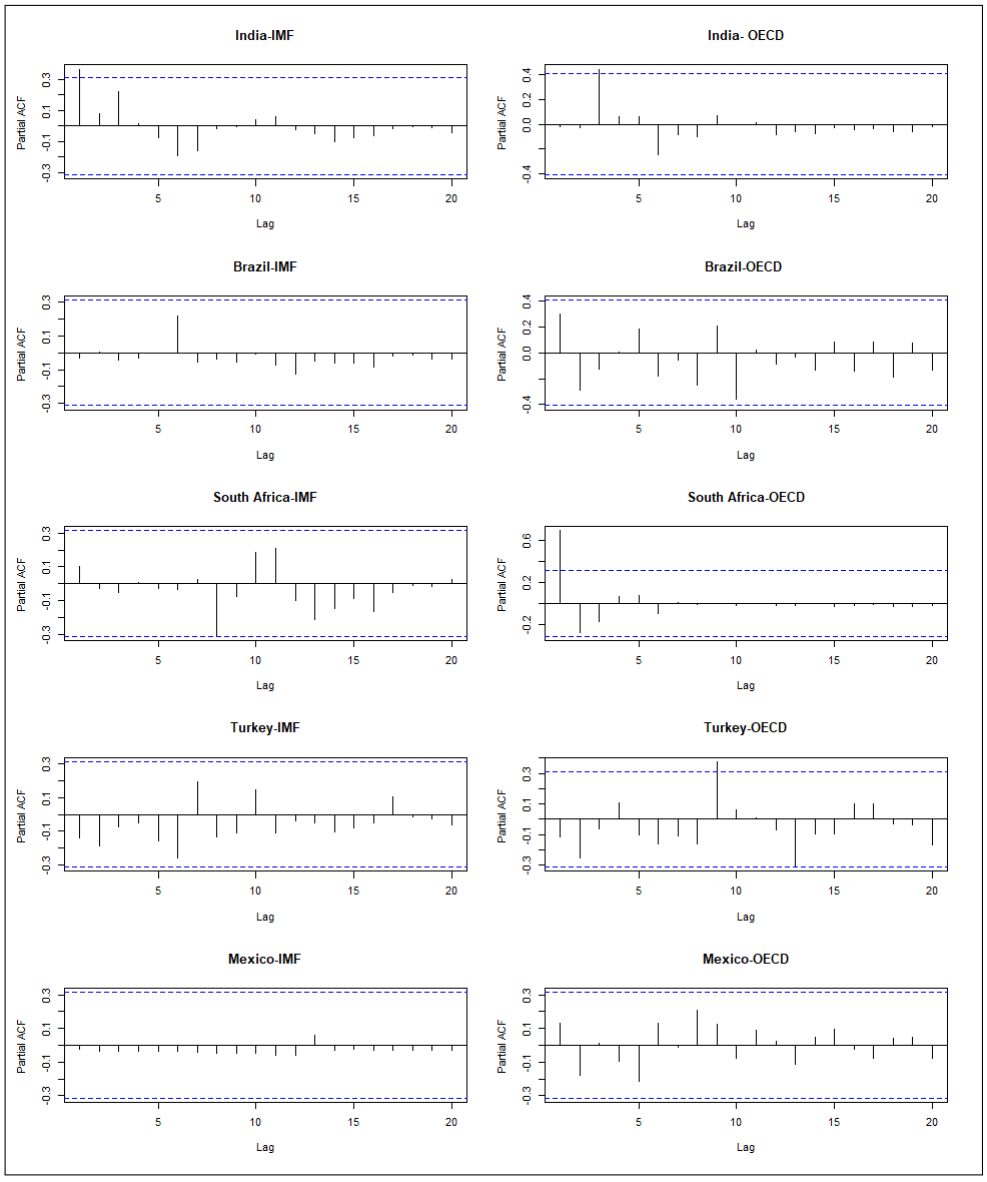


Figure 3.13: PACF plot for developing countries

