Forecasting Real Gross Domestic Product (GDP) per capita using ARIMA model

Hassan OUKHOUYA

November 26th, 2021

Contents

About me

- ***Hassan OUKHOUYA***
- ***E-mail***: hassan.oukhouya@um5r.ac.ma
- ***LinkedIn***: https://www.linkedin.com/in/hassan-oukhouya-3901b816b/
- ***ORCID iD***: https://orcid.org/0000-0002-5058-2008
- ***Upwork***: https://www.upwork.com/services/product/time-series-analysis-with-python-or-r-studio-1449669530698514432?ref=project_share

1 Overview

Time series analysis and dynamic modeling are important to study topics with a variety of applications in business, economics, finance, and computer science. The purpose of time series analysis is to examine the

projected path of time series observations, build a model to characterize the data structure, and then predict the time series' future values. In economics, Real GDP Per Capita serves to reflect whether the nation is progressing or declining in economic growth. An increase in Real GDP signals that the economy is doing well, given that products and services are growing in value, while a decrease signals the opposite. Forecasting of GDP of production industries plays a major role in the optimal decision formulas for government and industry in the United States (US). In this project, we will forecast whether the country's economy is expanding or contracting due to its output on the factors of production. We can use `Quandl` to obtain data going back to 1947 for the Real gross domestic product per capita, Federal Reserve Bank of St. Louis uses the symbol "rxVQhZ8_nxxeo2yy4Uz4" (Quandle code : FRED/GDPC1). We will use historical values of GDP per capita for the US, **from January 01, 1947 to July 01, 2021**, for all four quarters per year. The GDP data has been seasonally adjusted. The quarters were collected (299 observations). The first 267 are to be used as training data, while the others are to be used as a test set. A train and test set was created. The range was as follows:

Training Set Range: **01-01-1947 - 01-10-2013**

Test Set Range: **01-01-2014 - 01-07-2021**

1.1 Research question

The main research question investigated in this study is:

How to model and forecast Real Gross Domestic Product per Capita using ARIMA model?

1.2 Objectives

The contributions of this work can therefore be resumed as follows:

- 1^{st} Analysis of the GDP series.
- 2^{nd} Modelling and calibrating the GDP series.
- 3^{th} Forecasting the GDP series.

2 Statistical Background

2.1 ARIMA Models

ARMA models can be extended to non-stationary series by permitting the data series to be differencing, resulting in ARIMA models. $ARIMA(p, d, q)$ is a general non-seasonal model with three parameters: p is the autoregressive order, d is the degree of differencing, and q is the moving-average order. The $ARIMA(p, d, q)$ model is mathematically expressed using lag polynomials as follows:

$$\phi_p(B)(1-B)^d Y_t = \theta_q(B)\varepsilon_t, \quad \Big(1 - \sum_{i=1}^{p} \phi_i B^i\Big)(1-B)^d Y_t = \Big(1 + \sum_{j=1}^{q} \theta_j B^j\Big)\varepsilon_t$$

2.2 Box-Jenkins Approach

The Box-Jenkins (1970) methodology to time series analysis, named after statisticians George Box and Gwilym Jenkins, uses ARIMA models to identify the best fit of a time series model to previous values of a time series. Figure below shows the four iterative stages of modeling according this approach.

- **Model identification**: Assuring that the variables are stationary, identifying seasonality in the series, and identifying which autoregressive or moving average component should be implemented in the model using the plots of the Auto- Correlation Function (ACF) and Partial Auto-Correlation Function (PACF) of the series.

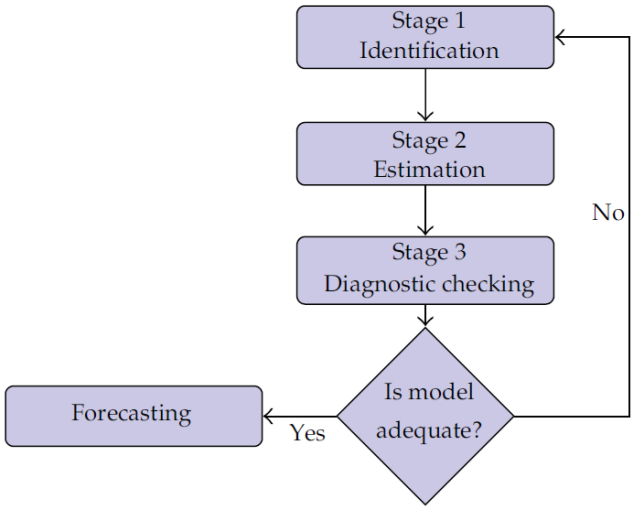


Figure 1: Stages in the Box-Jenkins iterative approach

- **Model estimation**: Using computing algorithms, find the coefficients that best fit the ARIMA model you've chosen. Maximum Likelihood Estimate (MLE) or non-linear least-squares estimation are the most frequent methods.
- **Model checking**: By checking if the estimated model satisfies the criteria of a stationary univariate process. The residuals should be independent of one another and constant in mean and variance over time; visualizing the residuals' ACF and PACF can help identify misspecification. If the estimation isn't good enough, we'll have to go back to step one and try again. Additionally, the estimated model should be compared to different ARIMA models in order to determine which model is appropriate for the data. The two most popular model selection criteria are Akaike's Information Criterion (AIC) and Bayesian Information Criteria (BIC), both of which are defined as follows:

$$AIC = 2m - 2\ln(\hat{L})$$
$$BIC = \ln(n)m - 2\ln(\hat{L})$$

where $\hat{L}$ denotes the maximum value of the likelihood function for the model, m is the number of parameters estimated by the model, and n is the number of observations (sample size). Practically, AIC and BIC are used with the classical criterion: the Mean Squared Error (MSE).

- **Forecasting**: We can utilize the selected ARIMA model for forecasting if it conforms to the criteria of a stationary univariate process.

2.3 Important librarys

The packages being used in this study series are here in listed:

```
## Load Packages
library(zoo,warn.conflicts=FALSE)
library(lubridate,warn.conflicts=FALSE)
library(mgcv,warn.conflicts=FALSE)
library(rugarch,warn.conflicts=FALSE)
# visualization
suppressPackageStartupMessages(library(ggplot2))
# ARMA modeling
```

```
suppressPackageStartupMessages(library(forecast))
# structural changes
suppressPackageStartupMessages(library(strucchange))
# ARMA order identification
suppressPackageStartupMessages(library(TSA))
library(Metrics)
library(tseries)
library(timeSeries)
library(xts)
library(forecast)
library(pastecs)
library(tidyr)
library(dplyr)
library(splines)
library(tidyverse)
rm(list=ls())
library(FinTS)
library(rugarch)
library(dynlm)
library(vars)
library(nlWaldTest)
library(broom)
library(readxl)
# Getting  data
library(Quandl)
library(zoo)
library(xts)
library(dygraphs)
library(knitr)
library(urca)
library(vars)
```

3 Collection data

- **Source**: U.S. Bureau of Economic Analysis
- **Release**: Gross Domestic Product
- **Units**: Chained 2012 Dollars, Seasonally Adjusted Annual Rate
- **Frequency**: Quarterly
- BEA Account Code: A939RX
- For more information about this series, please visit the Bureau of Economic Analysis.
- **Suggested Citation**: U.S. Bureau of Economic Analysis, Real gross domestic product per capita [A939RX0Q048SBEA], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/A939RX0Q048SBEA, November 22, 2021.

3.1 Importing and cleaning data

```
#importing the data
GDP <- read.csv("GDP.csv", head = TRUE)
```

```
#cleaning the data

#dates to date format
GDP$Date<-as.Date(GDP$Date,format='%Y/%m/%d')


#prices to timeseries format
GDP_US <- ts(GDP$GDP_per_capita,start=c(1947,1,1),freq=4)
```

Another way to get data from Federal Reserve Economic Data

Data for the quarterly Real GDP is imported from Quandl website (Quandle code : `FRED/GDPC1`).

```
#Quandl.api_key("A939RX0Q048SBEA")
#rGDP <- Quandl("FRED/GDPC1", type="ts" ) # or rGDP <- Quandl("FRED/GDPC1", type="zoo")
#str(rGDP)
```

3.2 Data structure

```
str(GDP_US)
```

```
##  Time-Series [1:299] from 1947 to 2022: 14213 14111 14018 14171 14326 14505 14525 14474 14212 14107
```

```
dim(GDP_US)
```

```
## NULL
```

```
frequency(GDP_US)
```

```
## [1] 4
```

```
head(GDP_US)
```

```
##       Qtr1  Qtr2  Qtr3  Qtr4
## 1947 14213 14111 14018 14171
## 1948 14326 14505
```

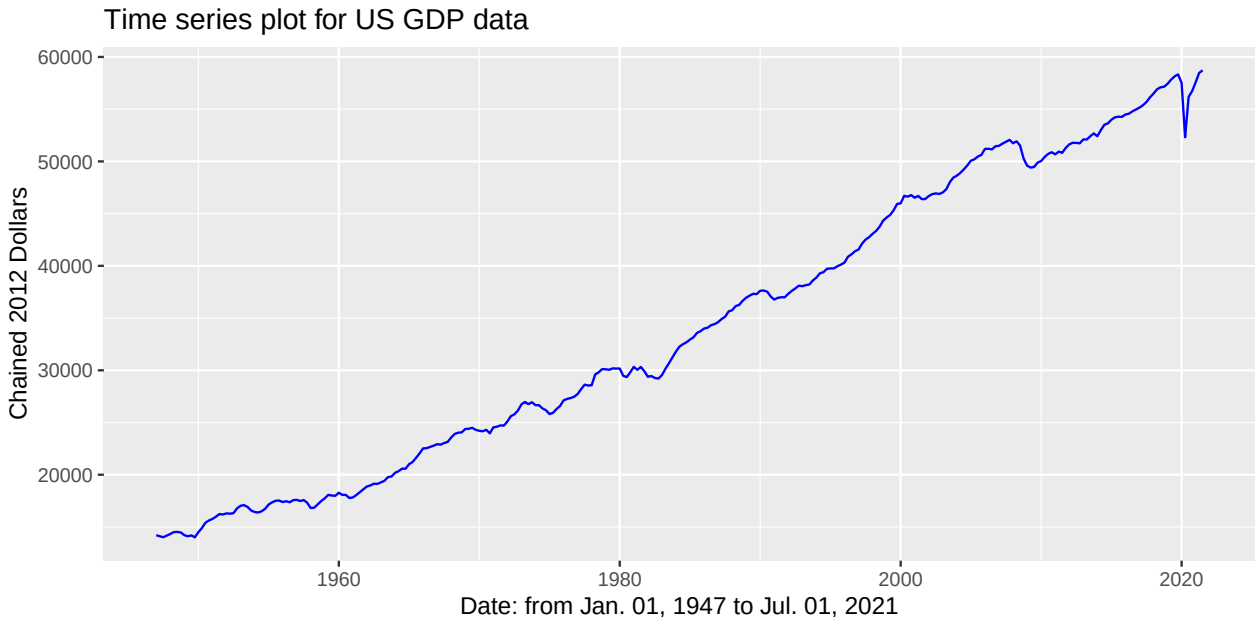
```
tail(GDP_US)
```

```
##       Qtr1  Qtr2  Qtr3  Qtr4
## 2020       52314 56182 56732
## 2021 57568 58478 58717
```

