Aliah University

Department of Mathematics and Statistics

End Term Examination (Spring Semester) - 2022

Project Work On: Forecasting GDP & HDI OF India Using Hybrid Model

Submitted By:

Name- Ranadip Kar
Roll No.- STI202007
Regn. No.- 1342 of 2020-2021
Course Code- STAT272

ACKNOWLEDGEMENT

I would like to express my special thanks of gratitude to my teacher **Dr. Hare Krishna Maity** who gave me the golden opportunity to do this wonderful project on the topic **“Forecasting GDP & HDI Of India Using Hybrid Model”**, which also helped me to pursue my project work and I came to know about so many new things. I am really thankful to her. I also heartily express my gratitude to all the teachers of this department for their constant support and guidance.

Secondly, I would also like to thank my parents and friends who helped me a lot in finishing this project within the limited time. I am making this project not only for marks but to also increase my knowledge. Thanks again to all who helped me.

Ranadip Kar
Dept. of Statistics,
Aliah University.

TO WHOM IT MAY CONCERN

This is to certify that the 4^{th} semester student of M.Sc. in Statistics with Roll N0.- STI202007, Reg. NO: 1342 of 2020-2021, has completed the project work (Academic session: 2022) entitled **“Forecasting GDP & HDI Of India Using Hybrid Model”** from the department of Statistics, Aliah University under the supervision of the faculties of the department.

Aspiring for every success in life, thanking you.

Dr. Hare Krishna Maity

Aliah University

CONTENTS

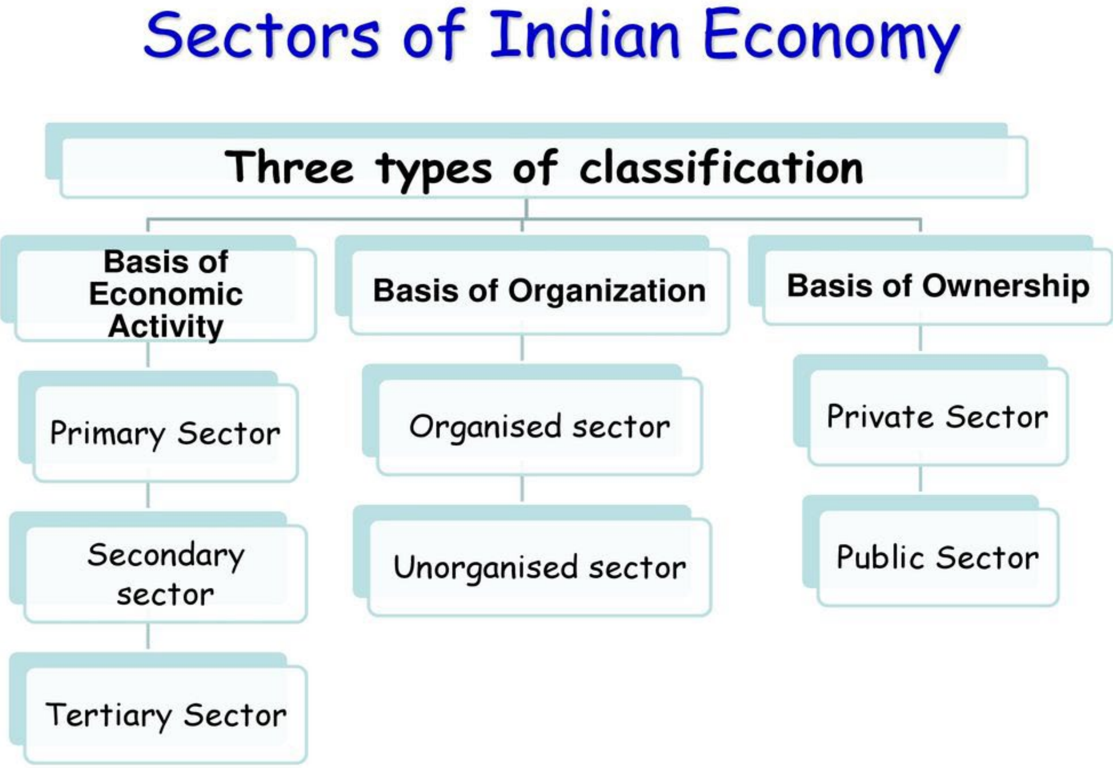
INTRODUCTION

Over the last decade, India has emerged as a significant player in the global economic arena. The nation's economy has embraced international trade more openly, its workforce continues to expand robustly, and investment rates have surged following economic reforms.

Gross Domestic Product (GDP) stands as a key indicator of economic performance, measuring a nation's total output over a specific period. This metric is seasonally adjusted to account for quarterly fluctuations due to climate or holidays. The most scrutinized GDP measure is inflation-adjusted, focusing on changes in output rather than price variations. While annual GDP figures are often used to compare economies by size, policymakers, financial market participants, and business leaders typically focus on GDP growth or contraction rates, reported on an annualized basis for easy comparison of quarterly and annual figures.

India's economy is among the world's fastest-growing, with projections suggesting it could become the second-largest global economy by 2050. This growth is driven by a multitude of factors and spans various sectors, each contributing to the nation's economic landscape.

The Indian economy's multifaceted nature and rapid growth trajectory position it as a focal point in the global economic discourse, with its development patterns and policies holding significant implications for international trade, investment, and economic cooperation. There are many sectors of the Indian economy.



Primary Sector: In the primary sector of the economy, activities directly utilize natural resources, such as agriculture, mining, fishing, forestry, and dairy. This sector is fundamental as it provides the base for other products and is also known as the agriculture and allied sector. Workers in this sector are called red-collar workers due to the outdoor nature of their work.

Secondary Sector: The secondary sector, also known as the industrial sector, manufactures finished products from primary sector materials, such as industrial production and cotton fabric. It focuses on manufacturing goods rather than producing raw materials. Workers in this sector are called blue-collar workers.

Tertiary Sector/Service Sector: The tertiary sector supports the development of the primary and secondary sectors by providing essential services like transportation, banking, insurance, and finance. While it doesn't produce goods, it adds value similarly to the secondary sector. Jobs in this sector are known as white-collar jobs.

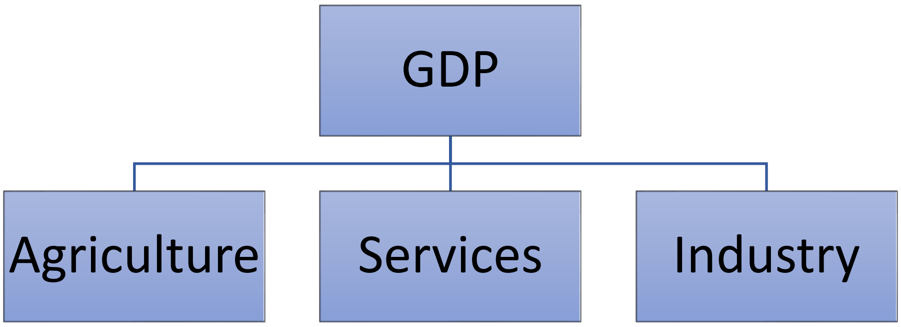
Organised Sector: The organized sector offers fixed employment terms, job security, and social benefits. It includes government-registered institutions like schools and hospitals. Workers have set hours and receive overtime pay for extra work.

Unorganised Sector: The unorganised sector encompasses home-based, self-employed, and wage workers not covered by specified welfare Acts. It features non-unionised, casual, and seasonal employment with scattered enterprises, resulting in low incomes and unstable work. Workers lack legal protections and union representation. Despite these challenges, the sector significantly contributes over 60% to national income, outpacing the organised sector in many industries.

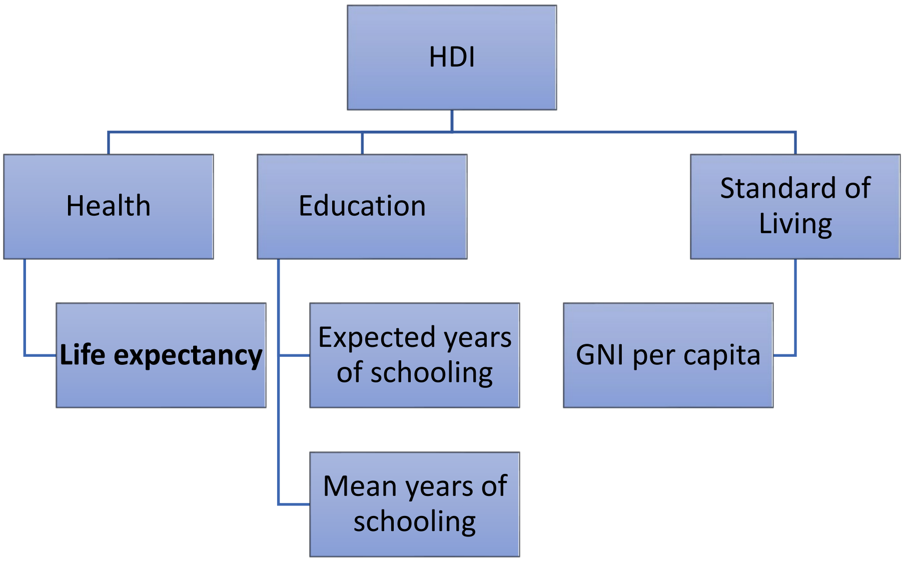
The Public Sector: The public sector predominantly consists of government-owned assets and focuses on providing essential governmental services. Its primary aim isn't profit generation; instead, governments finance services through taxes and other means.

The Private Sector: In the private sector, assets and services are owned and managed by individuals or companies for profit. It operates independently of state control but is regulated by government policies. Payment is required for services rendered.

There are three major sectors of GDP of India. These are Agriculture Sector, Industry and Services sector.



The Human Development Index (HDI) is a statistical tool used to measure a country's overall achievement in its social and economic dimensions. The social and economic dimensions of a country are based on the health of people, their level of education attainment and their standard of living. The Human Development Index examines three important criteria of economic development (life expectancy, education and income levels) and uses this to create an overall score between 0 and 1. The value 1 indicates a high level of economic development and 0 a very low level.



OBJECTIVE

The main objectives of this project are:

- ✓ To fit a model which can predict GDP more accurately (2021-2030).
- ✓ Fitting model to predict GDP for 2011-2020 and compare our estimated value with the original one (original GDP for 2011-2020) and collect our required information.
- ✓ Predict Population for next 10 years (2021-2030).
- ✓ Predict GNI per capita (2020-2030) with GDP and Population dat.
- ✓ Predict HDI for 2020-2030 with help of three indicators.

METHODOLOGY

Time Series Analysis and Forecasting: Time Series Analysis studies a response variable over time, using time as the reference for predicting future values. It involves sequential time-based data such as years, months, days, and seconds. Applications include weather forecasting, stock market predictions, and signal processing. TSA is distinct from other analyses due to its focus on time sequences. Future predictions are made using models like AR, MA, ARMA, and ARIMA.

Time-series forecasting is a technique that uses historical and current data to predict future values over a specific period or at a particular point in the future.

A time series can be constructed by any data that is measured over time at evenly spaced intervals. Historical stock prices, earnings, GDP, or other sequences of financial or economic data can be analyzed as a time series.