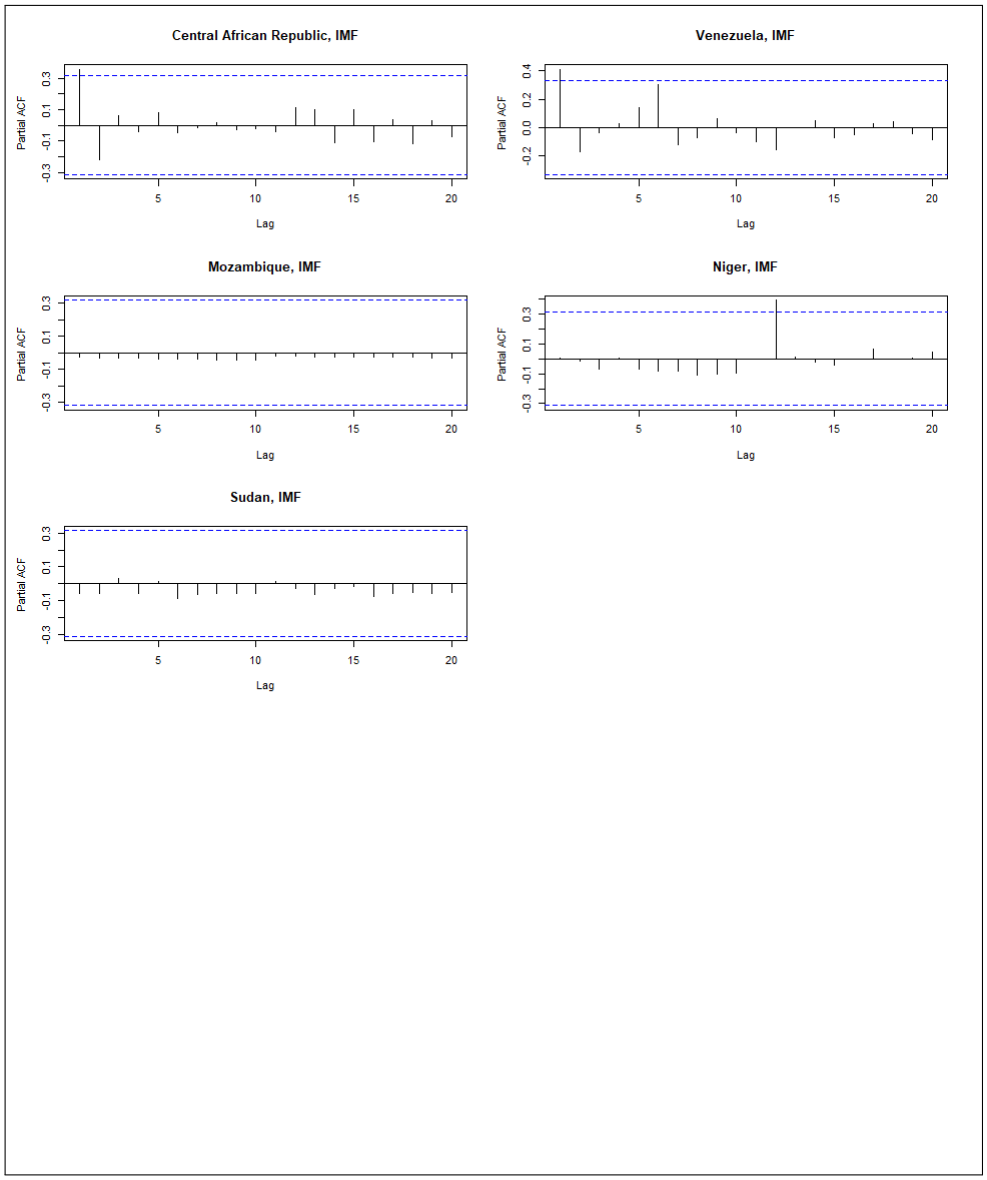
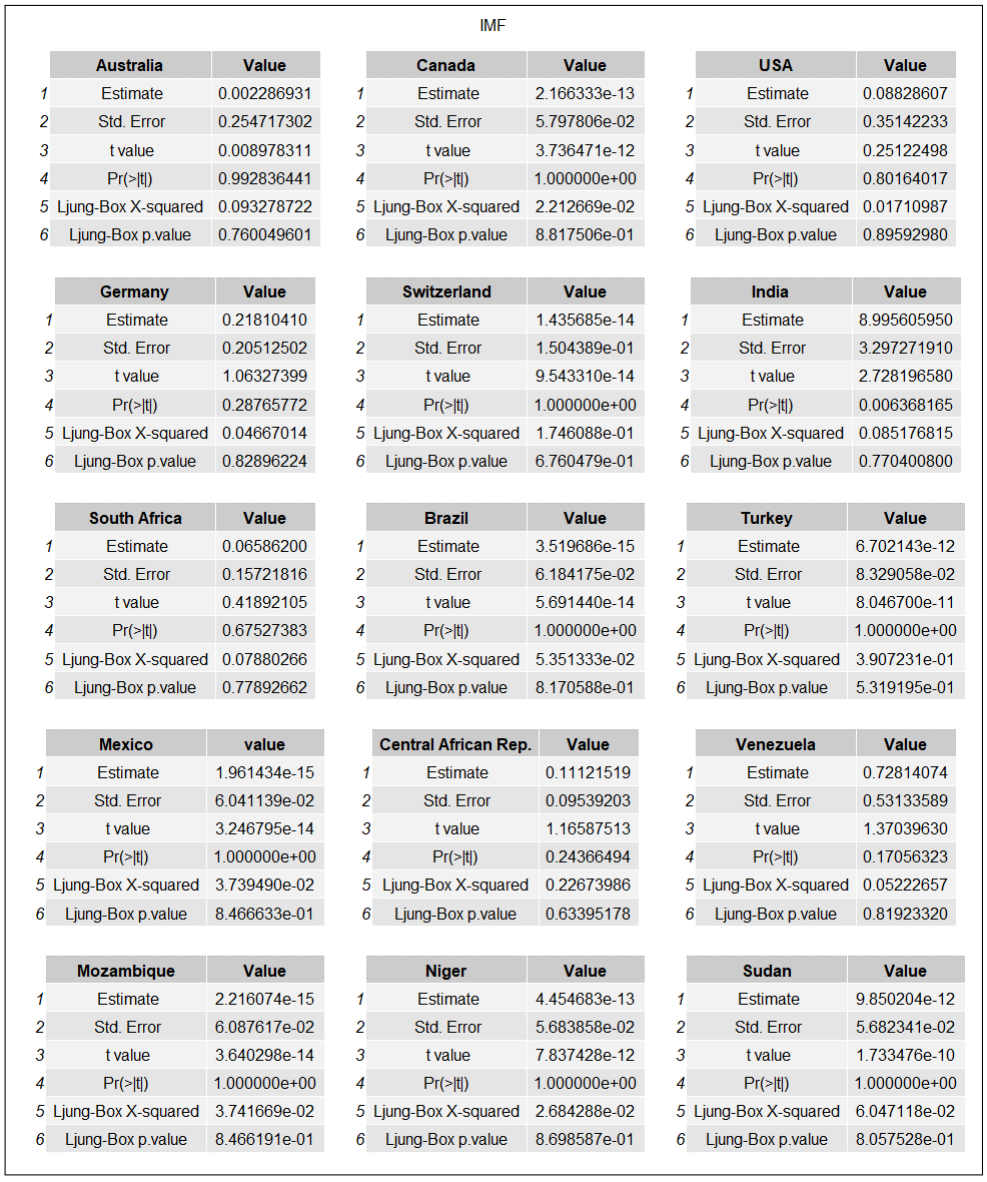