4 Exploratory Data Analysis

4.1 Visualize the Time Series

```
autoplot(GDP_US,col="blue" ,main = 'Time series plot for US GDP data',xlab=
           'Date: from Jan. 01, 1947 to Jul. 01, 2021', ylab='Chained 2012 Dollars')
```



The figure above represents the series. The first observations we can take from this series are the following:

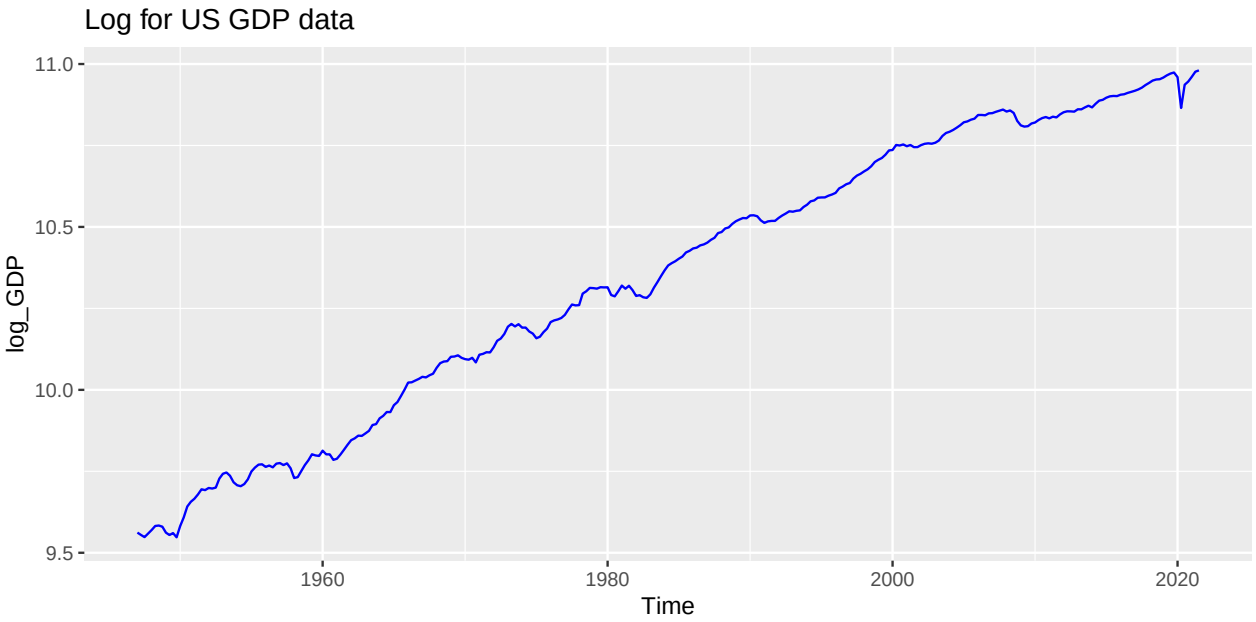
- The series for Real GDP shows the upward sloping trend.
- We can use the log of the series to remove this variation of amplitude in the series.
- The series is not stationary, the variance is not constant over time.
- We can see a brutal fall of GDP at the beginning of 2020 in the COVID-19 crisis.

4.2 Characteristics of the series

Some remarks were made at the first observation of the series, however it would be advisable to check these apprehensions and to pose certain assumptions if necessary so as to be able to determine all the characteristics of the series.

We move to the data log to be able to reduce the variations over time.

```
log_GDP= log(GDP_US)
autoplot(log_GDP, col="blue", main="Log for US GDP data")
```



By observing the log of the series, the variation has not totally disappeared, moreover the trend now presents a certain curve. By switching to the log, the trend problem has not been solved satisfactorily, the more linear it is, the better we can observe it globally.

4.2.1 Test of stationarity of the series

Augmented Dickey-Fuller Test

The Dickey-Fuller test, performs a stationarity test of the series with the null hypothesis (H_0) that the series is not stationary with an error level of 5%.

```
adf.test(GDP_US)
```

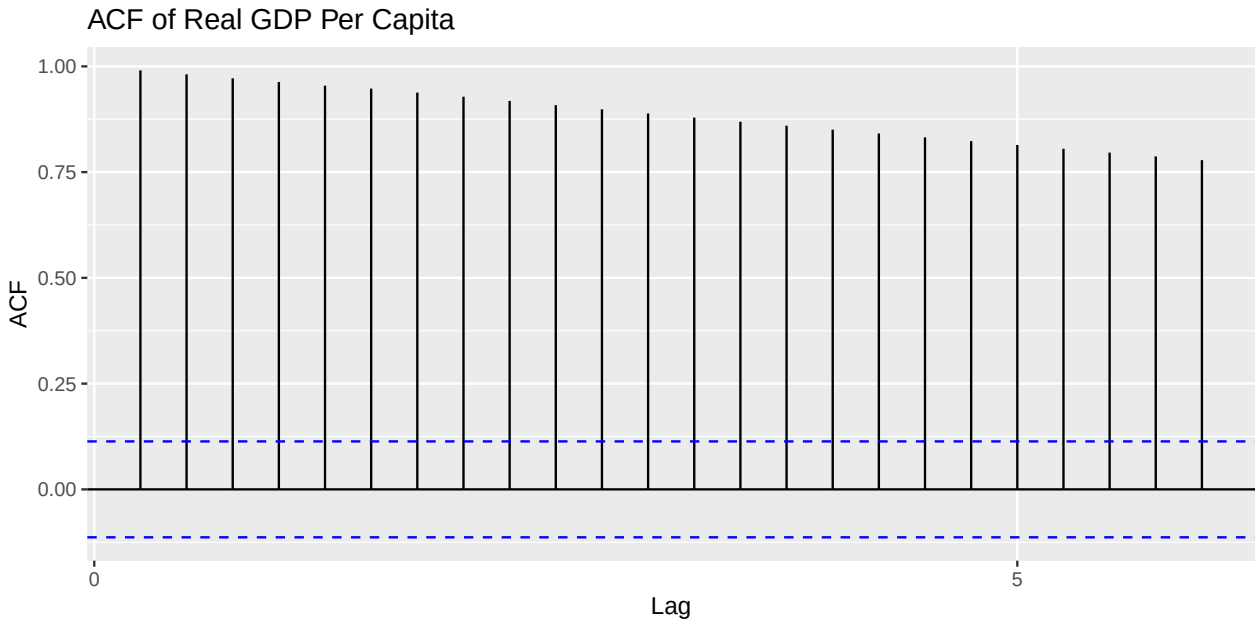
```
##
##  Augmented Dickey-Fuller Test
##
## data:  GDP_US
## Dickey-Fuller = -2.7362, Lag order = 6, p-value = 0.266
## alternative hypothesis: stationary
```

We do not reject the null hypothesis, the `p-value` being higher than the 5% level, the series is not stationary.

Hypothesis of stationarity by the autocorrelogram (ACF)

This hypothesis can be confirmed or refuted by observing the autocorrelogram of the series. In Indeed, a stationary series has an autocorrelogram which decreases in an exponential way to 0.

```
autoplot(acf(GDP_US, plot = FALSE),main="ACF of Real GDP Per Capita")
```

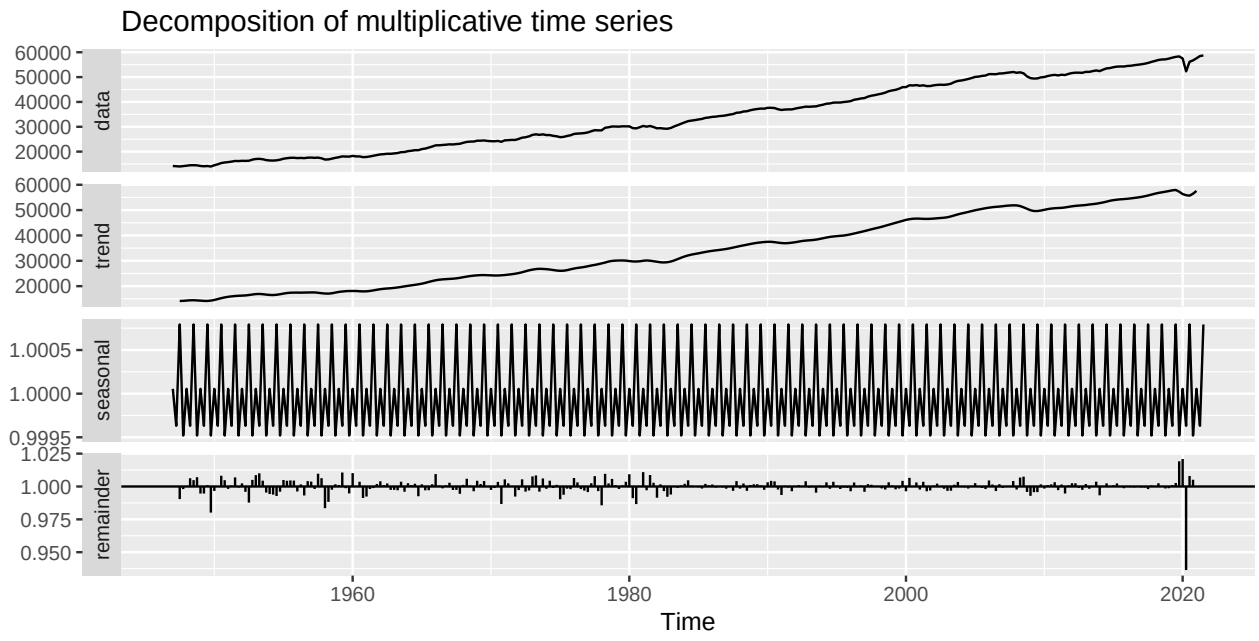


The autocorrellogram does not decrease exponentially towards 0, but we observe large spikes at lags 1, 12, 24.

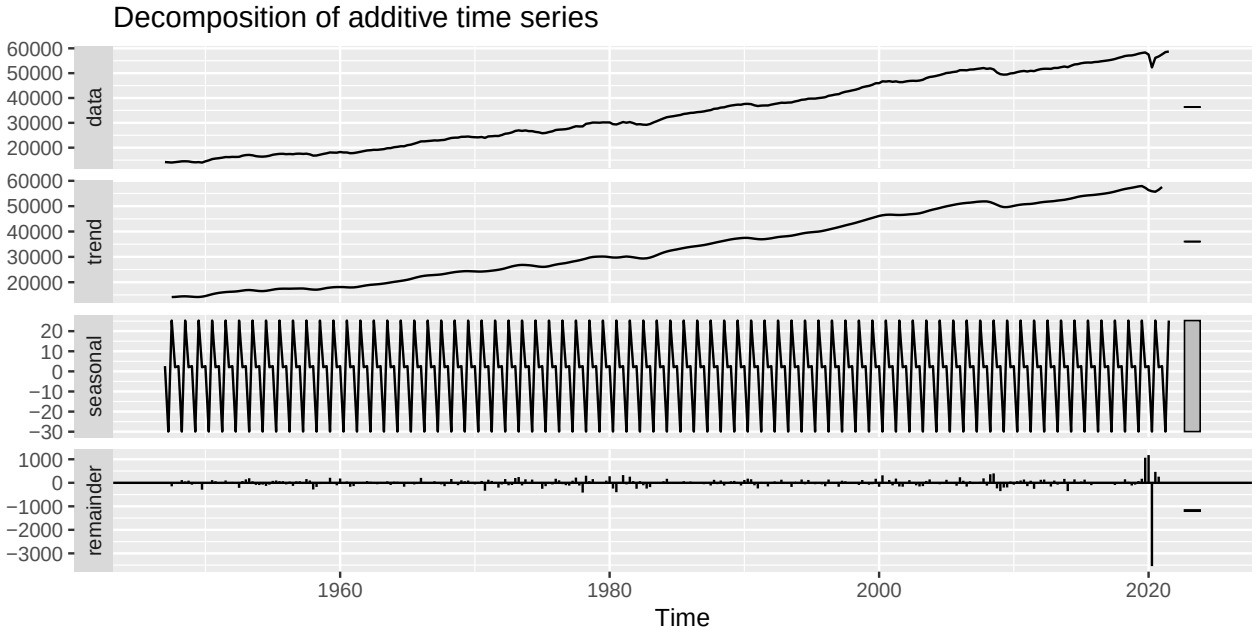
Decomposition of the series

We have observed at first view that the series is multiplicative, because of the increase of the trend in time, and the horizontal form of trend from a certain time. Let us observe its multiplicative decomposition and compare it with its additive decomposition in the figure below.

```
autoplot(decompose(GDP_US, type = "multiplicative"))
```




```
autoplot(decompose(GDP_US, type = "additive"))
```



The decomposition into multiplicative and additive only varies on the residuals, the trend and seasonality remain relatively the same for both decomposition methods. In the following we can therefore assume that the series is additive.