When forecasting time series values, three key elements must be considered:

- ✓ **Seasonality:** Regular patterns where certain months show peak values, such as higher activity in November and December for the travel industry due to holidays.
- ✓ **Trend:** The overall direction of the time series, indicating whether values are generally increasing or decreasing over time.
- ✓ **Unexpected Events:** Unpredictable changes, like the recent pandemic, which can cause significant deviations in the data.

Autoregressive models (AR model): In a multiple regression model, the variable of interest is forecasted using a linear combination of predictors. In an autoregression model, the variable of interest is forecasted using a linear combination of its past values. The term "autoregression" indicates that it is a regression of the variable against itself.

Thus, an autoregressive model of order p can be written as follows:

$$Y_t = \mathrm{c} + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \ldots\ldots.. + \phi_p Y_{t-p} + \varepsilon_t$$

where ε_t is white noise. This is like a multiple regression but with lagged values of Y_t as predictors. This is referred to as an AR(p) model, which is an autoregressive model of order p. An AR(1) autoregressive process is one in which the current value is based on the immediately preceding value, while an AR(2) process is one in which the current value is based on the previous two values.

Moving Average (MA): Moving Average (MA) is a statistical technique used in time series analysis to estimate underlying trends or patterns by averaging the values from a specified number of preceding periods. This method operates on the principle that the current value of a time series is influenced by the average of previous periods' values.

The mathematical formula for Moving Average (MA), is as follows:

MA(t) = (X(t) + X(t-1) + … + X(t-n+1)) / n where X(t) represents the value of the time series at time t, n represents the number of periods in the moving window, and MA(t) is the Moving Average value at time t.

The importance of MA lies in its capacity to estimate the underlying trend in a time series by eliminating noise effects. This method finds extensive application in time series forecasting, trend analysis, and data smoothing.

ARMA Model: ARMA stands for auto-regressive moving average, a forecasting technique that combines auto-regressive (AR) and moving average (MA) models. In an AR model, forecasts are linear and additive, calculated as the sum of past values multiplied by a scaling factor plus residuals. Moving average models consist of different types such as simple, cumulative, and weighted averages. ARMA models integrate both AR and MA methodologies to produce accurate forecasts.

ARIMA Model: An Autoregressive Integrated Moving Average (ARIMA) model is an extension of the Autoregressive Moving Average (ARMA) model. Both are employed for forecasting future points in time-series data. ARIMA, a type of regression analysis, assesses the impact of a dependent variable relative to changing variables. Its primary goal is to predict future movements in time series by examining differences between values rather than the values themselves. ARIMA models are particularly useful when dealing with non-stationary data, which must be transformed into stationary data for effective time series analysis.

The parameters of the ARIMA model are defined as follows:

- ✓ p: Represents the number of lag observations included in the model, known as the lag order.
- ✓ d: Represents the number of times the raw observations are differenced, referred to as the degree of differencing.
- ✓ q: Represents the size of the moving average window, known as the order of the moving average.

Auto Correlation Function (ACF): Autocorrelation in a time series refers to the correlation between observations at different time points. It shows whether values separated by an interval have a strong positive or negative relationship. Such correlations suggest that past values influence the current value. Analysts utilize autocorrelation and partial autocorrelation functions to grasp time series characteristics, select suitable models, and predict future outcomes.

Auto Correlation function takes into consideration of all the past observations irrespective ofits effect on the future or present time period. It calculates the correlation between the t and (t-k) time period. It includes all the lags or intervals between t and (t-k) time periods.

Partial Correlation Function (PACF): The PACF determines the partial correlation between time period t and t-k. It doesn't take into consideration all the time lags between t and t-k. The partial autocorrelation function (PACF) resembles the autocorrelation function (ACF), but it specifically shows correlations between observations that aren't explained by shorter lags. For instance, the partial autocorrelation at lag 3 captures the correlation not accounted for by lags 1 and 2. Put simply, the PACF at each lag isolates the correlation between two observations after removing the influence of intervening correlations.

Stationarity in Time Series Analysis: Stationarity indicates that the statistical characteristics of a time series—such as its mean, variance, and covariance—stay consistent over time. This stability is crucial for numerous time series modeling techniques, simplifying data dynamics and facilitating easier analysis, modeling, and forecasting. Stationarity is crucial in time series analysis, defining datasets where statistical properties remain constant over time. This stability implies independence between data points, aiding accurate forecasts. Stationary data exhibits consistent behavior without trends or seasonality in time plots, with numerical indicators such as constant mean and variance supporting its characterization.

Augmented Dickey-Fuller Test (ADF): The Augmented Dickey-Fuller test (ADF test) is a widely used statistical test to determine if a given time series is stationary. It is among the most frequently employed tests for assessing stationarity in a series.
This test identifies whether a time series exhibits a unit root, which is a characteristic indicating a stochastic trend that causes deviations from its mean value. The presence of a unit root renders a time series non-stationary, posing challenges in drawing statistical inferences and making future predictions based on the data.

The ADF test is conducted under the following assumptions:

- Null Hypothesis (HO): The series is non-stationary, or it has a unit root.
- Alternate Hypothesis (HA): The series is stationary, or it does not have a unit root.

If the null hypothesis fails to be rejected, the test suggests that the series may be non-stationary.

Conditions to Reject the Null Hypothesis (HO): If the Test statistic < Critical Value and p-value < 0.05, the Null Hypothesis (HO) is rejected. This indicates that the time series does not have a unit root and is stationary. It lacks a time-dependent structure.

DESCRIPTION OF DATA

- ✓ **GDP Data:** GDP of India between 1960 and 2020, along with data for the three major sectors (i.e., Agriculture, Industry, and Services) for the same time period.
- ✓ **Population Data:** Population of India between 1950 and 2020.
- ✓ **HDI Data:** HDI of India between 1990 and 2019, along with the three major indicators (life expectancy, education and Standard of living) for the same period of time.

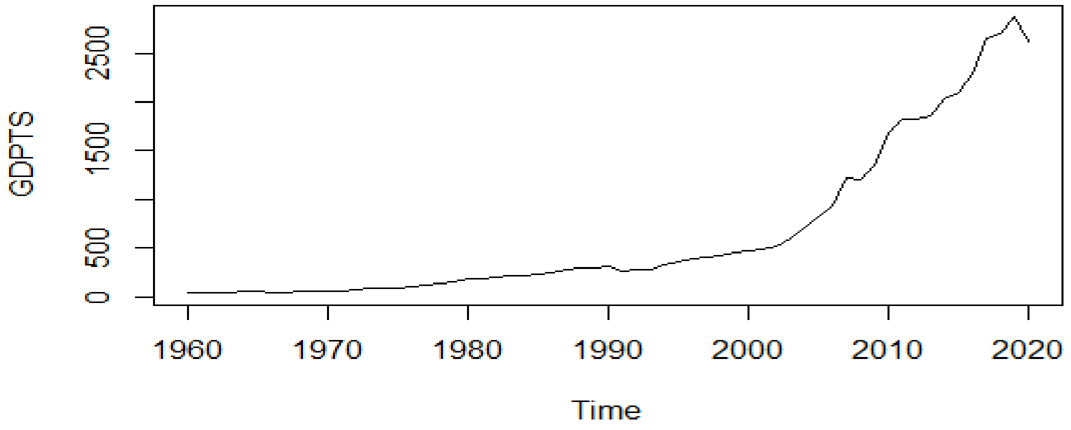
COMPUTATIONAL WORK

In this project, all calculations have been conducted using the R programming language.

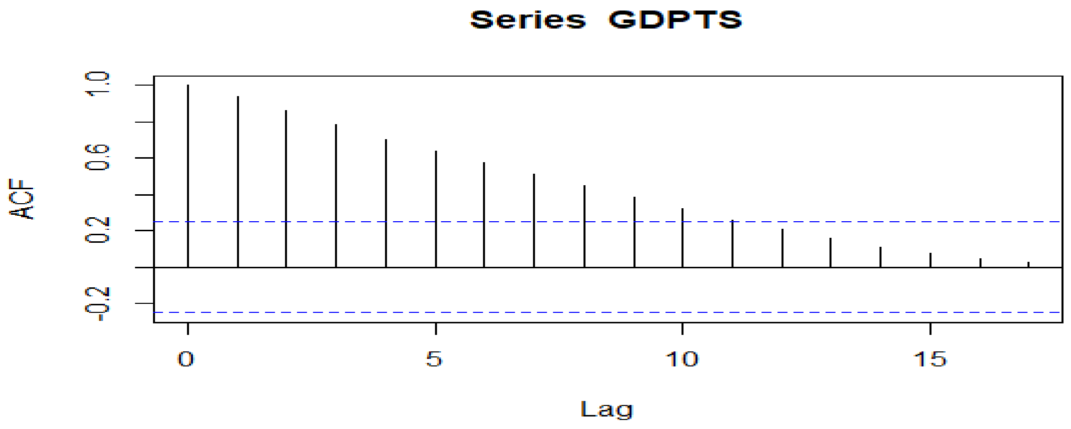
The first step involves predicting the GDP for the years 2021 to 2030 based on GDP data from 1960 to 2020, utilizing time series forecasting techniques.

Code and results:

```
> library(tseries)
> library(forecast)
> library(readxl)
> # Loading the Data Set
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-
Industry.xlsx")
> View(DATA)
> # Here we are taking GDP data and converting the data set into time series form
> GDPTS=ts(DATA$`GDP(in Billion
US$)`,start=min(DATA$Year),end=max(DATA$Year),frequency=1)
> plot(GDPTS)
```



```
> # now checking whether the data set is stationary or not
> acf(GDPTS)
```



```
> pacf(GDPTS)
> adf.test(GDPTS)

        Augmented Dickey-Fuller Test

data:  GDPTS
Dickey-Fuller = -0.16683, Lag order = 3, p-value = 0.99
alternative hypothesis: stationary

Warning message:
In adf.test(GDPTS) : p-value greater than printed p-value
```

- *Here p-value is 0.99 so it is not less than 0.05 so our data is not stationary. So now we will fit ARIMA model to forecast our GDP.*

```
> # Fitting ARIMA model
> GDPMODEL=auto.arima(GDPTS,ic="aic",trace=TRUE)

 ARIMA(2,2,2)                    : 701.2097
 ARIMA(0,2,0)                    : 715.6442
 ARIMA(1,2,0)                    : 704.9784
 ARIMA(0,2,1)                    : 695.3485
 ARIMA(1,2,1)                    : 697.3276
 ARIMA(0,2,2)                    : 697.3287
 ARIMA(1,2,2)                    : 699.2674

 Best model: ARIMA(0,2,1)

> GDPMODEL
Series: GDPTS
ARIMA(0,2,1)

Coefficients:
          ma1
      -0.8397
s.e.   0.0775
```

```
sigma^2 = 7160:  log likelihood = -345.67
AIC=695.35   AICc=695.56   BIC=699.5
```

- *No auto regressive part presenting in time series, here second order difference of GDP is stationary, only MA portion is contributing in GDP and the corresponding coefficients are also given above.*

```
> # Forcast the GDP
> forecast(GDPMODEL,h=10)
> GDPFOR=data.frame(forecast(GDPMODEL,h=10))
> GDP1=GDPFOR$Point.Forecast
> GDP1
 [1] 2704.305 2785.631 2866.956 2948.281 3029.607 3110.932 3192.257 3273.582
3354.908
[10] 3436.233
```