Figure 3.14: PACF plot for under-developed countries

On observing the ACF and PACF plots from figure 3.9 to figure 3.14, we infer that 95% of the autocorrelation coefficients are within the confidence interval. Therefore, there was no covariance which indicated the existence of white noise, and also there was no relevant amount of autocorrelation present in the time series data.

3.4.2 ARCH Model

The ARCH model determines whether the squared residuals/errors of a time series model exhibit autocorrelation or not. If autocorrelation is present then the time series exhibits conditional heteroskedasticity.



IMF

	Australia	Value
1	Estimate	0.002286931
2	Std. Error	0.254717302
3	t value	0.008978311
4	Pr(>|t|)	0.992836441
5	Ljung-Box X-squared	0.093278722
6	Ljung-Box p.value	0.760049601

	Canada	Value
1	Estimate	2.166333e-13
2	Std. Error	5.797806e-02
3	t value	3.736471e-12
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	2.212669e-02
6	Ljung-Box p.value	8.817506e-01

	USA	Value
1	Estimate	0.08828607
2	Std. Error	0.35142233
3	t value	0.25122498
4	Pr(>|t|)	0.80164017
5	Ljung-Box X-squared	0.01710987
6	Ljung-Box p.value	0.89592980

	Germany	Value
1	Estimate	0.21810410
2	Std. Error	0.20512502
3	t value	1.06327399
4	Pr(>|t|)	0.28765772
5	Ljung-Box X-squared	0.04667014
6	Ljung-Box p.value	0.82896224

	Switzerland	Value
1	Estimate	1.435685e-14
2	Std. Error	1.504389e-01
3	t value	9.543310e-14
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	1.746088e-01
6	Ljung-Box p.value	6.760479e-01