Conclusion

In summary, the characteristics of the series are as follows:

- Non-stationary
- Additive
- Increasing trend
- Non-seasonal

5 Modeling and calibration of the GDP series using ARIMA Models

The previous section gave us a conclusion about a non-stationary series. The modeling consists in finding the time series model that will best reproduce the series. The logical choice of this model for the estimation of the series is an $ARIMA(p, d, q)$ model. The coefficients of this model are also provided by the characteristics of the series. The choice of these coefficients represents the main difficulty, however we can observe different transformations of the series in order to orient our different choices.

The good model will be the one whose coefficients provide a white noise as residuals. The residuals must follow a normal distribution with zero mean, constant variance and no autocorrelation. The objective of modeling the series is to be able to predict a certain number of future values with a minimum margin of error.

5.1 Model identification

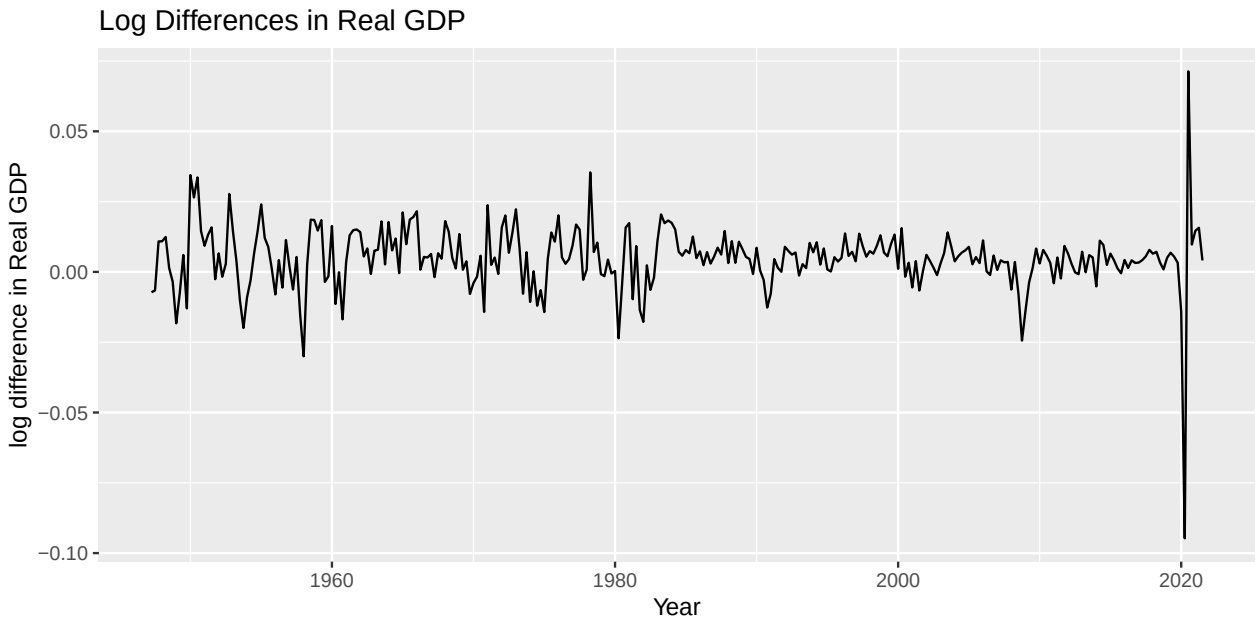
5.1.1 Stationarization of the series

The first essential step is the stationarization of the series. There are several ways to do this, but the most important is to choose a stationary series that would help to choose the right coefficients. We choose a stationary series by degree 1 differentiation, to remove the trend and seasonality if it exists.

Construct the time series with log changes in Real GDP (Y_t) and denote the variable name for Y_t by `difflogGDP`.

$$Y_t = \nabla \log GDP = \log GDP_t - \log GDP_{t-1}$$

```
difflogGDP <- diff(log_GDP)
autoplot(difflogGDP,  xlab="Year", ylab=
           "log difference in Real GDP",  main="Log Differences in Real GDP")
```



Detecting time series outliers

The `tsoutliers()` function in the forecast package for R is useful for identifying anomalies in a time series.

```
tsoutliers(difflogGDP)
```

```
## $index
## [1]  44 293 294
##
## $replacements
## [1] -0.001808288 -0.005406909  0.001318837
```

```
tsoutliers(GDP_US)
```

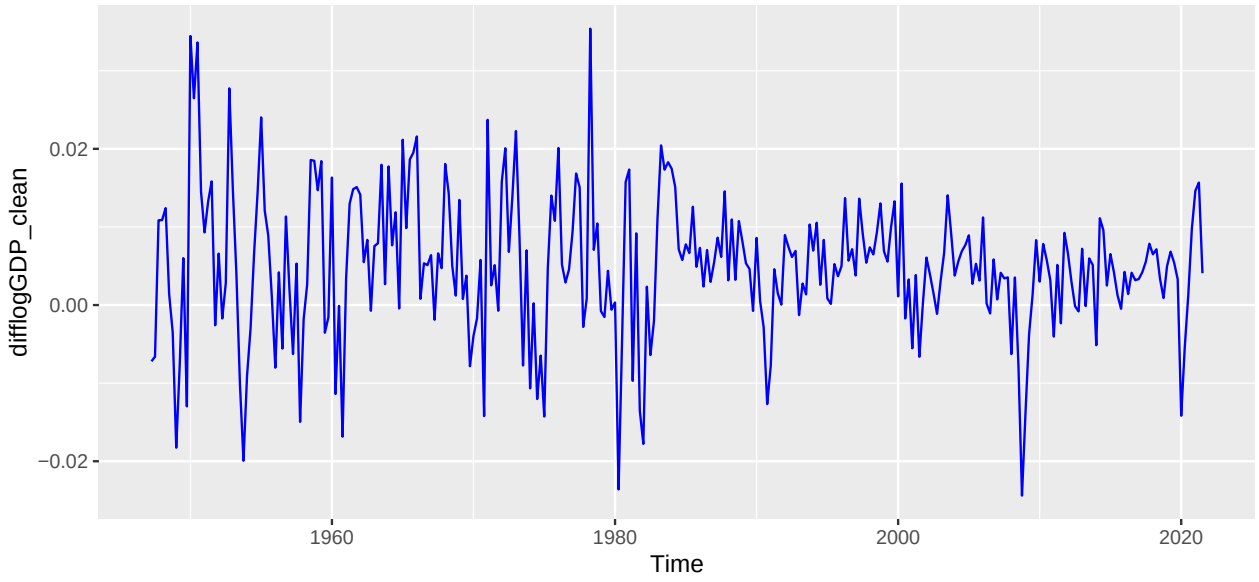
```
## $index
## [1] 294
##
## $replacements
## [1] 56849.13
```

The `tsclean()` function removes outliers identified in this way, and replaces them (and any missing values) with linearly interpolated replacements.

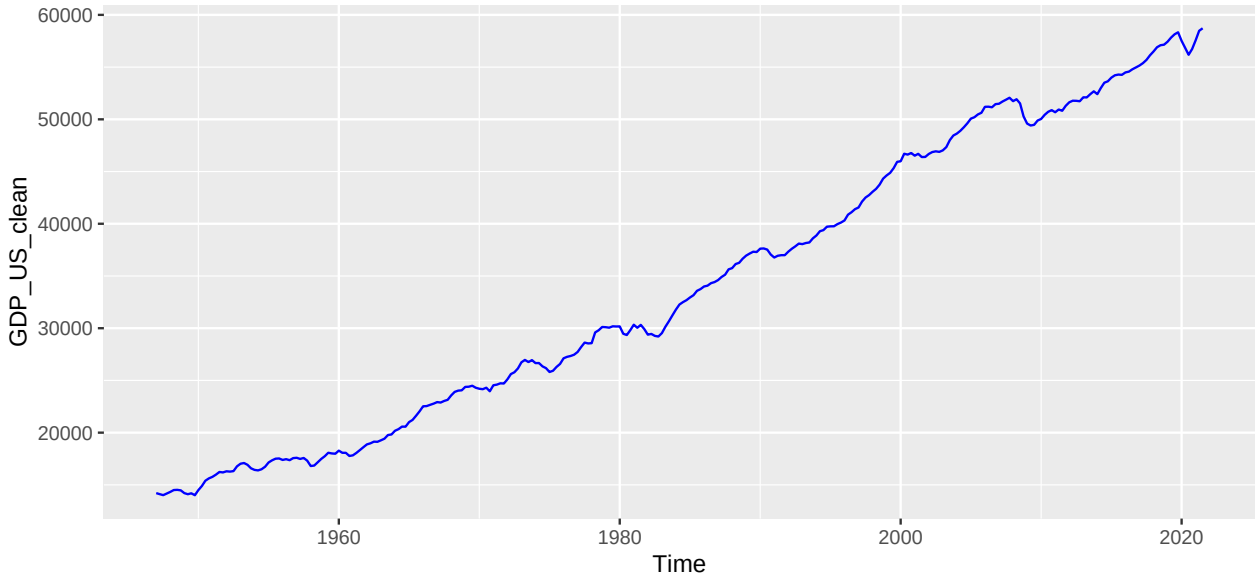
Note: Any outliers identified in this manner are replaced with linearly interpolated values using the neighbouring observations, and the process is repeated.

```
difflogGDP_clean = tsclean(difflogGDP)
GDP_US_clean = tsclean(GDP_US)
```