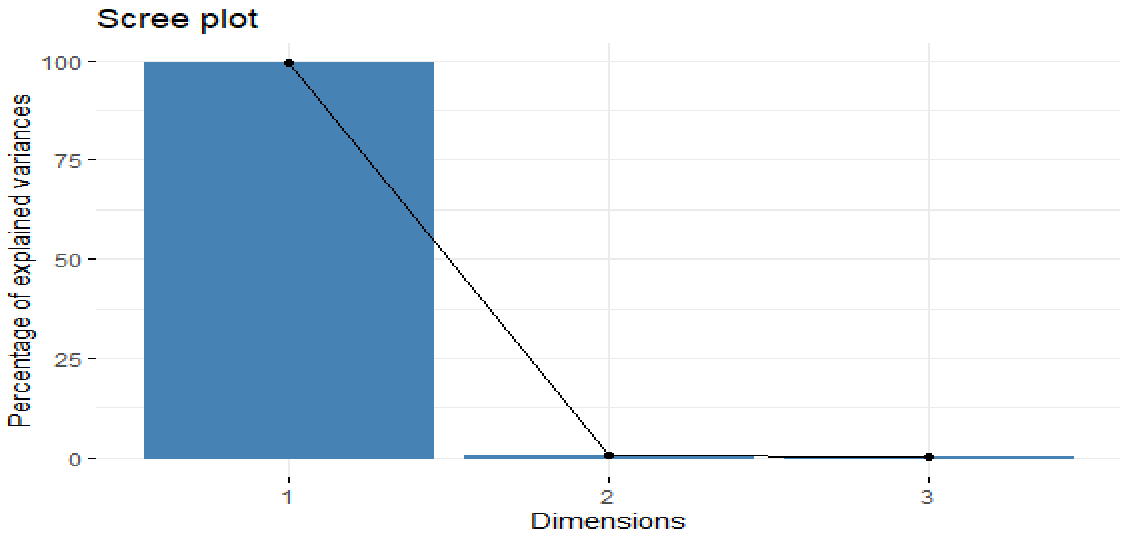
Forecast GDP for the year 2021-2030 (GDP1)										
Year	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030
GDP	2704.305	2785.631	2866.956	2948.281	3029.607	3110.932	3192.257	3273.582	3354.908	3436.233

Now, an attempt is made to predict the GDP-Index (GDP-Score) for the years 2021-2030 using data from the three major sectors of GDP: Agriculture, Services, and Industry, which are the main pillars of India's GDP.

Code and results:

```
> # Loading the Data Set of Agriculture & services & industry
> library(readxl)
> library(factoextra)
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-Industry.xlsx")
> ASI=DATA[c(3:5)]
> View(ASI)
> # PCA
> PCA=prcomp(ASI,scale. = TRUE)
> PCA
Standard deviations (1, .., p=3):
[1] 1.72659049 0.11879572 0.06908587

Rotation (n x k) = (3 x 3):
                          PC1        PC2        PC3
Agriculture(Billion US$) 0.5775332 -0.5157447  0.6328213
Industry(Billion US$)    0.5764878  0.8065076  0.1311763
Services(Billion US$)    0.5780287 -0.2890551 -0.7631055
> summary(PCA)
Importance of components:
                          PC1    PC2     PC3
Standard deviation     1.7266 0.1188 0.06909
Proportion of Variance 0.9937 0.0047 0.00159
```

```
Cumulative Proportion  0.9937 0.9984 1.00000
> #vIZUALIZATION AND SCREE PLOT
> fviz_eig(PCA)
```



```
> PCA_VAR=(PCA$sdev)^2
> PROP.VAR=c()
> for (i in 1:3){
+   PROP.VAR[i]=(PCA_VAR[i]/sum(PCA_VAR))*100
+ }
> PROP.VAR
[1] 99.3704907  0.4704141  0.1590953
```

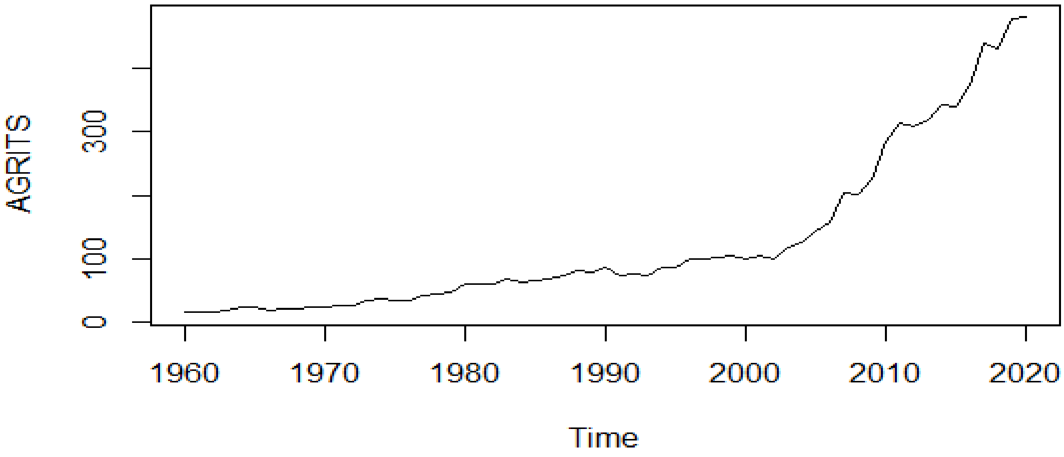
- ***Here we can see from our Principle Component Analysis that there are only one Principle Component.***

```
> # Factor Analysis
> FACT=factanal(ASI,factors=1)
> FACT

Call:
factanal(x = ASI, factors = 1)

Uniquenesses:
Agriculture(Billion US$)   Industry(Billion US$)   Services(Billion US$)
                  0.006                   0.018                   0.005

Loadings:
                         Factor1
Agriculture(Billion US$) 0.997
Industry(Billion US$)    0.991
Services(Billion US$)    0.998

               Factor1
SS loadings      2.972
Proportion Var   0.991

The degrees of freedom for the model is 0 and the fit was 0.0259
```

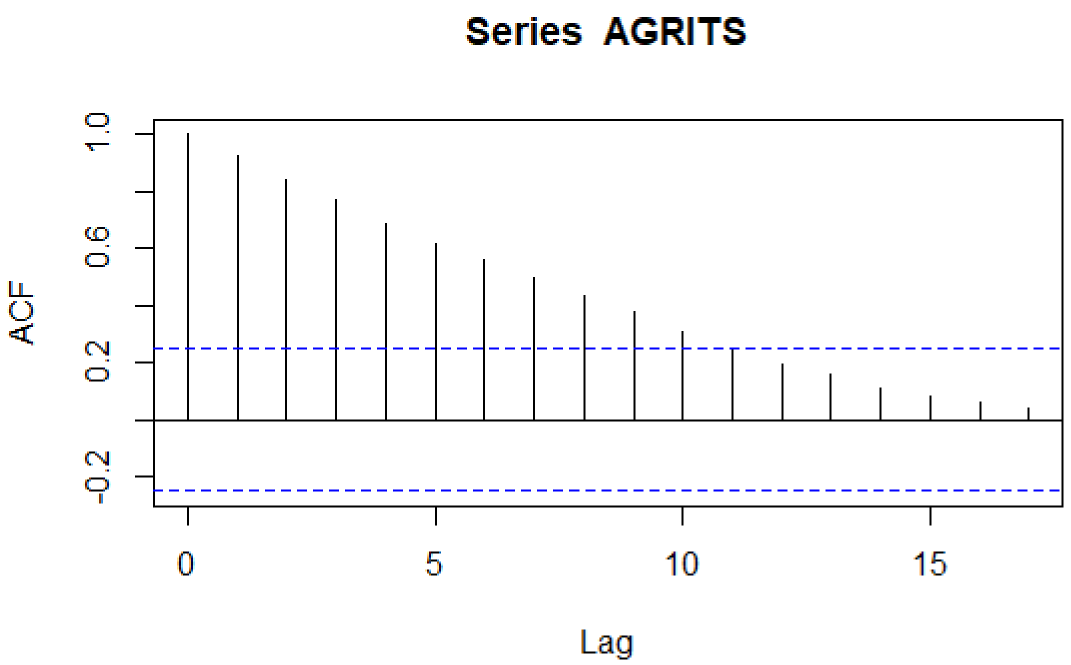
- ***To predict the GDP-Index (GDP-Score) by Agriculture, Industry & Services we can make a model like below:***

GDP=0.997×AGRICULTURE+0.991×INDUSTRY+0.998×SERVICES

```
> ### Forecasting AGRICULTURE for the year 2021-2030
> library(tseries)
> library(forecast)
> library(readxl)
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-Industry.xlsx")
> AGRITS=ts(DATA$`Agriculture(Billion
US$)`,start=min(DATA$Year),end=max(DATA$Year),frequency=1)
> plot(AGRITS)
```



```
> acf(AGRITS)
```



```
> pacf(AGRITS)
> adf.test(AGRITS)

        Augmented Dickey-Fuller Test

data:  AGRITS
Dickey-Fuller = 0.75727, Lag order = 3, p-value = 0.99
alternative hypothesis: stationary

Warning message:
In adf.test(AGRITS) : p-value greater than printed p-value
```

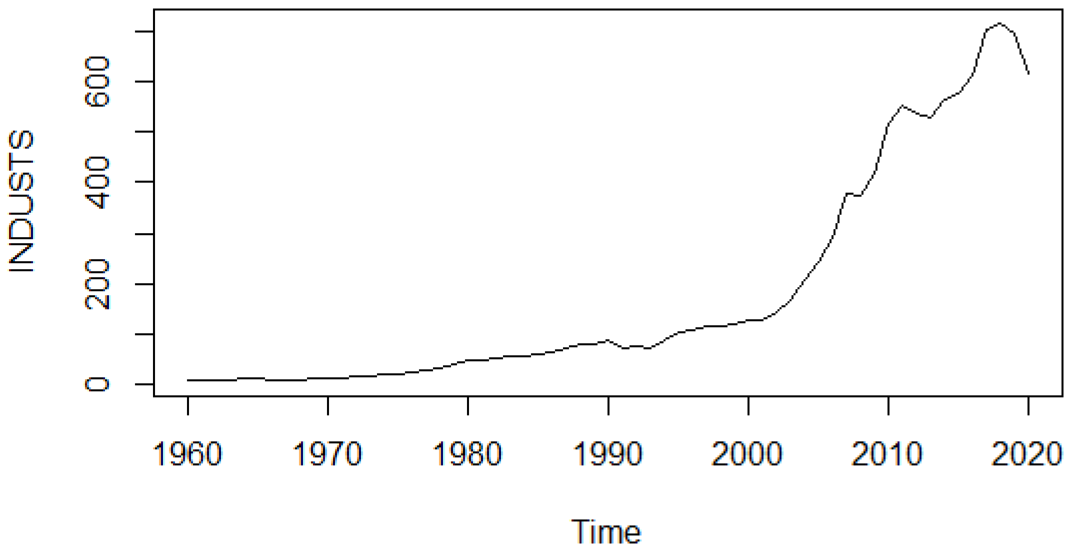
- *Here p-value is 0.99 so it is not less than 0.05 so our data is not stationary. So now we will fit ARIMA model to forecast our Agriculture.*

```
> AGRIMODEL=auto.arima(AGRITS,ic="aic",trace=TRUE)

 ARIMA(2,2,2)                    : 488.2244
 ARIMA(0,2,0)                    : 524.709
 ARIMA(1,2,0)                    : 503.4658
 ARIMA(0,2,1)                    : 485.9775
 ARIMA(1,2,1)                    : 485.1356
 ARIMA(2,2,1)                    : 486.394
 ARIMA(1,2,2)                    : 486.6867
 ARIMA(0,2,2)                    : 484.6959
 ARIMA(0,2,3)                    : 486.6826
 ARIMA(1,2,3)                    : 488.571

 Best model: ARIMA(0,2,2)

> AGRIMODEL
```

