**Brief Report on World Bank Development Indicators:**

**Table of Contents**

[Dataset Description 3](#_Toc481182687)

[Problem Statement 4](#_Toc481182688)

[Flow Chart 4](#_Toc481182689)

[Step 1: Data Download and Wrangling 5](#_Toc481182690)

[Step 2: Exploratory Data Analysis 6](#_Toc481182691)

[Step 3: Clustering 10](#_Toc481182692)

[Manual Clustering: 10](#_Toc481182693)

[Clustering Algorithm 11](#_Toc481182694)

[Step 4: Decomposing Time Series 12](#_Toc481182695)

[Step 5: Making Time Series Stationary 13](#_Toc481182696)

[Step 6: Building Models 17](#_Toc481182697)

[Step 7: Choose the best model 22](#_Toc481182698)

[Summary of Metrics 25](#_Toc481182699)

[Step 8: Deploy the best model in Azure 25](#_Toc481182700)

[Step 10: Web application 28](#_Toc481182701)

[Link 28](#_Toc481182702)

[Step 12: Run Docker Image 31](#_Toc481182704)

[Github Repo 33](#_Toc481182706)

[Individual Contribution 33](#_Toc481182707)

[Pie Chart 34](#_Toc481182708)

**World Banks World Economic Development Indicators Dataset**

# Dataset Description

Dataset Link:

<http://data.worldbank.org/data-catalog/world-development-indicators>

"World development indicators is a primary world bank collection of development indicators, compiled from officially recognized international resources."

The World Bank contains Catalog of international economic, financial, and socio-economic data sets from the World Bank. Among the most useful of the many datasets offered is [World Development Indicators](http://data.worldbank.org/data-catalog/world-development-indicators) featuring time series data from 1960 for 207 countries in the areas of population, labor, education, economics, the environment and much more. For ex. The countries include.

* Argentina
* Brazil
* Ecuador
* India
* South Africa

The original sources of individual economic indicator dataset have also been listed in case the data presented here is incomplete, the original sources can be looked up for correct data. Below is the list of some of these data sources:

* Bureau of Economic Analysis, Haver Analytics
* Bureau of Labor Statistics, Haver Analytics
* UC Berkley Data Lab for information on economic conditions around the world
* Wikipedia to understand Millennium Development Goals

Reference Links for all the information related to Economic analysis:

1. [http://guides.lib.berkeley.edu/c.php?g=4395&p=15528#s-lg-box-wrapper-36413](http://guides.lib.berkeley.edu/c.php?g=4395&p=15528)
2. <https://en.wikipedia.org/wiki/Millennium_Development_Goals>

The data presented in this dataset is in the form of Time Series. The values for each economic indicator for each country are available from 1960 to 2015. Due to availability of large number of indicators, we can identify the relationships between different indicators, how they impact values of other indicators and also the dependency of future values on the previous values of same indicator.

Based on these definitions, there are two types of Time Series Data Analysis possible:

* Univariate: The future values of variable depends on its previous values
* Multivariate: The values of dependent variable depend on its previous values as well as values of other variables in the dataset

# Problem Statement

* United nations committee is looking for an application which will give them forecasted value for WDI for a selected country for given year
* To measure the progress of Global development, members of United Nations proposed a list of 8 Millennium Development goals
* For example:
* To achieve universal primary education
* To eradicate extreme poverty and hunger
* To achieve these goals and to improve country’s overall GDP and development we need to understand the complex relation between these Millennium goals and Economic indicators
* By seeing the predicted values of these economic indicators, we can craft a plan to overcome difficult economic problems
* Our model will help user to predict value of world economic indicators for a specific country for a specific year.
* We have considered following indicators for our analysis:
* Agricultural Land (% of land Area)
* Age Dependency Ratio (% of working age population)
* Birth Rate Crude (per 1000 people)
* Export of goods and services (% of GDP)
* GDP (Current USD)
* GDP growth (Annual %)
* Population growth (Annual %)
* Trade (% of GDP)

The forecast generated using this model can help us in knowing following details:

* Economic development of a particular country: based on GDP trends
* Export of country: based on Exports of goods and services
* Dependent population of the country: based on Age dependency ratio
* Population impacting the economy: population growth