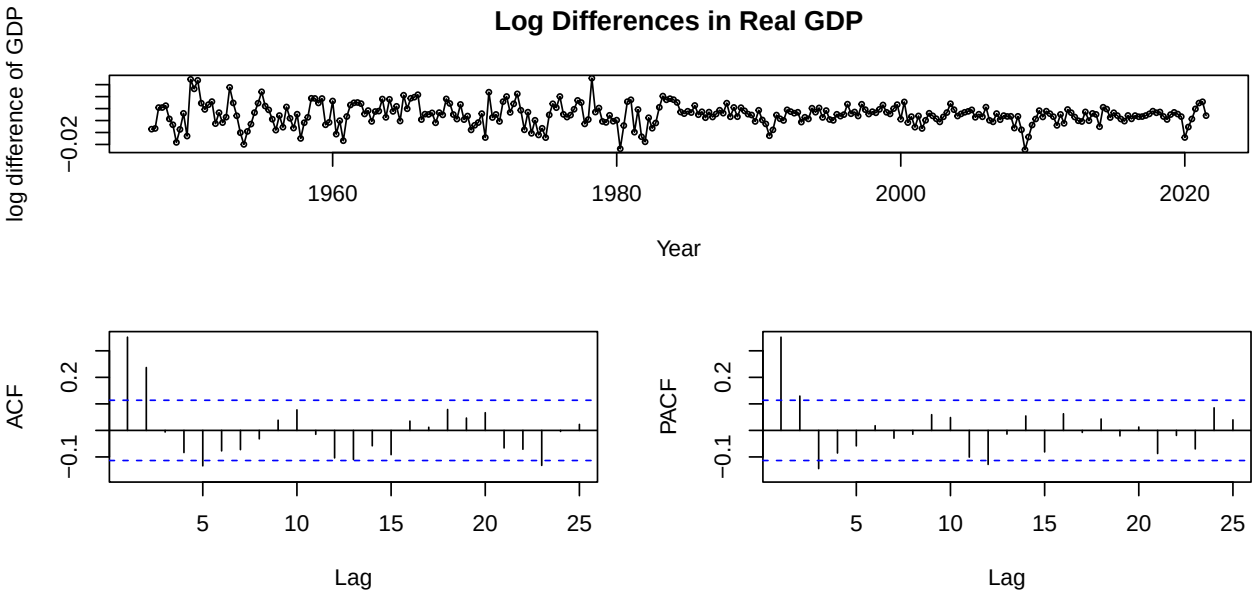
```
autoplot(difflogGDP_clean, type="l", col="blue" )
```




```
par(new=TRUE)
autoplot(GDP_US_clean, type="l", col="blue" )
```




```
tsdisplay(difflogGDP_clean,  xlab="Year", ylab="log difference of GDP",  main=
            "Log Differences in Real GDP")
```



After the transformation, the graph shows that the volatility of real GDP decreases over time until a very significant peak in early 2020 due to the COVID-19 crisis and The autocorrelogram of the series decreases in an exponnential way, which indicates that the series is stationary and can be verified by the ADF test.

```
adf.test(difflogGDP_clean)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  difflogGDP_clean
## Dickey-Fuller = -6.9726, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

With the confirmation of the Augmented Dickey-Fuller test (`p-value` lower than 5%) we obtain a stationary series.

Phillips-Perron Test

```
pp.test(difflogGDP_clean,alternative="stationary")
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  difflogGDP_clean
## Dickey-Fuller Z(alpha) = -193.87, Truncation lag parameter = 5, p-value
## = 0.01
## alternative hypothesis: stationary
```

The results of the PP test indicate that the GDP series is stationary.

In the next step, we fixed a breakpoint which will be used to split the series dataset in two parts; training (90%) and test (10%).

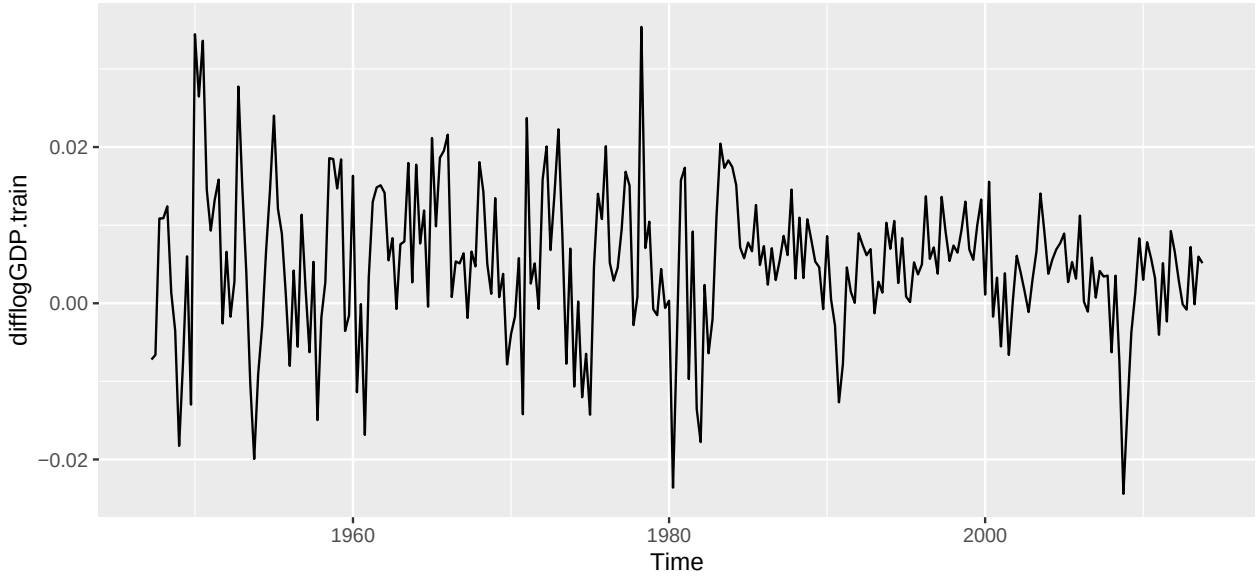
5.1.2 Split the log difference series GDP into training and test

```
# Delimit testing range (10%)
fstQ <- 1947.10 # 1947Q1
```

```r
# Delimit training range (90%)
lstQ <- 2013.90 # 2014Q1
difflogGDP.train <- window(difflogGDP_clean, end=lstQ)
difflogGDP.test <- window(difflogGDP_clean, start=lstQ+0.10)
```

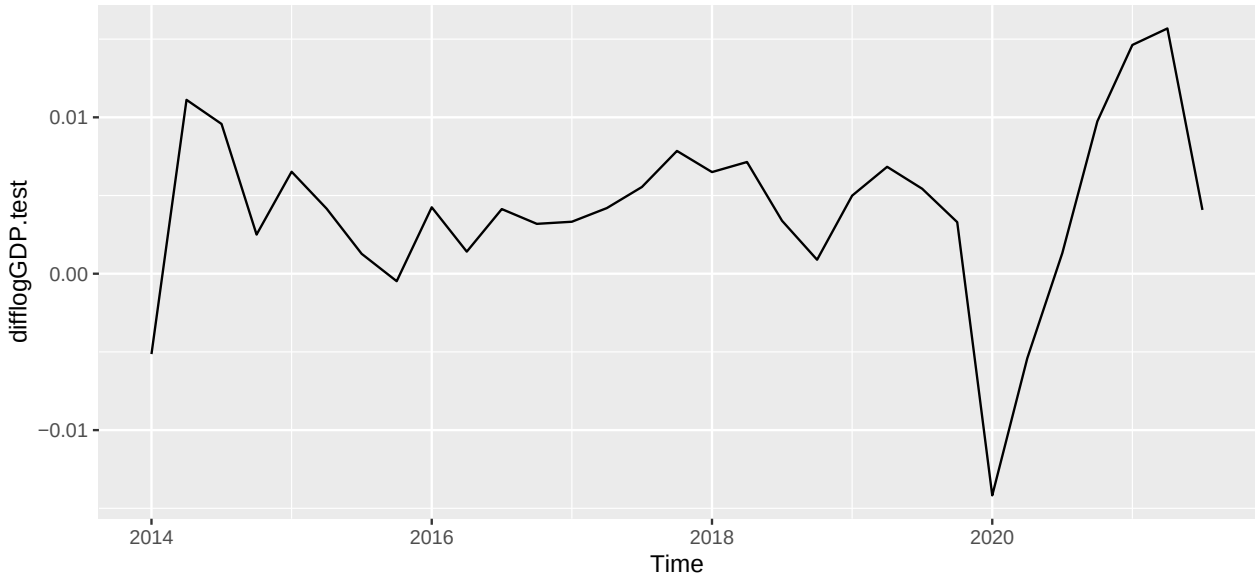
Plot of the training set

```r
autoplot(difflogGDP.train)
```



Plot of the test set

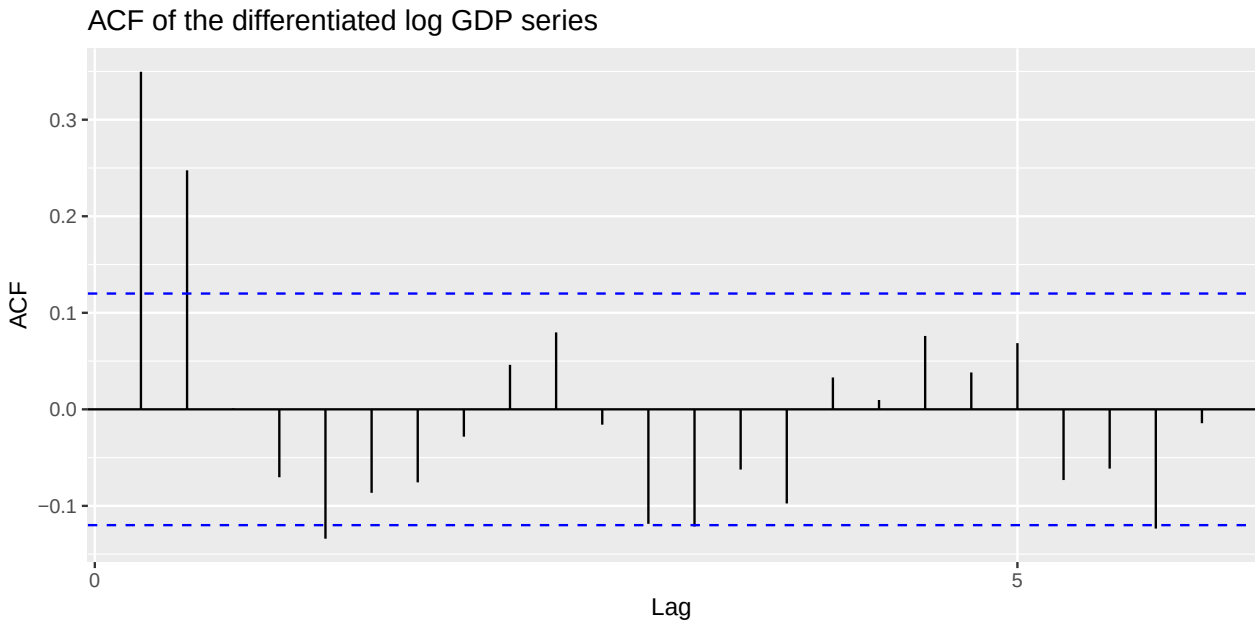
```r
autoplot(difflogGDP.test)
```



5.1.3 Choice of parameters p,d,q

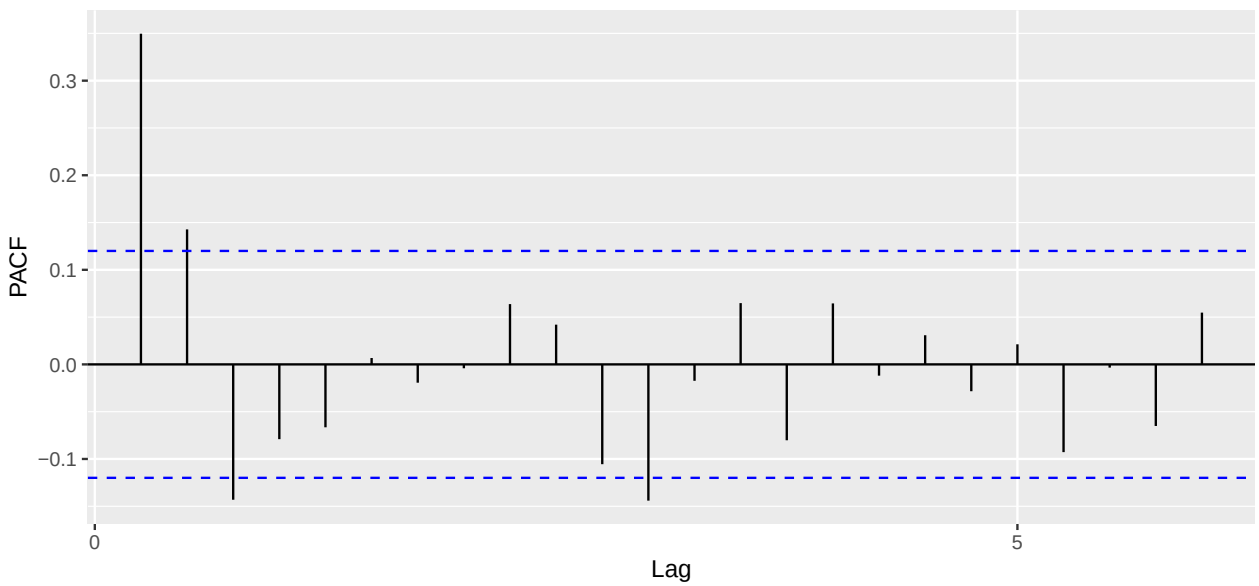
The choice of the parameters (p, d, q), are essentially based on the ACF and the PACF.