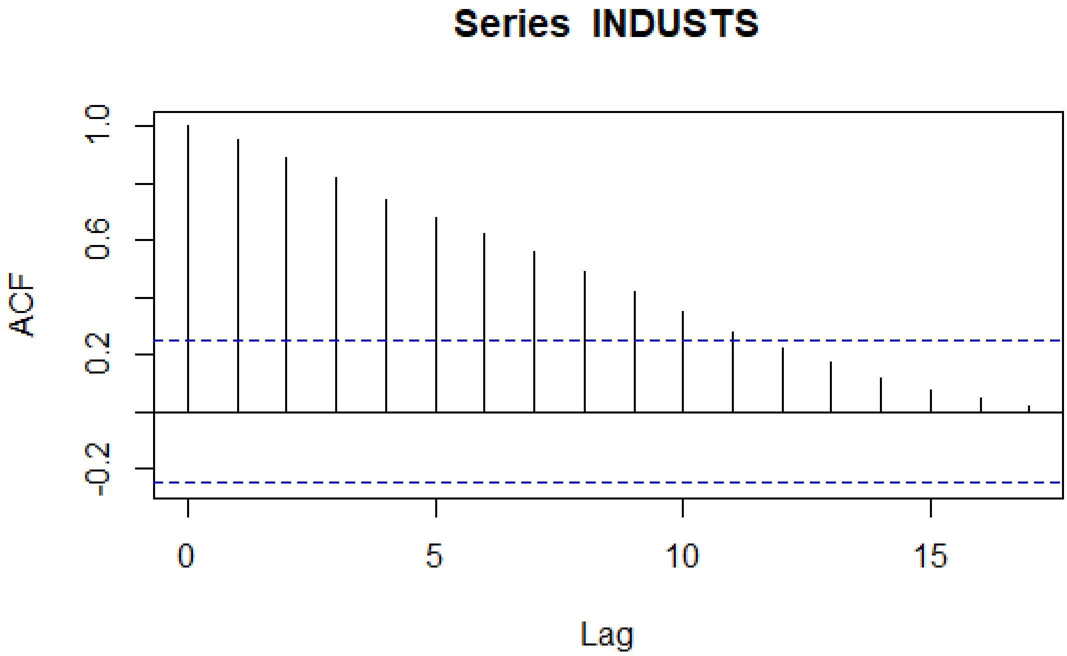
```
Series: AGRITS
ARIMA(0,2,2)

Coefficients:
         ma1     ma2
     -1.0671  0.2399
s.e.   0.1295  0.1232

sigma^2 = 197.4:  log likelihood = -239.35
AIC=484.7   AICc=485.13   BIC=490.93
```

- *No auto regressive part presenting in time series, here second order difference of Agriculture is stationary, only MA portion is contributing in GDP and the corresponding coefficients are also given above.*

```
> forecast(AGRIMODEL,h=10)
> AFOR=data.frame(forecast(AGRIMODEL,h=10))
> AGRICULTURE=AFOR$Point.Forecast
> AGRICULTURE
 [1] 507.5703 530.0581 552.5459 575.0336 597.5214 620.0092 642.4970 664.9848 687.4726
[10] 709.9604


> ### FORCASTING INDUSTRY FOR THE YEAR 2021-2030
> library(tseries)
> library(forecast)
> library(readxl)
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-Industry.xlsx")
> INDUSTS=ts(DATA$`Industry(Billion
US$)`,start=min(DATA$Year),end=max(DATA$Year),frequency=1)
> plot(INDUSTS)
```



```
> acf(INDUSTS)
```



```
> pacf(INDUSTS)
> adf.test(INDUSTS)

        Augmented Dickey-Fuller Test

data:  INDUSTS
Dickey-Fuller = -1.6088, Lag order = 3, p-value = 0.733
alternative hypothesis: stationary
```

- *Here p-value is 0.99 so it is not less than 0.05 so our data is not stationary. So now we will fit ARIMA model to forecast our* ***INDUSTRY****.*

```
> INDUSMODEL=auto.arima(INDUSTS,ic="aic",trace=TRUE)

 ARIMA(2,2,2)                    : Inf
 ARIMA(0,2,0)                    : 554.4518
 ARIMA(1,2,0)                    : 554.6342
 ARIMA(0,2,1)                    : 551.1236
 ARIMA(1,2,1)                    : 547.4027
 ARIMA(2,2,1)                    : 548.4162
 ARIMA(1,2,2)                    : 547.1756
 ARIMA(0,2,2)                    : 545.7389
 ARIMA(0,2,3)                    : 547.6301
 ARIMA(1,2,3)                    : Inf

 Best model: ARIMA(0,2,2)

> INDUSMODEL
Series: INDUSTS
ARIMA(0,2,2)
```

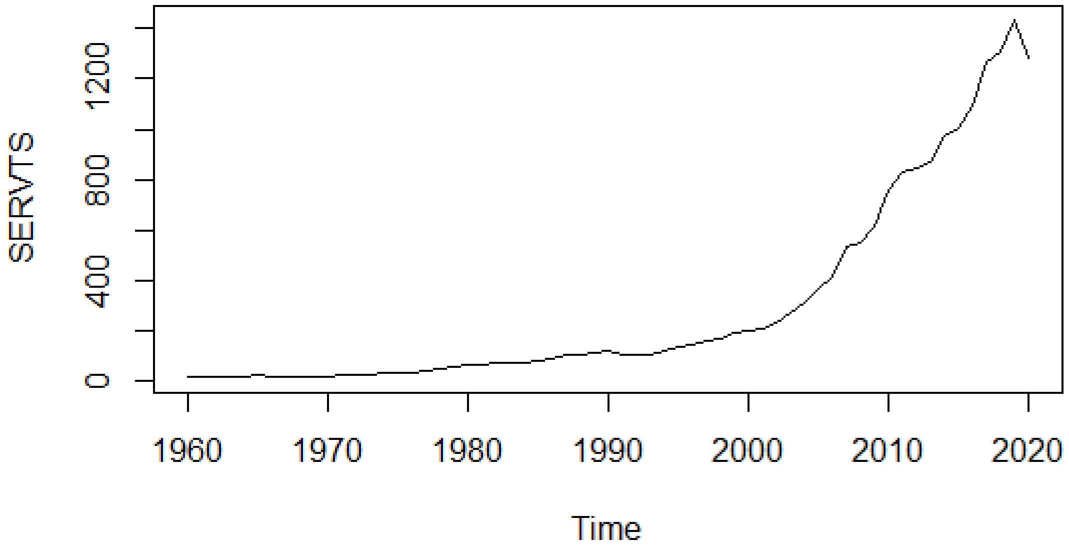
```
Coefficients:
         ma1      ma2
     -0.4033  -0.4845
s.e.   0.1601   0.1770

sigma^2 = 555.4:  log likelihood = -269.87
AIC=545.74   AICc=546.18   BIC=551.97
```

- *No auto regressive part presenting in time series, here second order difference of Industry is stationary, only MA portion is contributing in GDP and the corresponding coefficients are also given above.*

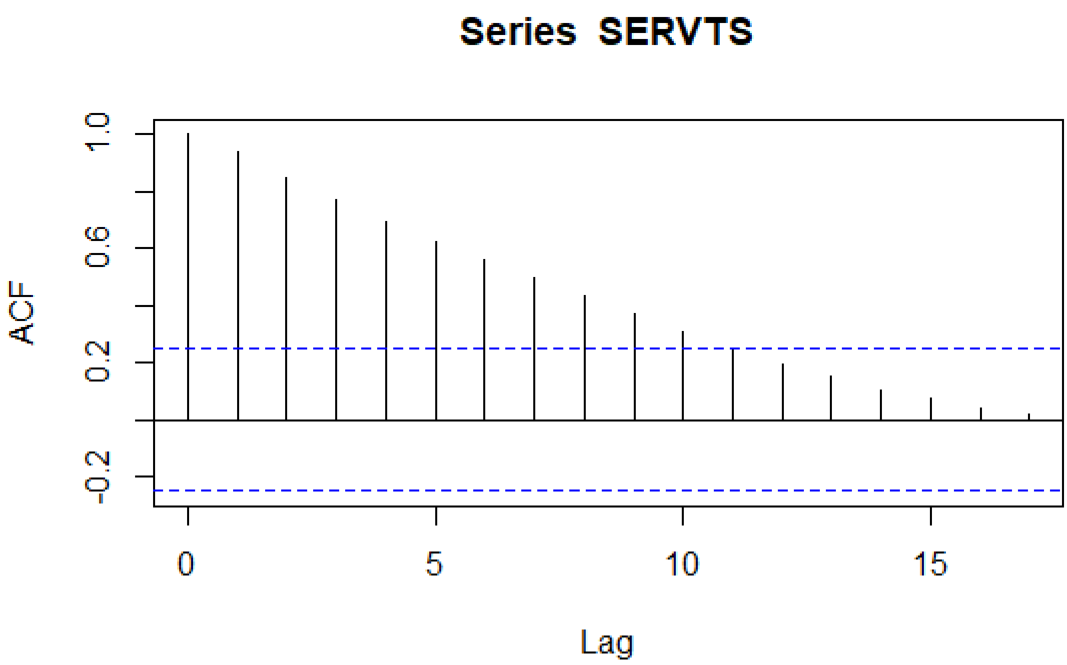
```
> forecast(INDUSMODEL,h=10)
> IFOR=data.frame(forecast(INDUSMODEL,h=10))
> INDUSTRY=IFOR$Point.Forecast
> INDUSTRY
 [1] 585.5778 597.3239 609.0700 620.8161 632.5622 644.3083 656.0544 667.8005 679.5466
[10] 691.2927
```

> ### FORCASTING SERVICES FOR THE YEAR 2021-2030

```
> library(tseries)
> library(forecast)
> library(readxl)
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-Industry.xlsx")
> SERVTS=ts(DATA$`Services(Billion
US$)`,start=min(DATA$Year),end=max(DATA$Year),frequency=1)
> plot(SERVTS)
```



```
> acf(SERVTS)
```



```
> pacf(SERVTS)
> adf.test(SERVTS)

        Augmented Dickey-Fuller Test

data:  SERVTS
Dickey-Fuller = 0.23418, Lag order = 3, p-value = 0.99
alternative hypothesis: stationary

Warning message:
In adf.test(SERVTS) : p-value greater than printed p-value
```