	India	Value
1	Estimate	8.995605950
2	Std. Error	3.297271910
3	t value	2.728196580
4	Pr(>|t|)	0.006368165
5	Ljung-Box X-squared	0.085176815
6	Ljung-Box p.value	0.770400800

	South Africa	Value
1	Estimate	0.06586200
2	Std. Error	0.15721816
3	t value	0.41892105
4	Pr(>|t|)	0.67527383
5	Ljung-Box X-squared	0.07880266
6	Ljung-Box p.value	0.77892662

	Brazil	Value
1	Estimate	3.519686e-15
2	Std. Error	6.184175e-02
3	t value	5.691440e-14
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	5.351333e-02
6	Ljung-Box p.value	8.170588e-01

	Turkey	Value
1	Estimate	6.702143e-12
2	Std. Error	8.329058e-02
3	t value	8.046700e-11
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	3.907231e-01
6	Ljung-Box p.value	5.319195e-01

	Mexico	value
1	Estimate	1.961434e-15
2	Std. Error	6.041139e-02
3	t value	3.246795e-14
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	3.739490e-02
6	Ljung-Box p.value	8.466633e-01

	Central African Rep.	Value
1	Estimate	0.11121519
2	Std. Error	0.09539203
3	t value	1.16587513
4	Pr(>|t|)	0.24366494
5	Ljung-Box X-squared	0.22673986
6	Ljung-Box p.value	0.63395178

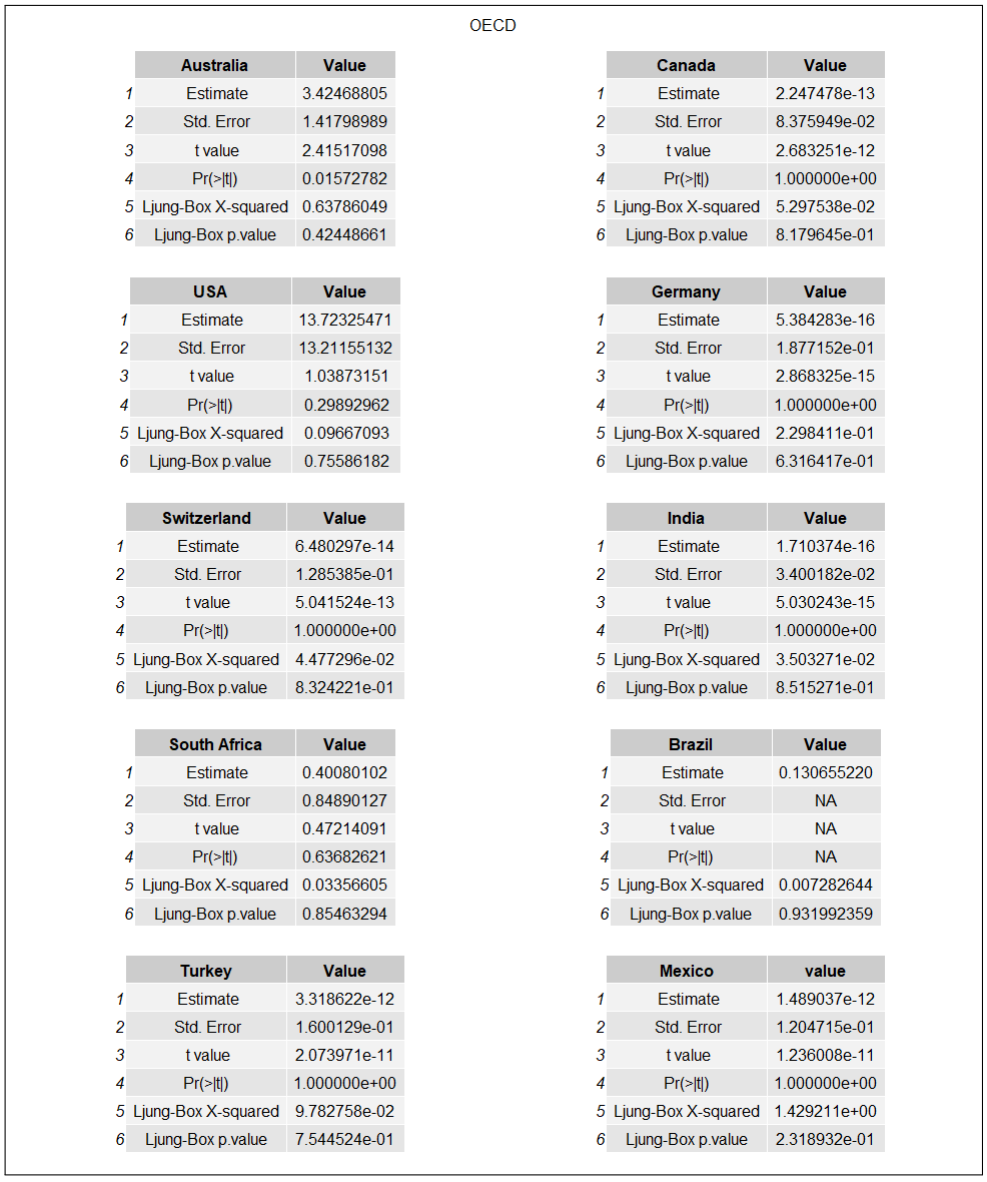
	Venezuela	Value
1	Estimate	0.72814074
2	Std. Error	0.53133589
3	t value	1.37039630
4	Pr(>|t|)	0.17056323
5	Ljung-Box X-squared	0.05222657
6	Ljung-Box p.value	0.81923320