```
autoplot(acf(difflogGDP.train, plot = FALSE),main=
           "ACF of the differentiated log GDP series")
```



To choose the parameters p, d, q that represent the modeling of the non-seasonal part, we observe the ACF (autocorrelogram) and PACF (partial autocorrelogram) from lag 1. The appreciation of the ACF gives the choice of q. We choose q in such a way as to be able to take into account the successive autocorrelations which go out of the acceptance zone (the blue dashed lines on the figure). By observation of the ACF (above), the first two autocorrelations are very significant, especially the first and the second, so the candidates are $q = 1, 2$.

```
autoplot(pacf(difflogGDP.train, plot = FALSE),main=
           "PACF of the differentiated log GDP series")
```



As for p, it is chosen on the same principle as q but applied to PACF. We observe the first partial autocorrelations, i.e. from 1, and we take the coefficient which takes into account the spikes which

are out of the acceptance zone (partial autocorrelation is null for the corresponding lag). Possible candidates for p are $p = 1, 2$, $p = 3$.

The parameter d represents the degree of differentiation of the non-seasonal part, since the series has been differentiated only once, i.e. over the period, it is assumed that the non-seasonal part has not been differentiated. therefore we choose $d = 1$.

5.1.4 Split the series GDP into training and test

```
# Delimit testing range (10%)
trainQ <- 1947.10 # 1947Q1
# Delimit training range (90%)
testQ <- 2013.90 # 2013Q2
GDP_US.train <- window(GDP_US_clean, end=testQ)
GDP_US.test <- window(GDP_US_clean, start=testQ+0.10)
```

With the parameters in hand, we can now try to build ARIMA model. The value found in the previous section might be an approximate estimate and we need to explore more (p, d, q) combinations which can also be done using the `auto.arima` function which is explored later.The one with the lowest BIC and AIC would be our choice.

We apply the procedure `auto.arima` available in the package `forecast`. We use the argument `stepwise=FALSE` for exploring the whole identification structure possibilities:

```
#Checking using auto.arima
ARIMA<-auto.arima((GDP_US.train), seasonal=FALSE)
ARIMA
```

```
## Series: (GDP_US.train)
## ARIMA(2,1,0) with drift
##
## Coefficients:
##          ar1     ar2     drift
##       0.3098  0.1508  143.6201
## s.e.  0.0604  0.0604   27.3786
##
## sigma^2 = 59399:  log likelihood = -1844.88
## AIC=3697.75   AICc=3697.91   BIC=3712.1
```

We have found another model with a `auto.arima` $ARIMA(2, 1, 0)$.

Conclusion

So the candidate models are:

- 1^{st} Model: The $ARIMA(3, 1, 2)$ model.
- 2^{nd} Model: The $ARIMA(3, 1, 1)$ model.
- 3^{th} Model: The $ARIMA(1, 1, 2)$ model.
- 4^{th} Model: The $ARIMA(1, 1, 1)$ model.
- 5^{th} Model: The $ARIMA(2, 1, 0)$ model.

5.2 Model estimation

Calibration consists in estimating the coefficients of the chosen model from the series.

5.2.1 The ARIMA(3,1,2) model

The first model applied is the model: $ARIMA(3, 1, 2)$.

The calibration results are:

```
#Implement ARIMA (3,1,2) model
model1 <- Arima(GDP_US.train,order=c(3,1,2),include.constant=T)
summary(model1)
```

```
## Series: GDP_US.train
## ARIMA(3,1,2) with drift
##
## Coefficients:
##          ar1     ar2      ar3      ma1      ma2     drift
##       0.8462  0.3795  -0.3021  -0.5227  -0.3810  145.5361
## s.e.  0.3321  0.4234   0.1142   0.3478   0.3473   19.2543
##
## sigma^2 = 59456:  log likelihood = -1843.51
## AIC=3701.02   AICc=3701.45   BIC=3726.13
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE      MAPE     MASE
## Training set -0.3940658 240.6306 181.3807 -0.0396611 0.6501856 0.230276
##                      ACF1
## Training set -0.005023785
```

We show that all parameters are not significant.

5.2.2 The ARIMA(3,1,1) model

The Second model applied is the model: $ARIMA(3,1,1)$.

The calibration results are:

```
#Implement ARIMA (3,1,1) model
model2 <- Arima(GDP_US.train,order=c(3,1,1),include.constant=T)
summary(model2)
```

```
## Series: GDP_US.train
## ARIMA(3,1,1) with drift
##
## Coefficients:
##           ar1     ar2     ar3     ma1     drift
##       -0.1812  0.3274  0.0072  0.5048  143.7315
## s.e.   0.4138  0.1342  0.0994  0.4092   26.1882
##
## sigma^2 = 59475:  log likelihood = -1844.04
## AIC=3700.08   AICc=3700.41   BIC=3721.61
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE     MAPE     MASE
## Training set 0.6822743 241.1286 182.5898 -0.02450532 0.655714 0.231811
##                      ACF1
## Training set -0.0004042123
```

We show that all parameters are not significant except `ar3`.

5.2.3 The ARIMA(1,1,2) model

The third model applied is the model: $ARIMA(1,1,2)$.

The calibration results are:

```
#Implement ARIMA (1,1,2) model
model3 <- Arima(GDP_US.train,order=c(1,1,2),include.constant=T)
summary(model3)
```

```
## Series: GDP_US.train
## ARIMA(1,1,2) with drift
##
## Coefficients:
##          ar1      ma1     ma2     drift
##       0.3606  -0.0426  0.1549  143.7157
## s.e.  0.2011   0.2019  0.0838   25.6789
##
## sigma^2 = 59407:  log likelihood = -1844.39
## AIC=3698.79   AICc=3699.02   BIC=3716.72
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE      MAPE      MASE
## Training set 0.6308826 241.4507 182.3738 -0.025266 0.6550279 0.2315368
##                       ACF1
## Training set -0.0004761539
```

We show that all parameters are not significant except `ma2`.

5.2.4 The ARIMA(1,1,1) model

The fourth model applied is the model: $ARIMA(1,1,1)$.

The calibration results are:

```
#Implement ARIMA (1,1,1) model
model4 <- Arima(GDP_US.train,order=c(1,1,1),include.constant=T)
summary(model4)
```

```
## Series: GDP_US.train
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1     drift
##       0.5912  -0.2575  143.6243
## s.e.  0.1035   0.1187   26.9300
##
## sigma^2 = 59865:  log likelihood = -1845.91
## AIC=3699.82   AICc=3699.97   BIC=3714.17
##
## Training set error measures:
##                      ME     RMSE      MAE         MPE      MAPE      MASE
## Training set 0.6927335 242.8404 184.4281 -0.02370764 0.6634858 0.2341448
##                      ACF1
## Training set -0.01794659
```

We show that all parameters are not significant.

5.2.5 The ARIMA(2,1,0) model

The fifth model applied is the model: $ARIMA(2,1,0)$.

The calibration results are:

```
#Implement ARIMA (2,1,0) model
model5 <- Arima(GDP_US.train,order=c(2,1,0),include.constant=T)
summary(model5)
```

```
## Series: GDP_US.train
## ARIMA(2,1,0) with drift
##
## Coefficients:
##          ar1     ar2     drift
##       0.3098  0.1508  143.6201
## s.e.  0.0604  0.0604   27.3786
##
## sigma^2 = 59399:  log likelihood = -1844.88
## AIC=3697.75   AICc=3697.91   BIC=3712.1
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE      MAPE     MASE
## Training set 0.738423 241.8929 183.3704 -0.02289844 0.6604698 0.232802
##                     ACF1
## Training set 0.01053612
```

We show that all parameters are significant except `drift`.

5.2.6 Model Selection

Comparison of the models

The above model was compared with different ARIMA models to select the best model for the data using different goodness-of-fit measures (log likelihood, AIC, AICc, MASE, and BIC). The results are presented in Table below:

Models	log likelihood	AIC	AICc	BIC	MASE	nbr of par. sig.
ARIMA(3,1,2)	-1843.51	3701.02	3701.45	3726.13	0.230	0
ARIMA(3,1,1)	-1844.04	3700.08	3700.41	3721.61	0.232	1
ARIMA(1,1,2)	-1844.39	3698.79	3699.02	3716.72	0.232	1
ARIMA(1,1,1)	-1845.91	3699.82	3699.97	3714.17	0.234	0
ARIMA(2,1,0)	-1844.88	3697.75	3697.91	3712.1	0.233	2

From the table we can see ARIMA(2,1,0) is the best model because have small AIC, AICc, BIC, and RMSE. The estimated regression equation of ARIMA(2,1,0) model is:

$$Y_t = c + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \varepsilon_t$$

$$Y_t = 27.38 + 0.31 Y_{t-1} + 0.15 Y_{t-2} + \varepsilon_t$$

5.3 Model checking

5.3.1 Verification of the diagnosis

Assumptions of the diagnosis:

1. Independence.

2. Normality.
3. Equality of variance.

Plotting the characteristic roots

Plot inverted AR and MA roots to check stationarity and invertibility. If time series can be represented as a finite order moving average process, it is **stationary**. If time series can be represented as a finite order autoregressive process, it is **invertible**. A causal invertible model should have all the roots outside the unit circle. Equivalently, **the inverse roots should lie inside the unit circle**.

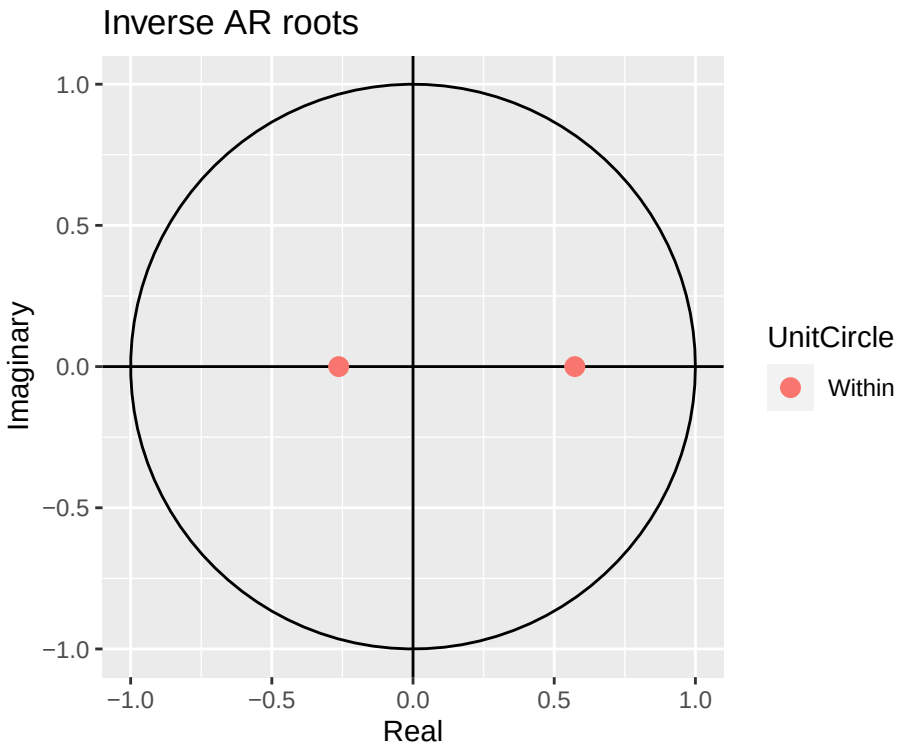
Chek the estimated model for adequacty

From the graph below, the red dot in the left hand plot correspond to the root of the polynomial $\phi(B)$. They are all inside the unit circle, as we would expect because `R` ensures the fitted model is both stationary and invertible. Any roots close to the unit circle may be numerically unstable, and the corresponding model will not be good for forecasting.