- *Here p-value is 0.99 so it is not less than 0.05 so our data is not stationary. So now we will fit ARIMA model to forecast our* ***Services****.*

```
> SERVMODEL=auto.arima(SERVTS,ic="aic",trace=TRUE)

 ARIMA(2,2,2)             : 614.5431
 ARIMA(0,2,0)             : 632.5982
 ARIMA(1,2,0)             : 615.896
 ARIMA(0,2,1)             : 611.261
 ARIMA(1,2,1)             : 611.5488
 ARIMA(0,2,2)             : 611.954
 ARIMA(1,2,2)             : 612.7175

 Best model: ARIMA(0,2,1)

> SERVMODEL
Series: SERVTS
ARIMA(0,2,1)
```

```
Coefficients:
        ma1
     -0.8401
s.e.   0.0715

sigma^2 = 1722:  log likelihood = -303.63
AIC=611.26   AICc=611.48   BIC=615.42
```

- *No auto regressive part presenting in time series, here second order difference of Services is stationary, only MA portion is contributing in GDP and the corresponding coefficients are also given above.*

```
> forecast(SERVMODEL,h=10)
> SFOR=data.frame(forecast(SERVMODEL,h=10))
> SERVICES=SFOR$Point.Forecast
> SERVICES
 [1] 1325.803 1369.152 1412.501 1455.850 1499.198 1542.547 1585.896 1629.245 1672.593
[10] 1715.942


> ## Now we will predict GDP-Index with our predicted value of agriculture & industry & services
> GDP_INDEX=c()
> for (i in 1:10) {
+   GDP_INDEX[i]=0.997*AGRICULTURE[i]+0.991*INDUSTRY[i]+0.998*SERVICES[i]
+ }
> GDP_INDEX
 [1] 2409.507 2486.830 2564.152 2641.475 2718.798 2796.121 2873.444 2950.766 3028.089
[10] 3105.412
```

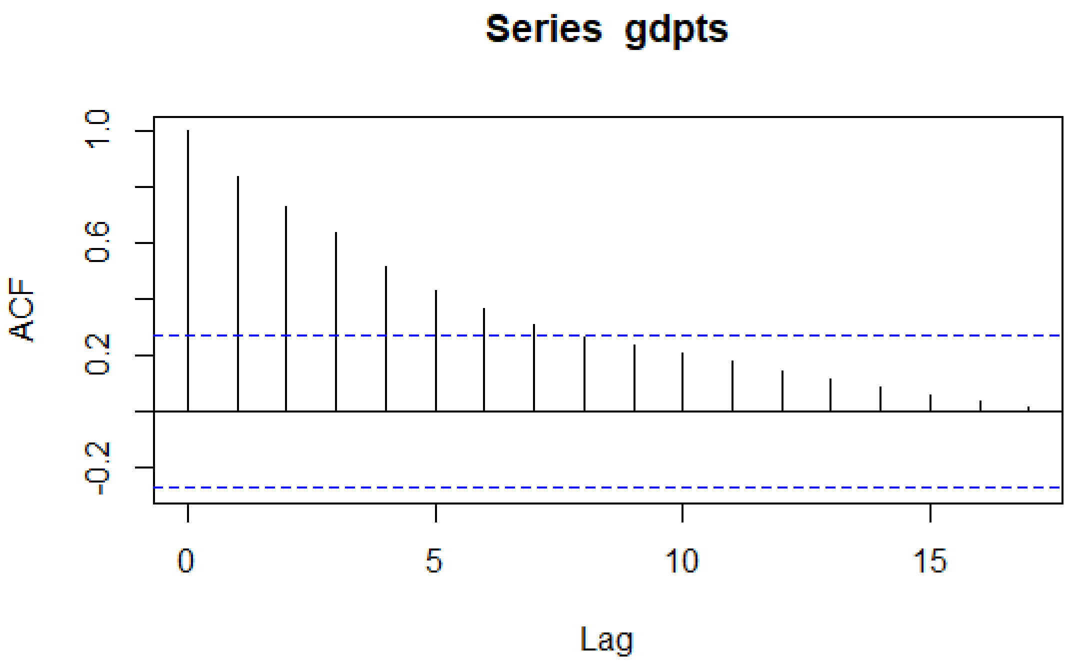
Here, a model will be fitted using the forecasted GDP data (referred to as GDP1, obtained through Time Series forecasting) and the predicted GDP data derived from the three major sectors of GDP (referred to as GDP2).

The analysis involves going back 10 years to 2010, and then forecasting and predicting GDP for the period 2011-2020, focusing on the three major sectors of GDP in India: Agriculture, Services, and Industry. These predictions will be compared with the actual GDP data for 2011-2020. The aim is to gather insights that will assist in predicting GDP for the years 2021-2030.

The forecasting of GDP for the years 2011 to 2020 will now commence, based on GDP data from 1960 to 2010, using Time Series forecasting methods.

Code and results:

```
> ### Forecasting GDP for year 2011-2020
> # Loading the Data Set of GDP
> library(tseries)
> library(forecast)
> library(readxl)
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-Industry.xlsx")
> # Here we are taking GDP data and converting the data set into time series form
> gdpts=ts(DATA$`GDP(in Billion US$)`,start=1960,end=2010,frequency=1)
> # now checking whether the data set is stationary or not
> acf(gdpts)
```



```
> pacf(gdpts)
> adf.test(gdpts)

        Augmented Dickey-Fuller Test

data:  gdpts
Dickey-Fuller = 1.526, Lag order = 3, p-value = 0.99
alternative hypothesis: stationary

Warning message:
In adf.test(gdpts) : p-value greater than printed p-value
```

- *Here p-value is 0.99 so it is not less than 0.05 so our data is not stationary. So now we will fit ARIMA model to forecast our **Services**.*

```
> # Fitting ARIMA model
> gdpmodel=auto.arima(gdpts,ic="aic",trace=TRUE)

 ARIMA(2,2,2)                    : Inf
 ARIMA(0,2,0)                    : 548.3737
 ARIMA(1,2,0)                    : 541.3545
 ARIMA(0,2,1)                    : 533.0167
 ARIMA(1,2,1)                    : 533.7552
 ARIMA(0,2,2)                    : 529.286
 ARIMA(1,2,2)                    : Inf
 ARIMA(0,2,3)                    : 528.4155
 ARIMA(1,2,3)                    : Inf
 ARIMA(0,2,4)                : Inf
 ARIMA(1,2,4)                : Inf

 Best model: ARIMA(0,2,3)

> gdpmodel
Series: gdpts
ARIMA(0,2,3)

Coefficients:
         ma1      ma2     ma3
     -0.8299  -0.0704  0.3457
s.e.   0.1458   0.2068  0.1464

sigma^2 = 2490:  log likelihood = -260.21
AIC=528.42   AICc=529.32   BIC=535.98
```

- *No auto regressive part presenting in time series, here second order difference of GDP is stationary, only MA portion is contributing in GDP and the corresponding coefficients are also given above.*

```
> # Forecast the GDP
> forecast(gdpmodel,h=10)
> gdpfor=data.frame(forecast(gdpmodel,h=10))
> gdp1=gdpfor$Point.Forecast
> gdp1
 [1] 1841.967 2004.567 2212.129 2419.692 2627.254 2834.817 3042.380 3249.942 3457.505
[10] 3665.068
```

Forecast GDP for the year 2011-2020 (gdp1)										
Year	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	**2018**	**2019**	**2020**
GDP	1841.967	2004.567	2212.129	2419.692	2627.254	2834.817	3042.380	3249.942	3457.505	3665.068

The prediction of the GDP-Index (GDP-Score) for the years 2011-2020 will be undertaken using data from the three major sectors of GDP: Agriculture, Services, and Industry. The original data for these sectors from 2011-2020 has been collected and will be used to proceed with the prediction.

Code and results:

```
> # Loading the Data Set of Agriculture & sevices & industry
> library(readxl)
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-Industry.xlsx")
> # Here we are taking Agriculture data for the year 2011-2020
> agriculture=tail(DATA$`Agriculture(Billion US$)`,10)
> agriculture
 [1] 313.4183 307.8728 318.3982 342.4094 340.2453 375.5165 439.0392 431.3715 478.7266
[10] 480.4125
> # Here we are taking industry data for the year 2011-2020
> industry=tail(DATA$`Industry(Billion US$)`,10)
> industry
 [1] 549.8625 537.2993 527.3995 563.9500 575.2770 610.8528 702.6400 712.6311 694.0179
[10] 616.9520
> # Here we are taking services data for the year 2011-2020
> services=tail(DATA$`Services(Billion US$)`,10)
> services
 [1]  828.4330  846.2183  867.0644  975.1612 1005.1741 1095.7533 1263.9793 1310.3008
 [9] 1431.2186 1282.4547

> # Now we will predict GDP-Index with our existing data of agriculture & industry & services
(2011-2020)
> gdp_index=c()
> for (i in 1:10) {
+   gdp_index[i]=0.997*agriculture[i]+0.991*industry[i]+0.998*services[i]
+ }
> gdp_index
 [1] 1684.168 1683.939 1705.426 1873.467 1912.488 2073.307 2395.490 2443.975 2593.418
[10] 2370.260
```

```
> ### Our original GDP data for the year 2011-2020
> library(readxl)
> DATA=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GDP-Agri-Service-Industry.xlsx")
> gdp=tail(DATA$`GDP(in Billion US$)`,10)
> gdp
 [1] 1823.05 1827.64 1856.72 2039.13 2103.59 2294.80 2651.47 2701.11 2870.50 2622.98
```

Currently, an adjustment factor (denoted as R) is being calculated to correct the GDP from the predicted GDP index. This adjustment will be used to determine the corrected GDP for the years 2011-2020.

Code and results:

```
> #### Now calculating adjustment factor
> r=c()
> for (i in 1:10) {
+   r[i]=gdp[i]/gdp_index[i]
+ }
> r
 [1] 1.082463 1.085336 1.088713 1.088426 1.099923 1.106831 1.106859 1.105212 1.106840
[10] 1.106621
> R=mean(r)
> R
[1] 1.097723
```

- *So our Adjustment Factor (R) is 1.097723*

```
> ### Now get our Correction GDP for the 2011-2020 predicting with the three factors
> gdp2=c()
> for (i in 1:10) {
+   gdp2[i]=gdp_index[i]*R
+ }
```

```
> gdp2
 [1] 1848.749 1848.497 1872.085 2056.547 2099.381 2275.916 2629.583 2682.806 2846.854
[10] 2601.888
```

Predicted GDP for the year 2011-2020 (gdp2) by three sectors										
Year	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	**2018**	**2019**	**2020**
GDP	1848.749	1848.497	1872.085	2056.547	2099.381	2275.916	2629.583	2682.806	2846.854	2601.888

The analysis involves two types of forecasted GDP values for the years 2011-2020: "gdp1" (GDP forecasted using time series analysis) and "gdp2" (GDP predicted using three factors). Additionally, the actual GDP data for the same period, referred to as "gdp," is available. The objective is to determine the contribution percentage of "gdp1" and "gdp2" to the original GDP for the years 2011-2020. The equation used for this analysis is:

SUM(original GDP) = X * SUM(GDP by TS forecasting) + (1 - X) * SUM(predicted GDP by 3 factors)

Here, X represents the contribution percentage to be determined.