	Mozambique	Value
1	Estimate	2.216074e-15
2	Std. Error	6.087617e-02
3	t value	3.640298e-14
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	3.741669e-02
6	Ljung-Box p.value	8.466191e-01

	Niger	Value
1	Estimate	4.454683e-13
2	Std. Error	5.683858e-02
3	t value	7.837428e-12
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	2.684288e-02
6	Ljung-Box p.value	8.698587e-01

	Sudan	Value
1	Estimate	9.850204e-12
2	Std. Error	5.682341e-02
3	t value	1.733476e-10
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	6.047118e-02
6	Ljung-Box p.value	8.057528e-01

Figure 3.15: Parameters obtained from the ARCH modeling of IMF data



OECD

	Australia	Value
1	Estimate	3.42468805
2	Std. Error	1.41798989
3	t value	2.41517098
4	Pr(>|t|)	0.01572782
5	Ljung-Box X-squared	0.63786049
6	Ljung-Box p.value	0.42448661

	Canada	Value
1	Estimate	2.247478e-13
2	Std. Error	8.375949e-02
3	t value	2.683251e-12
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	5.297538e-02
6	Ljung-Box p.value	8.179645e-01

	USA	Value
1	Estimate	13.72325471
2	Std. Error	13.21155132
3	t value	1.03873151
4	Pr(>|t|)	0.29892962
5	Ljung-Box X-squared	0.09667093
6	Ljung-Box p.value	0.75586182

	Germany	Value
1	Estimate	5.384283e-16
2	Std. Error	1.877152e-01
3	t value	2.868325e-15
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	2.298411e-01
6	Ljung-Box p.value	6.316417e-01

	Switzerland	Value
1	Estimate	6.480297e-14
2	Std. Error	1.285385e-01
3	t value	5.041524e-13
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	4.477296e-02
6	Ljung-Box p.value	8.324221e-01

	India	Value
1	Estimate	1.710374e-16
2	Std. Error	3.400182e-02
3	t value	5.030243e-15
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	3.503271e-02
6	Ljung-Box p.value	8.515271e-01

	South Africa	Value
1	Estimate	0.40080102
2	Std. Error	0.84890127
3	t value	0.47214091
4	Pr(>|t|)	0.63682621
5	Ljung-Box X-squared	0.03356605
6	Ljung-Box p.value	0.85463294

	Brazil	Value
1	Estimate	0.130655220
2	Std. Error	NA
3	t value	NA
4	Pr(>|t|)	NA
5	Ljung-Box X-squared	0.007282644
6	Ljung-Box p.value	0.931992359

	Turkey	Value
1	Estimate	3.318622e-12
2	Std. Error	1.600129e-01
3	t value	2.073971e-11
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	9.782758e-02
6	Ljung-Box p.value	7.544524e-01

	Mexico	value
1	Estimate	1.489037e-12
2	Std. Error	1.204715e-01
3	t value	1.236008e-11
4	Pr(>|t|)	1.000000e+00
5	Ljung-Box X-squared	1.429211e+00
6	Ljung-Box p.value	2.318932e-01

Figure 3.16: Parameters obtained from the ARCH modeling of OECD data

Ljung-Box test was one of the parameters returned in the ARCH tests. It is a type of statistical analysis which identifies the group of autocorrelations of a time series that are different from zero. In figures 3.15 and 3.16, the Ljung-Box test's p-value for each country was higher than "0.05". Therefore, we accepted the null hypothesis that the series exhibits conditional heteroskedastic behaviour. Furthermore, the series was applicable to GARCH tests.