```
library(knitr)
opts_knit$set(global.par = TRUE)
```

The important think in the previous chunk is to put `global.par=TRUE`

```
par(mar=c(5,5,0,0)) #it's important to have that in a separate chunk
autoplot(model5)
```



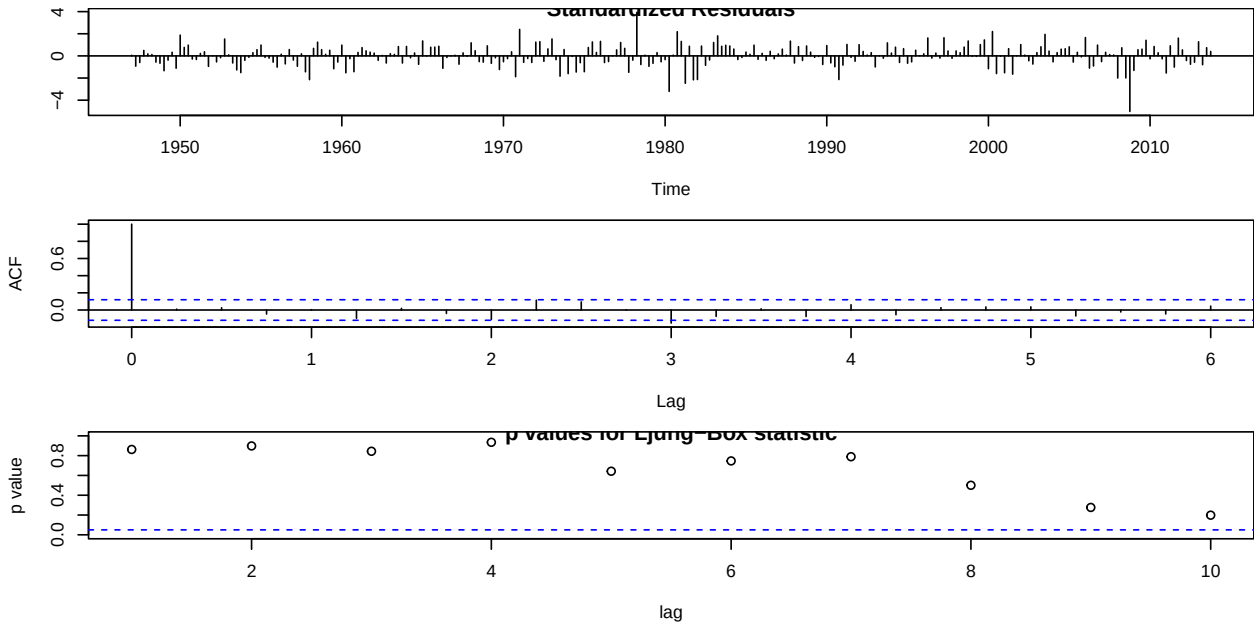
From the figures above, inverse AR roots lie inside the unit circle. Thus this time series are staionary and invertible.

Diagnose residuals

Ljung-Box autocorrelation test on residuals

The results below plot the standardized residuals, the autocorrelation function of the residuals, and the `p-values` for Ljung-Box statistic for lags 10.

```
tsdiag(model5)
```



The figure above represents the residuals, the ACF of the residuals and the Ljung-Box static test, which provides the hypothesis of no autocorrelation of the residuals up to lag 10.

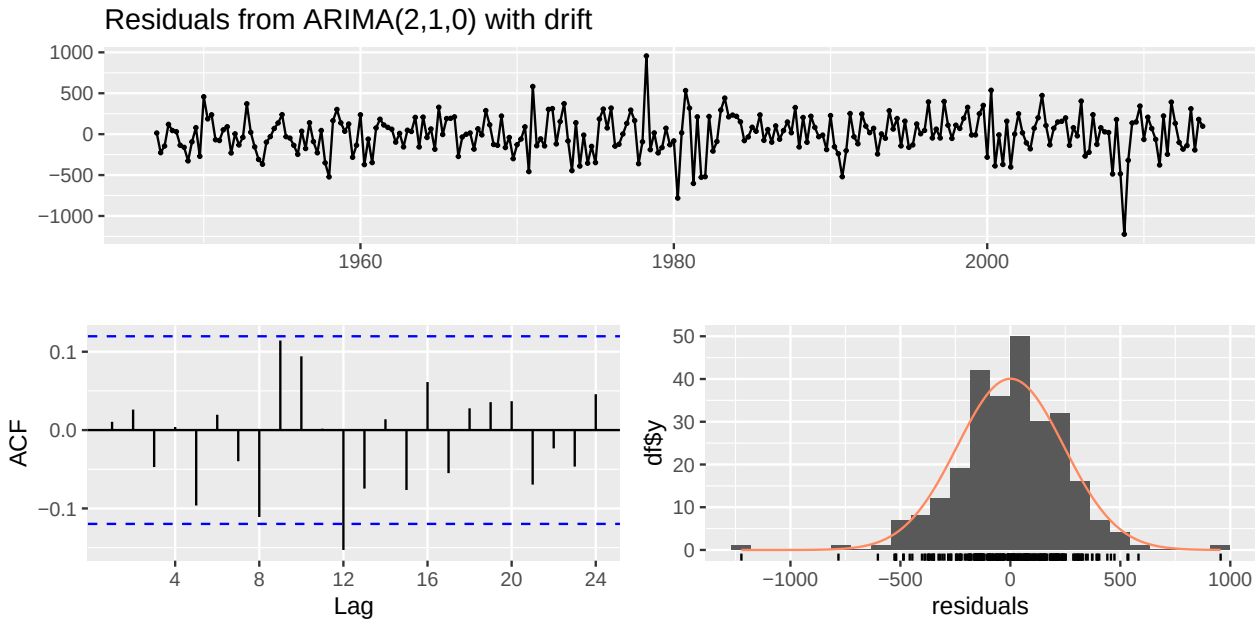
With the normality test below, we can assume that the residuals follow a normal distribution because the `p-value` (0.1966) is greater than the 5% significance level.

Test of normality of residuals

```
library(knitr)
opts_knit$set(global.par = TRUE)
```

The important think in the previous chunk is to put `global.par=TRUE`

```
par(mar=c(5,5,0,0)) #it's important to have that in a separate chunk
#Check Residuals for ARIMA (2,1,0) model
checkresiduals(model5)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0) with drift
## Q* = 7.3398, df = 5, p-value = 0.1966
##
## Model df: 3.   Total lags used: 8
```

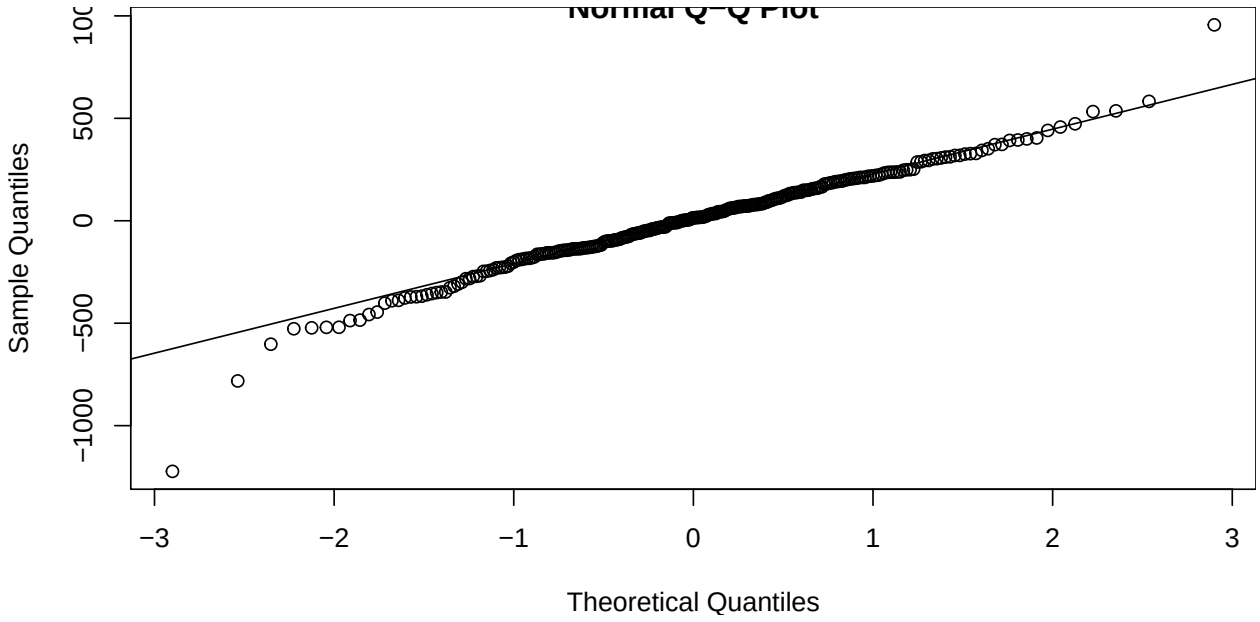
As we can see on the graph, there is no autocorrelation between residuals and they are normally distributed. However, models with a very large number of parameters to be estimated should be avoided. number of parameters to be estimated, which are synonymous with estimation errors.

The normality test of the residuals provides the following results:

```
res=residuals(model5)
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.97255, p-value = 5.008e-05
```