Code and results:

```
> x=(sum(gdp)-sum(gdp2))/(sum(gdp1)-sum(gdp2))
> x
[1] 0.006245015
```

- *So, our contribution percentage X=0.006245015*

The GDP for the years 2011-2020 is predicted using gdp1 and gdp2, along with the predicted value of X, referred to as gdp3.

Code and results:

```
> gdp3=c()
> for (i in 1:10) {
+   gdp3[i]=x*gdp1[i]+(1-x)*gdp2[i]
+ }
> gdp3
 [1] 1848.707 1849.472 1874.208 2058.815 2102.678 2279.406 2632.161 2686.348 2850.667
[10] 2608.528
```

Predicted GDP for the year 2011-2020 by gdp1 & gdp2 (gdp3)										
Year	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2017**	**2018**	**2019**	**2020**
GDP	1848.707	1849.472	1874.208	2058.815	2102.678	2279.406	2632.161	2686.348	2850.667	2608.528

Now, there is a keen interest in determining which model among gdp1, gdp2, and gdp3 provides the most accurate predictions. To achieve this, Root Mean Square Error (RMSE) will be employed. The actual GDP data for the years 2011-2020 and the predicted data from the three models are available. RMSE will be calculated individually for each model, and the model with the lowest RMSE will be deemed the best fit.

Code and results:

```
> ##### Calculating Root Mean Square Error (RMSE)
> # Importing the required package
> library(Metrics)
> # Calculating Root Mean Square Error (RMSE) for our gdp1
> RMSE1=rmse(gdp,gdp1)
> RMSE1
[1] 524.709
> # Calculating Root Mean Square Error (RMSE) for our gdp2
> RMSE2=rmse(gdp,gdp2)
> RMSE2
[1] 19.56353
> # Calculating Root Mean Square Error (RMSE) for our gdp3
> RMSE3=rmse(gdp,gdp3)
> RMSE3
[1] 18.05275
```

- *Here we can see that the model-3, i.e. gdp3 has the least RMSE. So we can say that our model-3 (gdp3) has the best fitting and this model will give us more accurate prediction than other two.*

Now, the correction of GDP (2021-2030) is computed from the GDP_INDEX (2021-2030) using the adjustment factor (R) that has been previously calculated.

Code and results:

```
> GDP2=c()
> for (i in 1:10) {
+   GDP2[i]=GDP_INDEX[i]*R
+ }
> GDP2
 [1] 2644.970 2729.849 2814.728 2899.607 2984.486 3069.365 3154.244 3239.123 3324.002
[10] 3408.881
```

Predicted GDP for the year 2021-2030 (GDP2) by three sectors

Year	**2021**	**2022**	**2023**	**2024**	**2025**	**2026**	**2027**	**2028**	**2029**	**2030**
GDP	2644.970	2729.849	2814.728	2899.607	2984.486	3069.365	3154.244	3239.123	3324.002	3408.881

Now, all the required information and conditions are in place to make more accurate predictions of India's GDP for the years 2021 to 2030.

Code and results:

```
> ## Now we will predict final GDP for the year 2021-2030
> GDP=c()
> for (i in 1:10) {
```

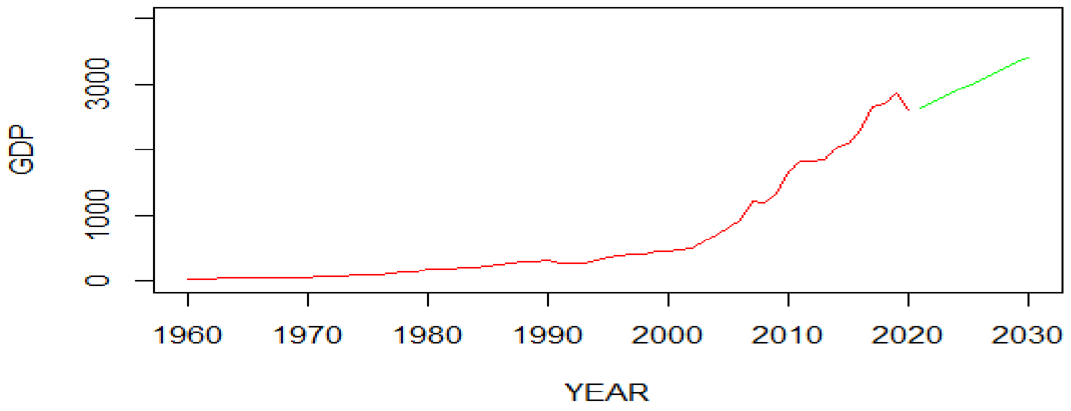
```
+   GDP[i]=x*GDP1[i]+(1-x)*GDP2[i]
+ }
> GDP
 [1] 2645.341 2730.197 2815.054 2899.911 2984.768 3069.624 3154.481 3239.338 3324.195
[10] 3409.051
```

Final Forecast GDP for the year 2021-2030 (GDP)										
Year	**2021**	**2022**	**2023**	**2024**	**2025**	**2026**	**2027**	**2028**	**2029**	**2030**
GDP	2645.341	2730.197	2815.054	2899.911	2984.768	3069.624	3154.481	3239.338	3324.195	3409.051

Plotting the predicted GDP alongside the previous GDP.

Code and results:

```
> dfGDP=data.frame(YEAR=c(2021:2030),GDP)
> plot(GDPTS,xlab="YEAR",ylab="GDP",xlim=c(1960,2030),ylim=c(0,4000),col="red")
> lines(dfGDP,col="green")
```



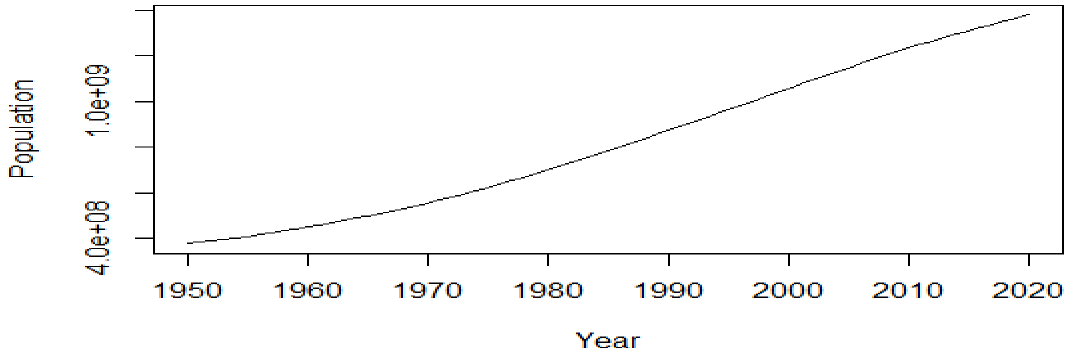
The focus now shifts to predicting the population from 2021 to 2030 using the Incremental Increase Method. The method's formula is provided below:

$$\textbf{Population} = \textbf{Initial Population} + (\mathbf{I} \times \mathbf{AP}) + \frac{I\times(1+I)}{2} \times \mathbf{AIP}$$

Where I = Number of years, AP = Average increases in population, AIP = Average incremental increase in population

Code and results:

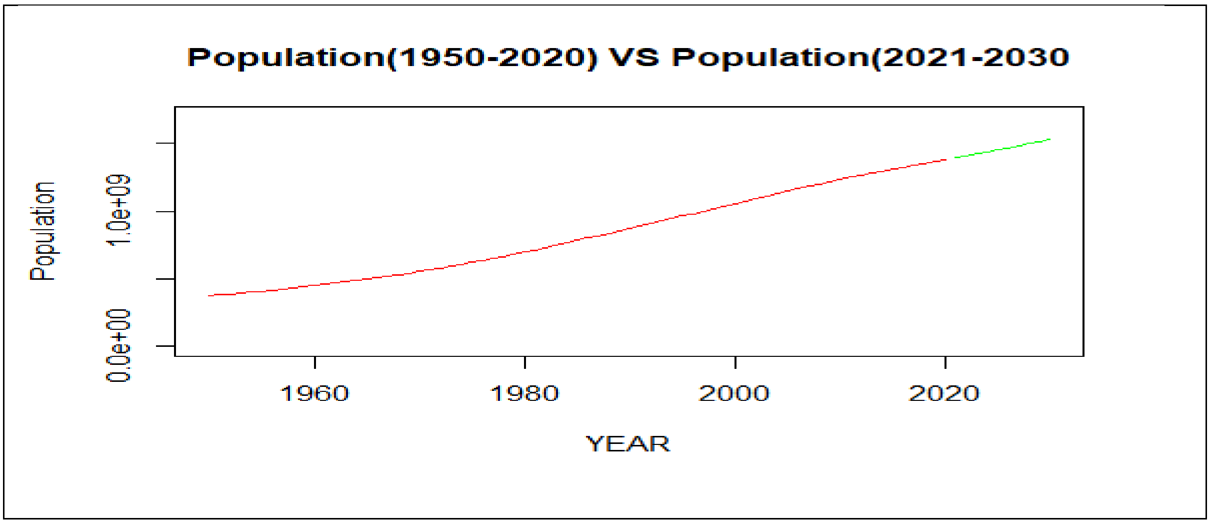
```
> ### Predicting Population (2021-2030)
> library(readxl)
> POPULATION=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/Population.xlsx")
> plot(POPULATION,type="l")
```



```
> P=POPULATION$Population
> IP=c()
> for (i in 2:71) {
+   IP[i]=P[i]-P[i-1]
+ }
> IP
> IPP=na.omit(IP)
> IPP
> AVE_IP=mean(IPP)
> AVE_IP
[1] 14338274
> IIP=c()
> for (i in 3:71) {
+   IIP[i]=IP[i]-IP[i-1]
+ }
> IIP
> IIPP=na.omit(IIP)
> IIPP
> AVE_IIP=mean(IIPP)
> AVE_IIP
[1] 109201.2
> FORECAST_POPULATION=c()
> for (i in 1:10) {
+   FORECAST_POPULATION[i]=1380004385+i*AVE_IP+(i*(1+i)/2)*AVE_IIP
+ }
> FORECAST_POPULATION
 [1] 1394451860 1409008537 1423674414 1438449493 1453333773 1468327255 1483429937
 [8] 1498641821 1513962906 1529393192
```

Now plotting our predicted Population(2021-2030) vs previous Population(1950-2020).

```
> dfPOPULATION=data.frame(YEAR=c(2021:2030),FORECAST_POPULATION)
> plot(POPULATION,xlab="YEAR",ylab="Population",xlim=c(1950,2030),ylim=c(0,1700000000),
+     col="red",type="l",main="Population(1950-2020) VS Population(2021-2030")
> lines(dfPOPULATION,col="green")
```



Predicted Population for the year 2021-2030										
Year	**2021**	**2022**	**2023**	**2024**	**2025**	**2026**	**2027**	**2028**	**2029**	**2030**
GDP	1394451860	1409008537	1423674414	1438449493	1453333773	1468327255	1483429937	1498641821	1513962906	1529393192

```
> ## Population of year 2020-2030
> population=c(tail(P,1),FORECAST_POPULATION)
> population
 [1] 1380004385 1394451860 1409008537 1423674414 1438449493 1453333773 1468327255
 [8] 1483429937 1498641821 1513962906 1529393192
```

```
> ## GDP for year 2020-2030
> G=DATA$`GDP(in Billion US$)`
> GDP_20_30=c(tail(G,1),GDP)
> GDP_20_30
 [1] 2622.980 2645.341 2730.197 2815.054 2899.911 2984.768 3069.624 3154.481 3239.338
[10] 3324.195 3409.051
```

Now, the analysis focuses on predicting GNI per capita using Multiple Linear Regression, where GNI serves as the dependent variable, while population and GDP act as the independent variables.