As mentioned previously in the data analysis section, we only considered the IMF data because it had more countries.

3.4.3 Best order of fit

From the calculation of mean squared prediction errors for each country with ARCH, GARCH, AR, MA orders starting from (0,0,0,0) to (3,3,3,3), we obtained their smallest mean squared prediction error and order associated with it. We compiled the results as follows -

	ARCH	GARCH	AR	MA	Smallest.pred.err
Algeria	3	3	2	3	15.2391460
Antigua and Barbuda	1	3	3	3	0.0014093
Argentina	1	0	1	2	0.0161035
Australia	0	1	3	3	0.0105663
Austria	2	1	2	3	0.3009332
Bahamas, The	2	0	0	3	0.2483002

Figure 3.17: Smallest mean squared prediction errors

3.4.4 Clustering

Observations from the clustering plot in figure 3.18 depicted that the majority of countries have mean squared prediction errors approximately equal to zero. The mean of errors for the most massive cluster was "3.25." Eleven countries had errors with a mean of "73." Three countries had significant errors having a mean of "423.5." (Refer 3.19)

These obtained results indicate that the GARCH model produced during data analysis is an approximation of the prediction model used by IMF since a majority of countries taken into consideration have negligible errors in their fitted values.

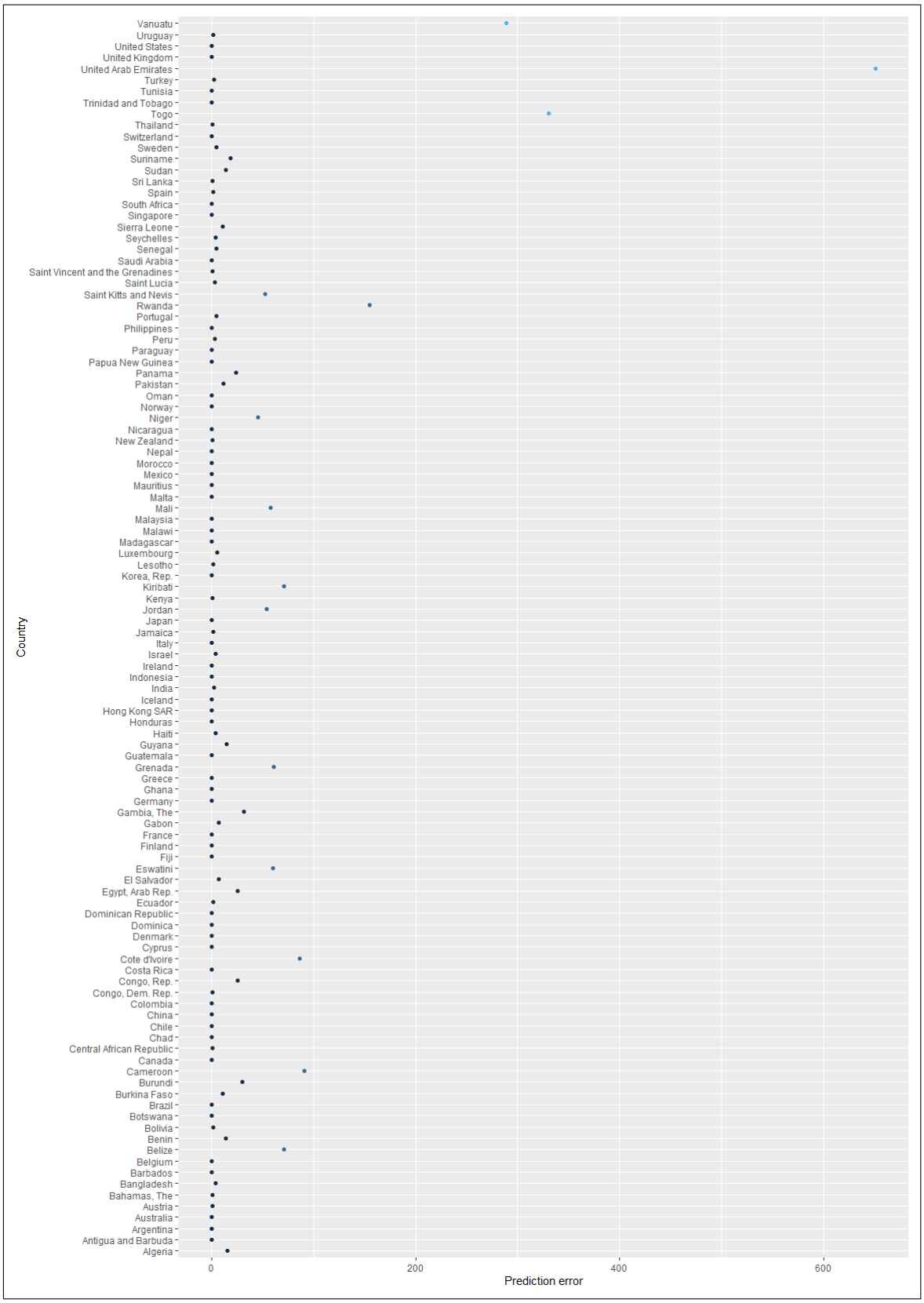
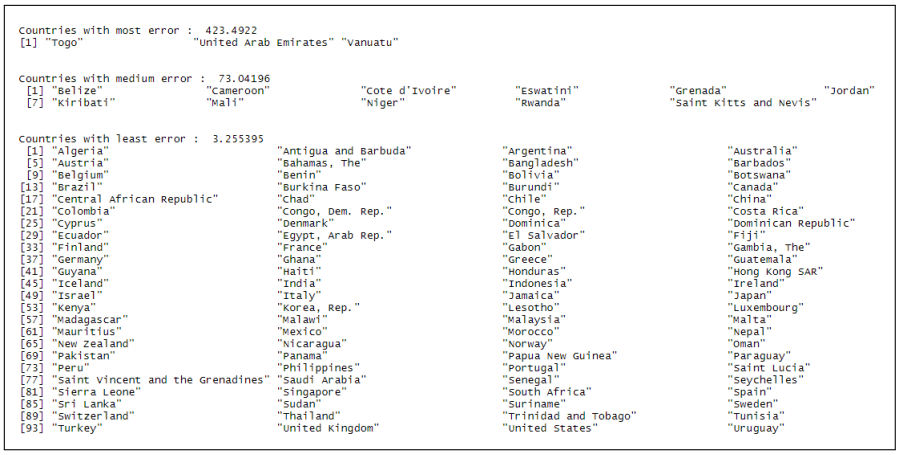