```
qqnorm(res);
qqline(res)
```



With the normality test above, we can suppose that the residuals follow a normal distribution (see histogram above), so we have a white noise. have a white noise. But we are aware that this assumption is very limited and that there would be a way to do better by choosing a better model. to do better by choosing a better model. However we should avoid models with a very large number of parameters to be estimated which are associated with estimation errors.

5.4 Forecasting

From the estimated model, we can make a prediction about the next 31 quarters of the series Real GDP per Capita.

Make Predictions using Multi-Steps Forecast for forward-looking 31 quarters (7 years and 3 quarters) and using One-Step Forecast without Re-Estimation on the test Data set.

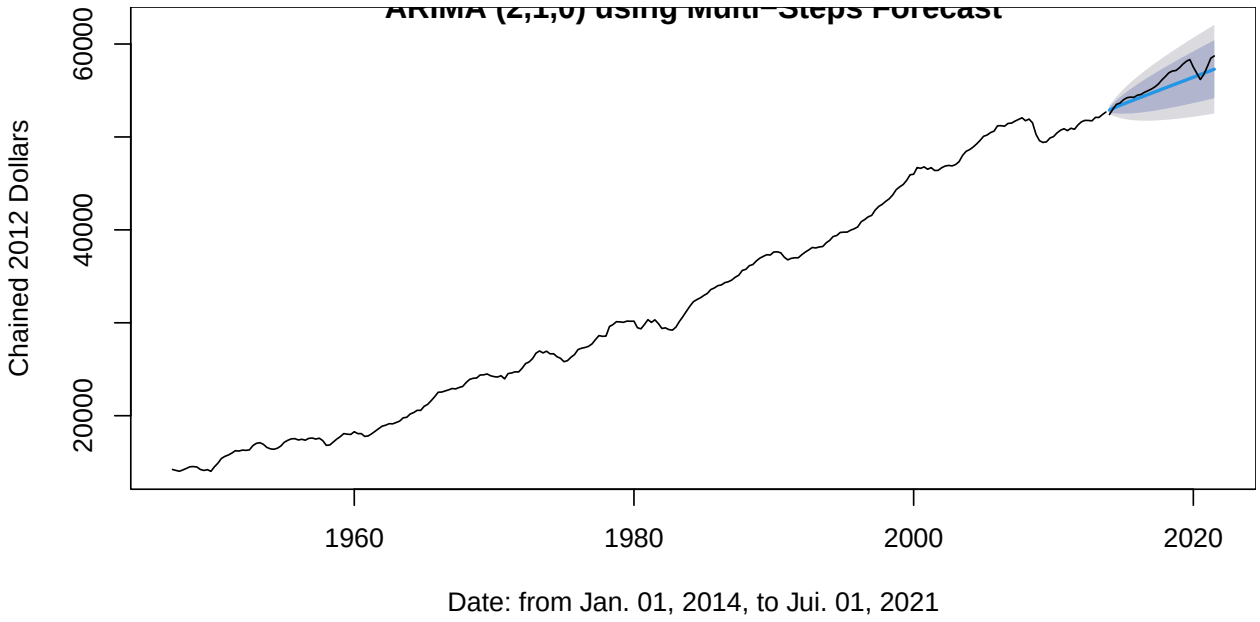
We can make predictions using two methods:

- a) Multi-Step Forecast
- b) One-Step Forecast without Re-Estimation

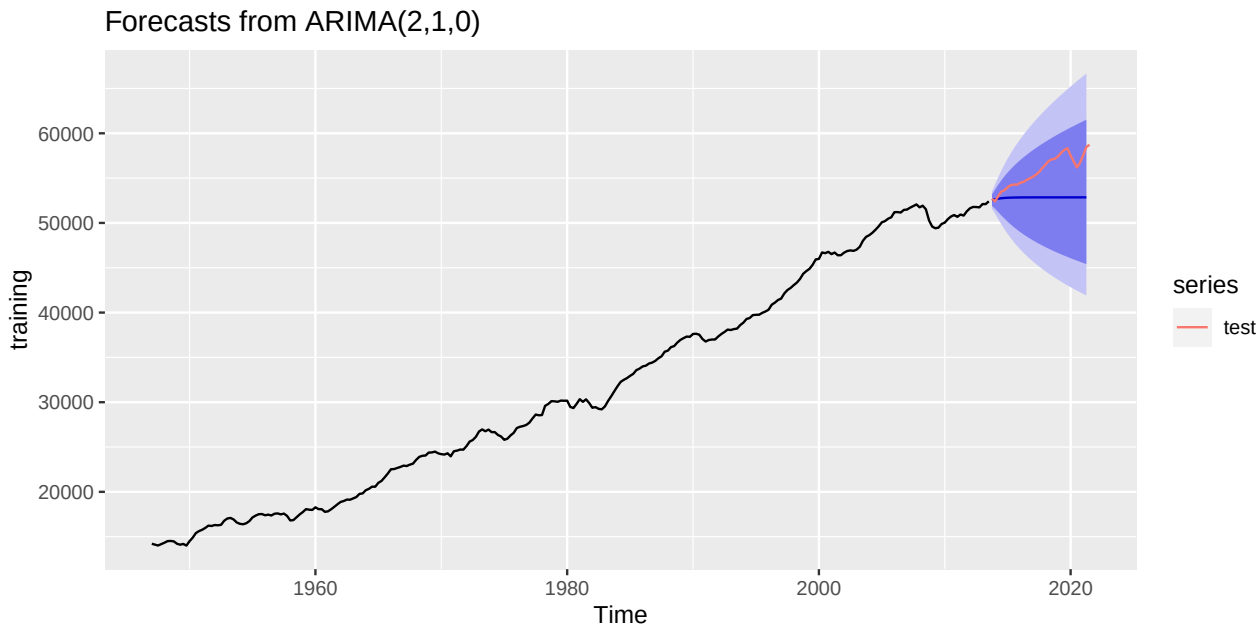
5.4.1 Multi-Step Forecast

The Multi-Step Forecast does the job of forecasting the next 31 quarters (7 years and 3 quarters) without using the test data set.

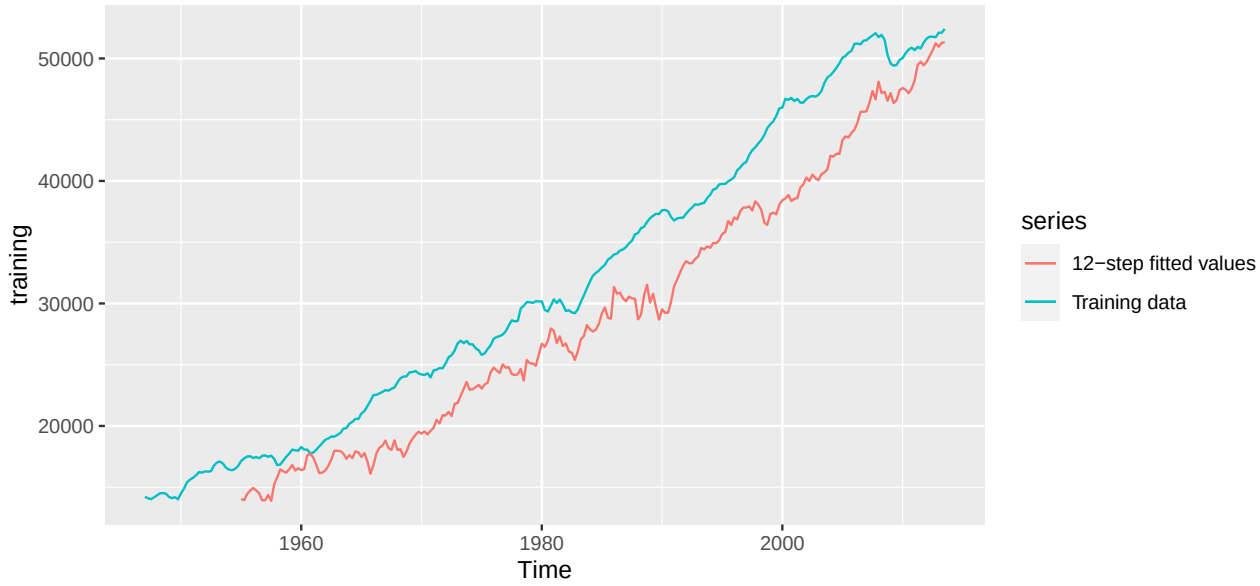
```
# Multi-Steps Forecast
plot(forecast(model5,h=31),main="ARIMA (2,1,0) using Multi-Steps Forecast",ylab=
       "Chained 2012 Dollars",xlab="Date: from Jan. 01, 2014, to Jui. 01, 2021")
lines(GDP_US.test,lty=1)
```



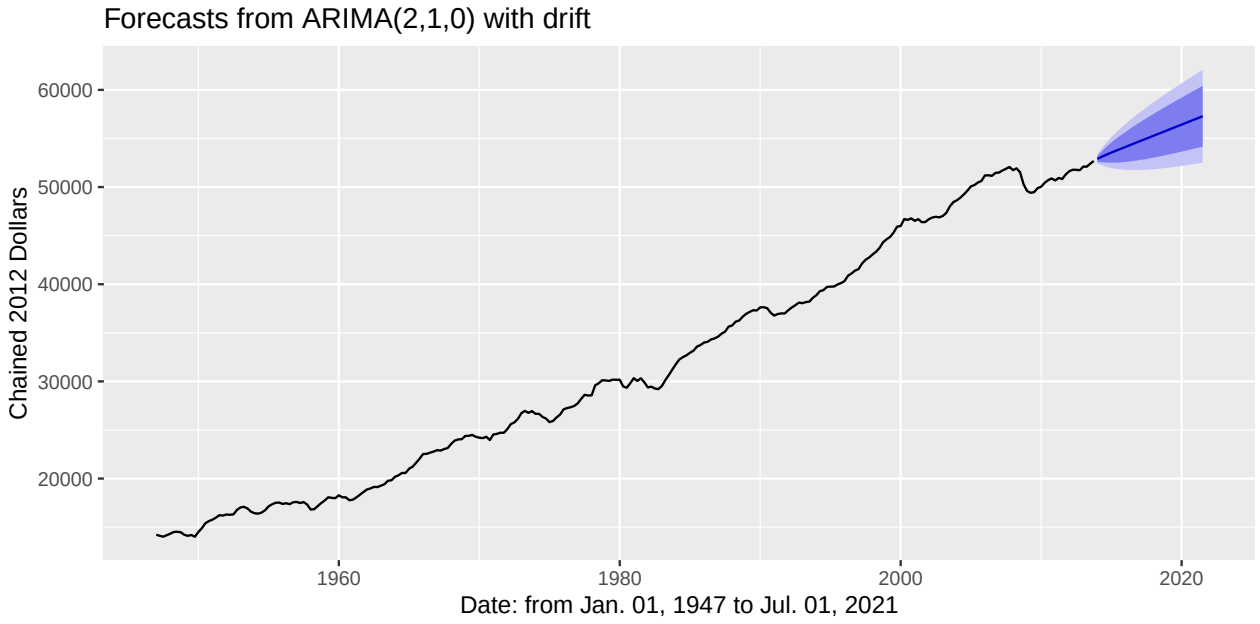

```
training <- subset(GDP_US_clean, end=length(GDP_US_clean)-32)
test <- subset(GDP_US_clean, start=length(GDP_US_clean)-31)
GDP.train <- Arima(training, order=c(2,1,0),
  seasonal=c(0,0,0), lambda=0)
GDP.train %>%
  forecast(h=31) %>%
  autoplot() + autolayer(test)
```