Code and results:

```
> ### Predicting GNI per capita
> # Loading the Data Set of GNI per capita
> GNI=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/GNI.xlsx")
> View(GNI)
> GNIPC=lm(GNI_Per_Capita ~ Population + GDP, data = GNI)
> summary(GNIPC)

Call:
```

```
lm(formula = GNI_Per_Capita ~ Population + GDP, data = GNI)

Residuals:
    Min      1Q  Median      3Q     Max
-391.79  -73.47   12.36   91.87  246.45

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -6.437e+02  4.998e+02  -1.288 0.208721
Population   2.320e-06  5.284e-07   4.391 0.000156 ***
GDP          1.365e+00  9.538e-02  14.308 4.02e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 151.6 on 27 degrees of freedom
Multiple R-squared:  0.9904,       Adjusted R-squared:  0.9897
F-statistic:  1397 on 2 and 27 DF,  p-value: < 2.2e-16

> summary(GNIPC)$coefficient
                Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) -6.437063e+02 4.998235e+02 -1.287867 2.087213e-01
Population   2.319886e-06 5.283838e-07  4.390532 1.564562e-04
GDP          1.364729e+00 9.538359e-02 14.307793 4.015757e-14
```

- *Now we forecasting GNI by our forecast value of population and GDP (2020-2030). Our Multiple Linear Regression will be:*
  *GNI_Per_Capita=(-6.437063e+02)+2.319886e-06*Population+1.364729e+00*GDP*

```
> GNI_Per_Capita=c()
> for (i in 1:11) {
+   GNI_Per_Capita[i]=(-6.437063e+02)+(2.319886e-
06)*population[i]+(1.364729e+00)*GDP_20_30[i]
+ }
> GNI_Per_Capita
 [1] 6137.403 6201.436 6351.012 6500.842 6650.925 6801.261 6951.851 7102.694 7253.790
[10] 7405.140 7556.743
```

Now, the Income Index is calculated from the predicted GNI per capita using the following formula.

$$\textbf{Income Index (InI)} = \frac{\ln\left(\mathrm{GNI}_{\mathrm{Per}_{\mathrm{Capita}}}\right) - \ln(100)}{\ln(75000) - \ln(100)}$$

Code and results:

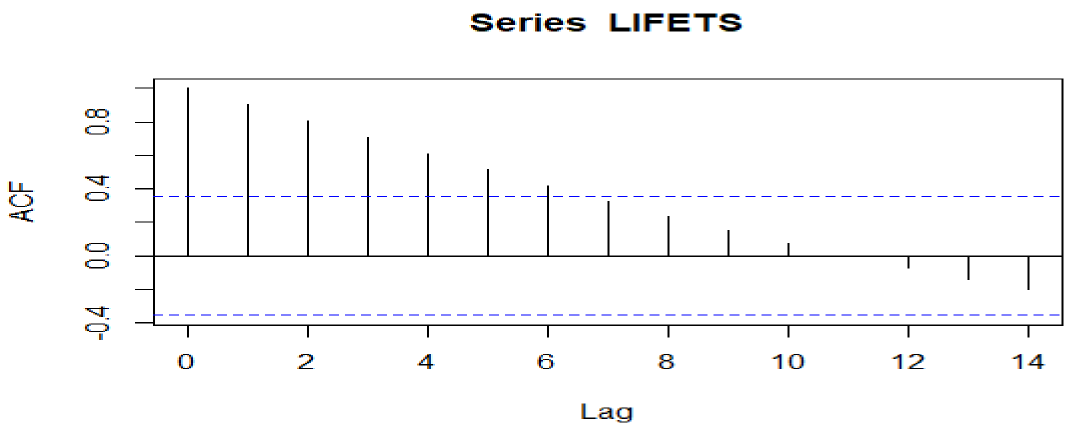
```
> ### Income Index (InI)
> library(SciViews)
> InI=c()
> for (i in 1:11) {
+   InI[i]=(ln(GNI_Per_Capita[i])-ln(100))/(ln(75000)-ln(100))
+ }
> InI
```

```
[1] 0.6218945 0.6234623 0.6270624 0.6305847 0.6340324 0.6374088 0.6407169 0.6439595
[9] 0.6471393 0.6502586 0.6533199
```

Now, life expectancy for the years 2020 to 2030 is forecasted using time series forecasting, utilizing a dataset spanning from 1900 to 2019.

Code and results:

```
> ############## Forecast Life Expectancy ###################
> # Loading the Data Set
> library(tseries)
> library(forecast)
> library(readxl)
> LIFE=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/Life expectancy.xlsx")
> # here we are converting the data set into time series form
> LIFETS=ts(LIFE$`Life Expectancy`,start=min(LIFE$Year),end=max(LIFE$Year),frequency=1)
> # now checking whether the data set is stationary or not
> acf(LIFETS)
```



```
> pacf(LIFETS)

> adf.test(LIFETS)

        Augmented Dickey-Fuller Test

data:  LIFETS
Dickey-Fuller = -0.6977, Lag order = 3, p-value = 0.9595
alternative hypothesis: stationary

> # here p-value is 0.99 so it is not less than 0.05 so our data is not stationary
> # ARIMA model
> LIFEMODEL=auto.arima(LIFETS,ic="aic",trace=TRUE)

 ARIMA(2,2,2)                : Inf
 ARIMA(0,2,0)                : -71.20855
 ARIMA(1,2,0)                : -81.70597
 ARIMA(0,2,1)                : -78.92248
 ARIMA(2,2,0)                : -80.11933
 ARIMA(1,2,1)                : -80.0026
 ARIMA(2,2,1)                    : -78.13763

 Best model: ARIMA(1,2,0)

> LIFEMODEL
Series: LIFETS
ARIMA(1,2,0)

Coefficients:
          ar1
      -0.6124
s.e.   0.1504

sigma^2 = 0.003039:  log likelihood = 42.85
AIC=-81.71   AICc=-81.23   BIC=-79.04
> # Forcast the data
> forecast(LIFEMODEL,h=11)
> fLE=data.frame(forecast(LIFEMODEL,h=11))
> FLE=fLE$Point.Forecast
> FLE
 [1] 69.93876 70.21503 70.46833 70.73569 70.99444 71.25846 71.51926 71.78203 72.04359
[10] 72.30590 72.56774
```

Now, the Life Expectancy Index (LEI) is calculated from the predicted life expectancy data (2020-2030). The formula used for determining the Life Expectancy Index (LEI) is:

$$\textbf{Life Expectancy Index (LEI)} = \boldsymbol{\frac{LE-20}{85-20}}$$

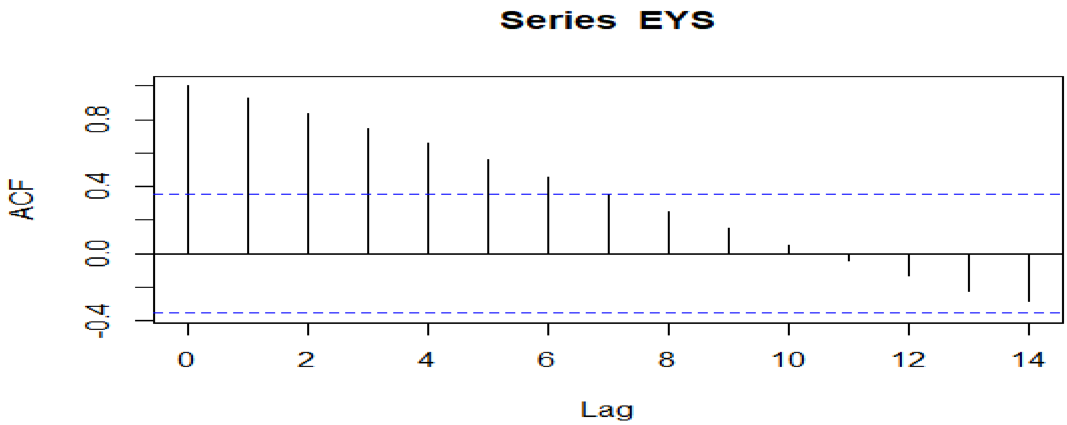
Code and results:

```
> ### Life Expectancy Index (LEI) ###
> LEI=c()
> for (i in 1:11) {
+   LEI[i]=(FLE[i]-20)/(80-20)
+ }
> LEI
 [1] 0.8323127 0.8369171 0.8411388 0.8455948 0.8499073 0.8543077 0.8586543 0.8630339
 [9] 0.8673932 0.8717649 0.8761291
```

Now, the forecast for Expected Year of Schooling and Mean Year of Schooling from the year 2020 to 2030 will be conducted using time series forecasting methods, leveraging data spanning from 1900 to 2019.

Code and results:

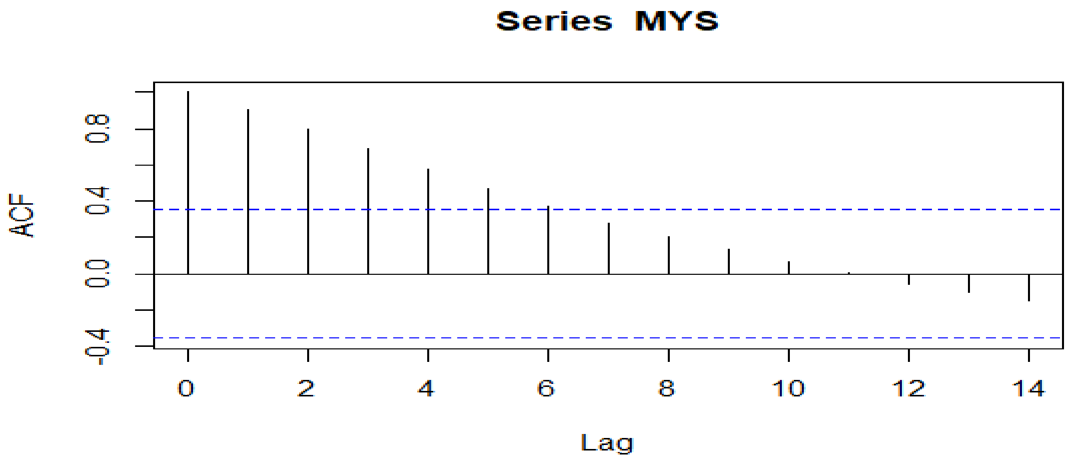
```
> # Loading the Data Set
> EDUCATION=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/Education.xlsx")
> ######### Forecasting Expected Year of Schooling
> EYS=ts(EDUCATION$`Expected years of schooling
(years)`,start=min(EDUCATION$Year),end=max(EDUCATION$Year),frequency=1)
> # now checking whether the data set is stationary or not
> acf(EYS)
```



```
> pacf(EYS)
> adf.test(EYS)

        Augmented Dickey-Fuller Test

data:  EYS
Dickey-Fuller = -2.5251, Lag order = 3, p-value = 0.3709
alternative hypothesis: stationary
```