Figure 3.18: K-means clustering

Figure 3.19 shows the countries belonging to each cluster.



```
Countries with most error :  423.4922
[1] "Togo"                "United Arab Emirates" "Vanuatu"


Countries with medium error :  73.04196
 [1] "Belize"               "Cameroon"             "Cote d'Ivoire"        "Eswatini"             "Grenada"              "Jordan"
 [7] "Kiribati"             "Mali"                 "Niger"                "Rwanda"               "Saint Kitts and Nevis"


Countries with least error :  3.255395
 [1] "Algeria"                          "Antigua and Barbuda"              "Argentina"                        "Australia"
 [5] "Austria"                          "Bahamas, The"                     "Bangladesh"                       "Barbados"
 [9] "Belgium"                          "Benin"                            "Bolivia"                          "Botswana"
[13] "Brazil"                           "Burkina Faso"                     "Burundi"                          "Canada"
[17] "Central African Republic"         "Chad"                             "Chile"                            "China"
[21] "Colombia"                         "Congo, Dem. Rep."                 "Congo, Rep."                      "Costa Rica"
[25] "Cyprus"                           "Denmark"                          "Dominica"                         "Dominican Republic"
[29] "Ecuador"                          "Egypt, Arab Rep."                 "El Salvador"                      "Fiji"
[33] "Finland"                          "France"                           "Gabon"                            "Gambia, The"
[37] "Germany"                          "Ghana"                            "Greece"                           "Guatemala"
[41] "Guyana"                           "Haiti"                            "Honduras"                         "Hong Kong SAR"
[45] "Iceland"                          "India"                            "Indonesia"                        "Ireland"
[49] "Israel"                           "Italy"                            "Jamaica"                          "Japan"
[53] "Kenya"                            "Korea, Rep."                      "Lesotho"                          "Luxembourg"
[57] "Madagascar"                       "Malawi"                           "Malaysia"                         "Malta"
[61] "Mauritius"                        "Mexico"                           "Morocco"                          "Nepal"
[65] "New Zealand"                      "Nicaragua"                        "Norway"                           "Oman"
[69] "Pakistan"                         "Panama"                           "Papua New Guinea"                 "Paraguay"
[73] "Peru"                             "Philippines"                      "Portugal"                         "Saint Lucia"
[77] "Saint Vincent and the Grenadines" "Saudi Arabia"                     "Senegal"                          "Seychelles"
[81] "Sierra Leone"                     "Singapore"                        "South Africa"                     "Spain"
[85] "Sri Lanka"                        "Sudan"                            "Suriname"                         "Sweden"
[89] "Switzerland"                      "Thailand"                         "Trinidad and Tobago"              "Tunisia"
[93] "Turkey"                           "United Kingdom"                   "United States"                    "Uruguay"
```