```
autoplot(training, series="Training data") +
  autolayer(fitted(GDP.train, h=31),
    series="12-step fitted values")
```



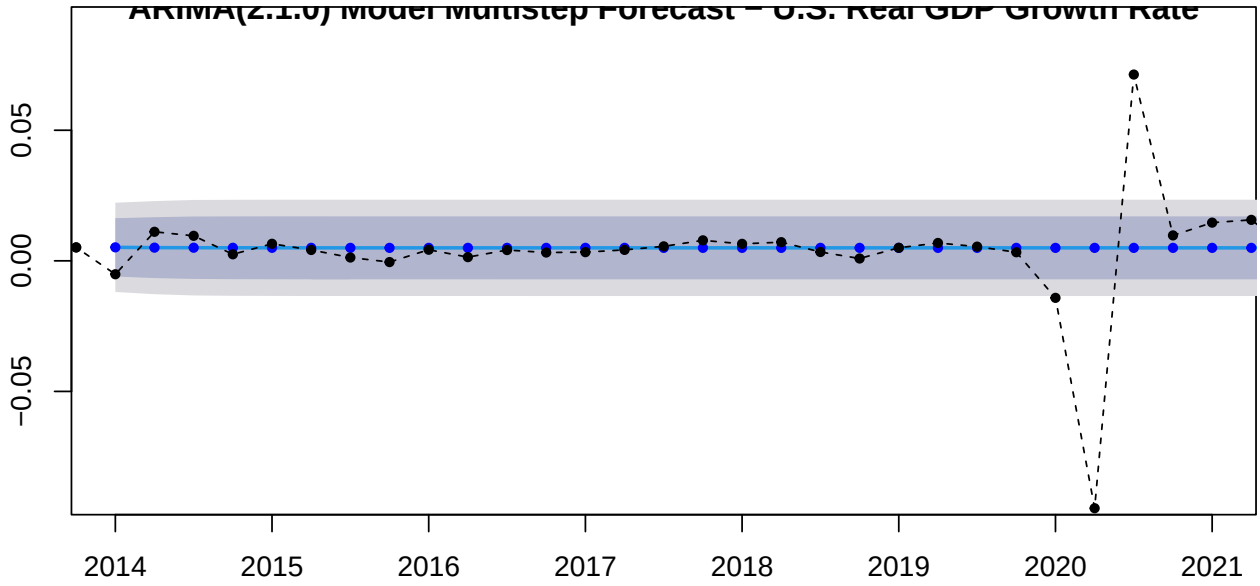

```
autoplot(forecast(model5,h=31),ylab="Chained 2012 Dollars",xlab=
           "Date: from Jan. 01, 1947 to Jul. 01, 2021")
```



The graphs above represents the Real GDP Per Capita as well as the prediction for the next 7 years and 3 quarters.

The multistep forecasts using ARMA(2,0)

```
arima200 <- Arima(difflogGDP.train, order=c(2,0,0))
arma21.f.h <- forecast(arima200, length(difflogGDP.test))
plot(arma21.f.h, type="o", pch=20, xlim=c(2014,2021), ylim=c(-0.09,0.09),
main="ARIMA(2.1.0) Model Multistep Forecast - U.S. Real GDP Growth Rate")
lines(arma21.f.h$mean, type="p", pch=20, lty="dashed", col="blue")
lines(difflogGDP, type="o", pch=20, lty="dashed")
```

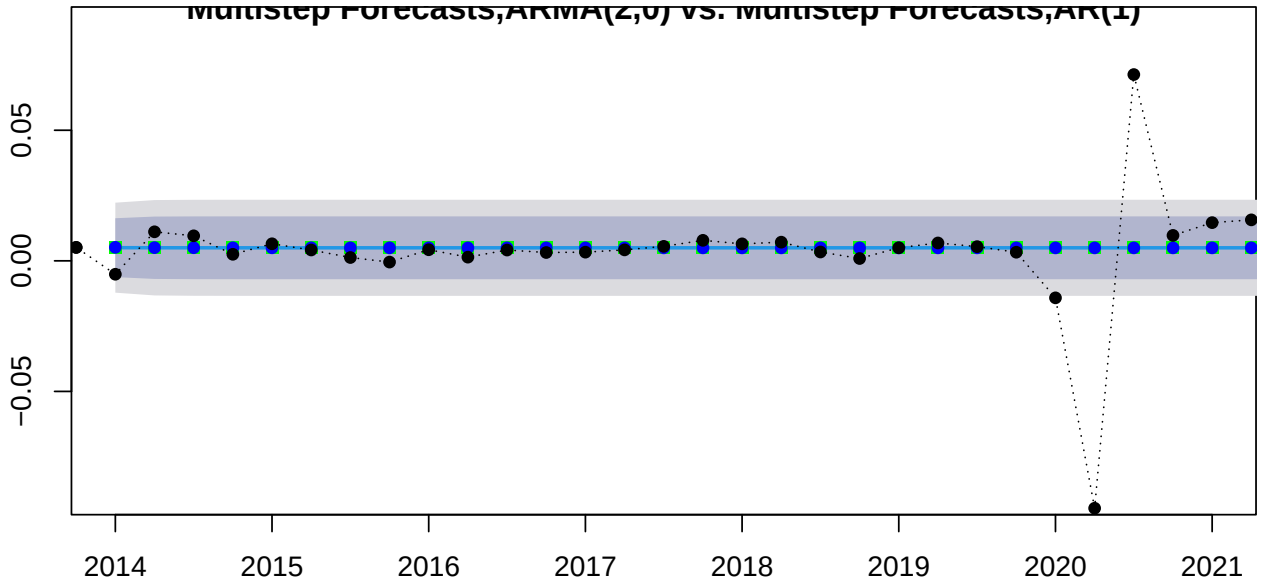


Below is the same forecast, but with a retrospective to 1947.

```
GDP.fcst <- forecast(arima200, h=31)
plot(GDP.fcst, xlim=c(1947,2021))
```

Comparison: Multistep forecasts using AR(1) vs. Multistep forecasts using ARMA(2,0)

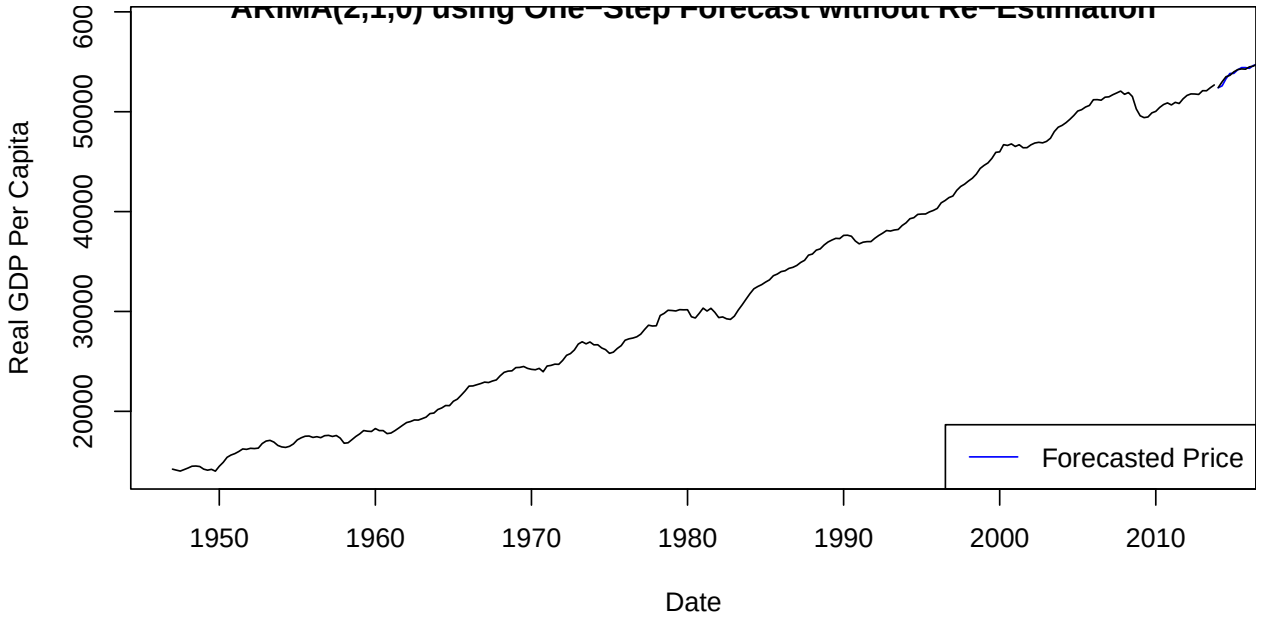
```
ar1 <- Arima(difflogGDP.train, order=c(1, 0, 0))
ar1.f <- forecast(ar1, length(difflogGDP.test))

plot(ar1.f, type="o", pch=16, xlim=c(2014, 2021), ylim=c(-0.09, 0.09), main=
       "Multistep Forecasts,ARMA(2,0) vs. Multistep Forecasts,AR(1)")
lines(ar1.f$mean, type="p", pch=15, lty="dashed", col="green")
lines(arma21.f.h$mean, type="p", pch=16, lty="dashed", col="blue")
lines(difflogGDP, type="o", pch=16, lty="dotted")
```



5.4.2 One-Step Forecast without Re-Estimation

The One-Step Forecast does forecasting considering test data set.

```
# One-Step Forecast without Re-Estimation
model <- Arima(GDP_US.test,model=model5)$fitted
plot(GDP_US.train,main="ARIMA(2,1,0) using One-Step Forecast without Re-Estimation",ylab=
       "Real GDP Per Capita",xlab="Date",ylim=c(min(GDP_US),max(GDP_US)))
lines(model,col="blue")
lines(GDP_US.test,lty=1)
legend("bottomright",col="blue",lty=1,legend="Forecasted Price")
```



In general we can conclude that the prediction does not betray the structure of the series.

6 Conclusion

In our study, we studied the real GDP series. The purpose of this research was to model, calibrate and predict the series. The modeling consisted in finding models that will estimate them with the least possible error. For the series, the ARIMA model was used, and the autoregression and moving average coefficients were chosen selectively from the observation and assessment of the autocorrelogram and partial autocorrelogram. We tested a number of models, then selected those that provided residuals without autocorrelation (ljung-box test) and with normal distribution (normality test). Then we calibrated the selected models by the likelihood method, i.e. we estimated the parameters of the ARIMA model. After modeling and calibration we predicted the next 31 values of the series. Overall, the prediction does not contradict the structure of the series. However we are conscious that the models are not unique, and that it is possible to find other models with better estimation performances. These models can be found by differentiating the series in different ways. For both series, we had residuals without autocorrelation, but not necessarily with normal distribution and zero mean, because of the extreme values present in the residuals. So we assumed that the residuals are white noise and we made the predictions. All the difficulty of the time series is in the modeling, and with a deeper study and with advanced technical and theoretical means, it is possible to "find" the best model for each series.

Happy forecasting!

7 References

1. Forecasting Real GDP Rate through Econometric Models: An Empirical Study from Greece
2. Calculating Real GDP and Growth rates in R
3. Applying the ARIMA Model to the Process of Forecasting GDP and CPI in the Jordanian Economy
4. Modelling GDP for Sudan using ARIMA
5. Forecasting Egyptian GDP Using ARIMA Models
6. A Model for Forecasting: Real GDP
7. Homework 3, Problem 3: Real Gross Domestic Product
8. One-step forecasts on test data
9. Is the United States of America (USA) really being made great again? witty insights from the Box-Jenkins ARIMA approach