- *here p-value is 0.99 so it is not less than 0.05 so our data is not stationary.*

```
> # ARIMA model
> EYSMODEL=auto.arima(EYS,ic="aic",trace=TRUE)

 ARIMA(2,1,2) with drift         : -9.329845
 ARIMA(0,1,0) with drift         : -15.38025
 ARIMA(1,1,0) with drift         : -13.57421
 ARIMA(0,1,1) with drift         : -13.57319
 ARIMA(0,1,0)                    : 0.2769611
 ARIMA(1,1,1) with drift         : -13.21303

 Best model: ARIMA(0,1,0) with drift

> EYSMODEL
Series: EYS
ARIMA(0,1,0) with drift

Coefficients:
       drift
      0.1586
s.e.  0.0322

sigma^2 = 0.03109:  log likelihood = 9.69
AIC=-15.38   AICc=-14.92   BIC=-12.65
> # Forcast the data
> forecast(EYSMODEL,h=11)
> fEYS=data.frame(forecast(EYSMODEL,h=11))
> FEYS=fEYS$Point.Forecast
> FEYS
 [1] 12.35862 12.51724 12.67586 12.83448 12.99310 13.15172 13.31034 13.46897 13.62759
[10] 13.78621 13.94483


> ######### Forecasting Mean Year of Schooling
> MYS=ts(EDUCATION$`Mean years of schooling
(years)`,start=min(EDUCATION$Year),end=max(EDUCATION$Year),frequency=1)
> # now checking whether the data set is stationary or not
> acf(MYS)
```



```
> pacf(MYS)
> adf.test(MYS)

        Augmented Dickey-Fuller Test

data:  MYS
Dickey-Fuller = -2.9517, Lag order = 3, p-value = 0.2071
alternative hypothesis: stationary
```

- *Here p-value is 0.99 so it is not less than 0.05 so our data is not stationary.*

```
> # ARIMA model
> MYSMODEL=auto.arima(MYS,ic="aic",trace=TRUE)

 ARIMA(2,1,2) with drift         : Inf
 ARIMA(0,1,0) with drift         : -59.88513
 ARIMA(1,1,0) with drift         : -60.64695
 ARIMA(0,1,1) with drift         : -61.39169
 ARIMA(0,1,0)                    : -27.68774
 ARIMA(1,1,1) with drift         : -59.42605
 ARIMA(0,1,2) with drift         : -59.43547
 ARIMA(1,1,2) with drift         : Inf
 ARIMA(0,1,1)                    : -42.3751

 Best model: ARIMA(0,1,1) with drift

> MYSMODEL
Series: MYS
```

```
ARIMA(0,1,1) with drift

Coefficients:
         ma1   drift
      0.3742  0.1195
s.e.  0.1778  0.0191

sigma^2 = 0.006125:  log likelihood = 33.7
AIC=-61.39   AICc=-60.43   BIC=-57.29
> # Forcast the data
> forecast(MYSMODEL,h=11)
> fMYS=data.frame(forecast(MYSMODEL,h=11))
> FMYS=fMYS$Point.Forecast
> FMYS
 [1] 6.589382 6.708845 6.828309 6.947773 7.067236 7.186700 7.306164 7.425627 7.545091
[10] 7.664555 7.784019
```

Now, the calculation of the Expected Year of Schooling Index (EYSI) and Mean Year of Schooling Index (MYSI) is conducted based on the predicted Expected Year of Schooling and Mean Year of Schooling (2020-2030) respectively. The formula used to compute the Expected Year of Schooling Index (EYSI) is as follows:

Expected Years of Schooling Index (EYSI) = $\frac{\boldsymbol{EYS}}{\mathbf{18}}$

The formula for calculating Mean Year of Schooling Index (MYSI) is-

Mean Years of Schooling Index (MYSI) = $\frac{\boldsymbol{MYS}}{\mathbf{15}}$

Code and results:

```
> ### Expected Year of Schooling Index (EYSI)
> EYSI=c()
> for (i in 1:11) {
+   EYSI[i]=FEYS[i]/18
+ }
> EYSI
 [1] 0.6865900 0.6954023 0.7042146 0.7130268 0.7218391 0.7306513 0.7394636 0.7482759
 [9] 0.7570881 0.7659004 0.7747126


> ### Mean Year of Schooling Index (MYSI)
> MYSI=c()
> for (i in 1:11) {
+   MYSI[i]=FMYS[i]/18
+ }
> MYSI
 [1] 0.3660768 0.3727136 0.3793505 0.3859874 0.3926242 0.3992611 0.4058980 0.4125349
 [9] 0.4191717 0.4258086 0.4324455
```

Currently, the Education Index (EI) is being calculated using the Expected Year of Schooling Index (EYSI) and the Mean Year of Schooling Index (MYSI). The formula for this calculation is provided below:

$$\textbf{Education Index (EI)} = \frac{\boldsymbol{EYSI+MYSI}}{\mathbf{2}}$$

Code and results:

```
> ### Education Index (EI)
> EI=c()
> for (i in 1:11) {
+   EI[i]=(EYSI[i]+MYSI[i])/2
+ }
> EI
 [1] 0.5263334 0.5340580 0.5417825 0.5495071 0.5572317 0.5649562 0.5726808 0.5804054
 [9] 0.5881299 0.5958545 0.6035791
```

Now, the prediction of HDI for the years 2020 to 2030 is based on the Education Index (EI), Life Expectancy Index (LEI), and Income Index (InI). The formula used for predicting HDI is:

$$\textbf{HDI}=\sqrt[3]{\boldsymbol{InI \times EI \times LEI}}$$

Code and results:

```
> ################## Predict HDI for the year 2020-2030 ##########################
> FORECAST_HDI=c()
> for (i in 1:11) {
+   FORECAST_HDI[i]=(LEI[i]*EI[i]*InI[i])^(1/3)
+ }
> FORECAST_HDI
 [1] 0.6482682 0.6531712 0.6586699 0.6641910 0.6696373 0.6750704 0.6804545 0.6858134
 [9] 0.6911341 0.6964264 0.7016858
```

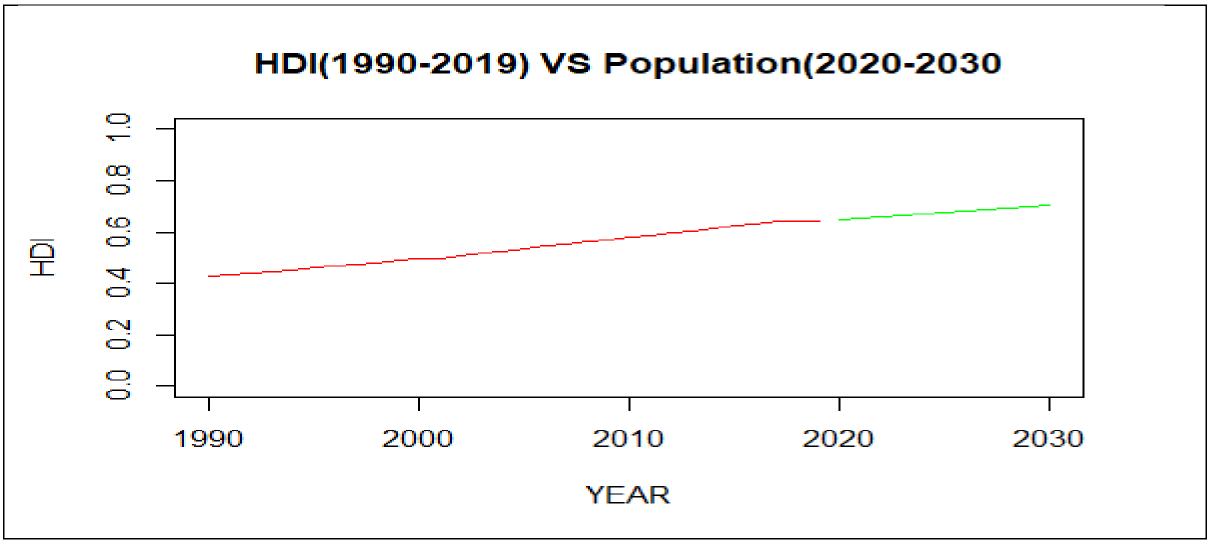
Predicted HDI (2020-2030)

Year	**2020**	**2021**	**2022**	**2023**	**2024**	**2025**	**2026**	**2027**	**2028**	**2029**	**2030**
HDI	0.6482682	0.6531712	0.6586699	0.6641910	0.6696373	0.6750704	0.6804545	0.6858134	0.6911341	0.6964264	0.7016858

They are now plotting the predicted HDI (2020-2030) against the previous HDI (1990-2019).

Code and results:

```
> ######## Plotting our predicted HDI(2020-2030) VS previous HDI (1990-2019)
> library(readxl)
> HDI=read_excel("C:/Users/Ranadip/Desktop/PROJECT/DATA/HDI.xlsx")
> dfHDI=data.frame(YEAR=c(2020:2030),FORECAST_HDI)
> plot(HDI,xlab="YEAR",ylab="HDI",xlim=c(1990,2030),ylim=c(0,1),
+      col="red",type="l",main="HDI(1990-2019) VS Population(2020-2030")
> lines(dfHDI,col="green")
```


HDI(1990-2019) VS Population(2020-2030
HDI
0.0
0.2
0.4
0.6
0.8
1.0
1990
2000
2010
2020
2030
YEAR

CONCLUSION

- ✓ In GDP prediction, it is observed that the RMSE for the gdp3 model is the lowest. Therefore, it can be concluded that the gdp3 model is the best-fitted model and is expected to provide more accurate predictions.
- ✓ According to the predictions, the Indian economy shows an upward trend.
- ✓ Based on the predictions, the Human Development Index (HDI) of India is increasing.
- ✓ According to the predictions, the population of India is increasing rapidly.

BIBLIOGRAPHY

Website:

- ✓ *https://censusofindia2021.com/literacy-rate-of-india-2021/*
- ✓ *https://www.statista.com/statistics/271330/unemployment-rate-in-india/*
- ✓ *https://www.macrotrends.net/countries/IND/india/unemployment-rate*
- ✓ *https://www.theglobaleconomy.com/India/Literacy_rate/*

Books:

- ✓ Introduction to Time Series and Forecasting – Peter J. Brockwell, Richard A. Davis
- ✓ Introductory Time Series with R- Paul S.P. Cowpertwait and Andrew V. Metcalfe
- ✓ Forecasting: Principles and Practice- Rob J. Hyndman and George Athanasopoulos
- ✓ Practical Time Series Forecasting with R: A Hands-On Guide - Galit Shmueli and Kenneth C. Lichtendahl Jr