Figure 3.19: Clusters of countries

Chapter 4

Conclusion

Both Spoken Tutorial scriptwriting and case study project on the analysis of the difference between the predicted and actual GDP have contributed to promoting the usage of R FLOSS. The newly created Spoken Tutorials scripts shall be a part of R tutorial series on Machine Learning. It will help AI enthusiasts in learning practical machine learning skills using R. The GDP analysis project can help various economists, statisticians and financial organizations as a reference when applying models for GDP forecasting. Scope of further research can be in refining the forecasting methods to minimize the error between predicted and actual GDP.

The entire FOSSEE fellowship experience was very informative and enjoyable. Every fellow learned new skills and methods which he/she can make use of in the future. Even though the fellowship was conducted remotely, it didn't hinder the experience and interactions between the fellows and instructors. Overall each fellow learned the different facets of working in an organization while contributing to the society.

References

[1] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "Uci repository of machine learning databases," 1998.

[2] W. N. Venables and B. D. Ripley, "Modern applied statistics with S-PLUS," *Statistics and Computing, Springer*, 2013.

[3] A. Kassambara and F. Mundt, "Factoextra: extract and visualize the results of multivariate data analyses," *R package version*, vol. 1, no. 5, pp. 337–354, 2017.

[4] H. Wickham, "ggplot2: elegant graphics for data analysis," *Statistics and Computing, Springer*, 2016.

[5] F. Leisch, "A toolbox for k-centroids cluster analysis," *Computational statistics & data analysis*, vol. 51, no. 2, pp. 526–544, 2006.

[6] M. Maechler, P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik, "cluster: Cluster Analysis Basics and Extensions (2019)," *R package version*, vol. 2, no. 0, 2017.

[7] W. B. Development Data Group, "World bank open data." `https://data.worldbank.org/`, 06 2020.

[8] W. Bank, "Gdp (current us $." `https://data.worldbank.org/indicator/NY.GDP.MKTP.CD`.

[9] IMF, "World economic outlook (april 2020)." `https://www.imf.org/external/datamapper/datasets/WEO`, 04 2020.

[10] OECD, "Oecd ilibrary | oecd economic outlook." `https://www.oecd-ilibrary.org/economics/oecd-economic-outlook_16097408`, 04 2020.

[11] IMF, "World economic outlook." `https://www.imf.org/external/datamapper/NGDP_RPCH@WEO/WEOWORLD`.

[12] OECD, "Gdp oecd data." `https://data.oecd.org/gdp/real-gdp-forecast.htm#indicator-chart`.

[13] A. Kassambara, "ggpubr:"ggplot2" based publication ready plots," *R package version 0.1*, vol. 7, 2018.

[14] H. Wickham, R. Francois, L. Henry, K. Müller, *et al.*, "dplyr: A grammar of data manipulation," *R package version 0.4*, vol. 3, 2015.

[15] R. C. Team *et al.*, "R: A language and environment for statistical computing," 2013.

[16] A. Trapletti, K. Hornik, and B. LeBaron, "tseries: Time series analysis and computational finance," *R package version 0.10-11*, 2007.

[17] B. Auguie and A. Antonov, "gridExtra: miscellaneous functions for "grid" graphics," *R package version*, vol. 2, no. 601, p. 602, 2017.

[18] A. Ghalanos, "rugarch: Univariate garch models, r package version 1.3-3," 2014.

[19] R. J. Hyndman and G. Athanasopoulos, "Forecasting: principles and practice," 2018.

[20] A. V. Metcalfe and P. S. Cowpertwait, "Introductory time series with R," *Statistics and Computing, Springer*, 2009.

[21] A. Kassambara, "Machine Learning Essentials: Practical Guide in R," 2018.

[22] E. Holmes, M. Scheuerell, and E. Ward, "Applied time series analysis for fisheries and environmental data," vol. 2725, 2019.