We are restricting our analysis here to Univariate Time Series Analysis but at the same time building a base for Multivariate Time Series Analysis. Following points summarize our approach in handling this project:

* Step 1: Data Download and Data Wrangling
* Step 2: Exploratory Data Analysis: Summarizations and Analysis
* Step 3: Clustering the data
* Step 4: Decomposing Time series’
* Step 5: Making Time Series Stationary
* Step 6: Building AR, MA, ARMA and ARIMA untuned models
* Step 7: Choosing the best tuned model
* Step 8: Dockerize the projects
* Step 9: Upload the Docker image and Web app to cloud (S3)
* Step 10: Deploy the best model in Azure and create REST API
* Step 11: Consume the REST API in the web app

The web application developed in this process will allow the user to select a state, year and economic indicator of his/her choice and get the requested values. The above mentioned steps are represented using a Flow chart and explained in further detail in following sections.

# 

# Flow Chart

Start

Download data using Python script

**DOCKER**

Data wrangling and transformations

Exploratory Data Analysis in Python and Tableau

Clustering

Decompose the data & build AR, MA, ARMA & ARIMA models

Check AIC for all models

Good AIC?

No

Tune the model

Yes

Deploy in Azure and build REST API

Build Web App and consume the REST API

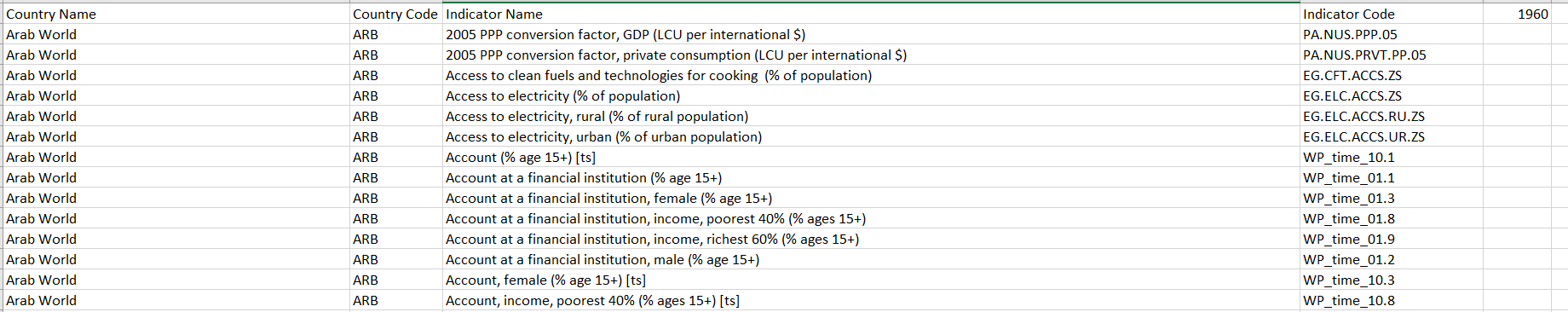
Stop

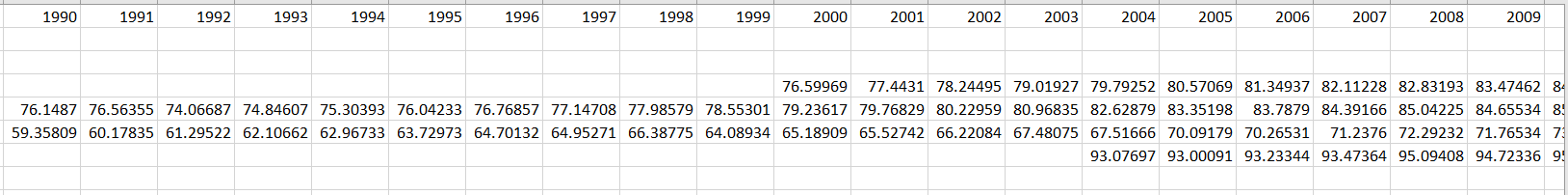
## Step 1: Data Download and Wrangling

* The first step here is to programmatically download the data from FED Data website:

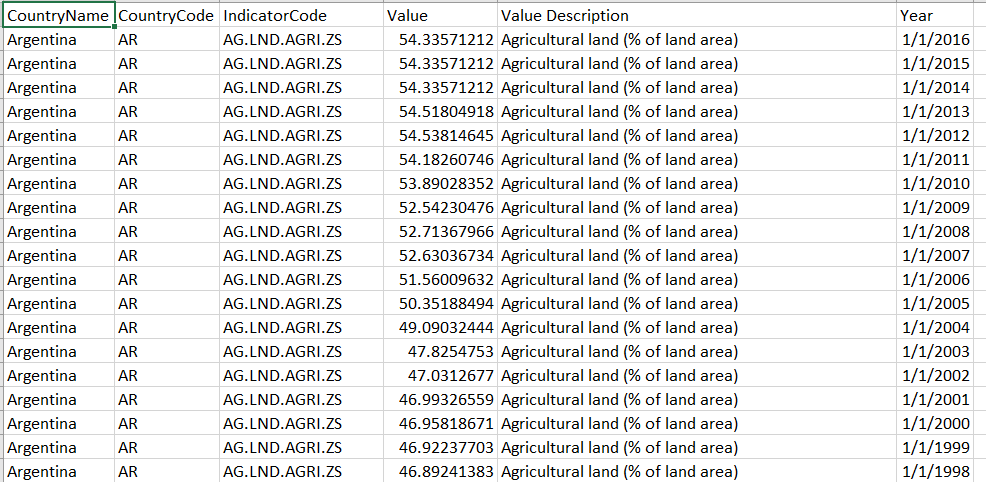
<https://www.richmondfed.org/research/regional_economy/reports/regional_profiles#tab-2>

* The data for each indicator is present in separate csv files. Also data for individual states is present in separate tabs of the csv file. So first we have to clean the data, transform it, and separate the data in tabs into different files.
* Below is the sample screenshot of original csv files

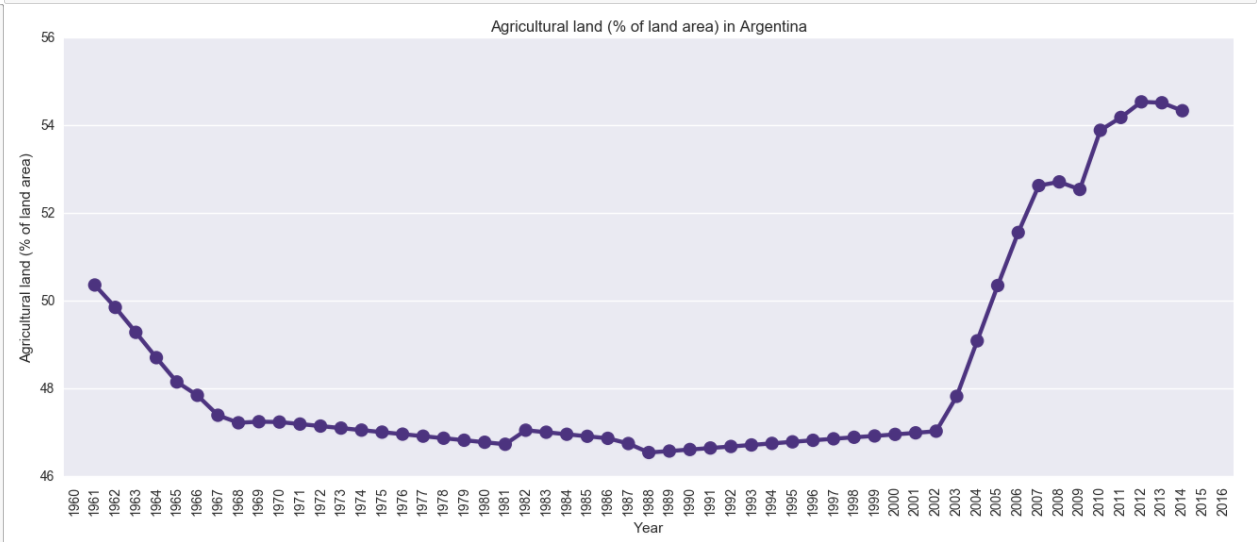




* Following is the screenshot of transformed files created :



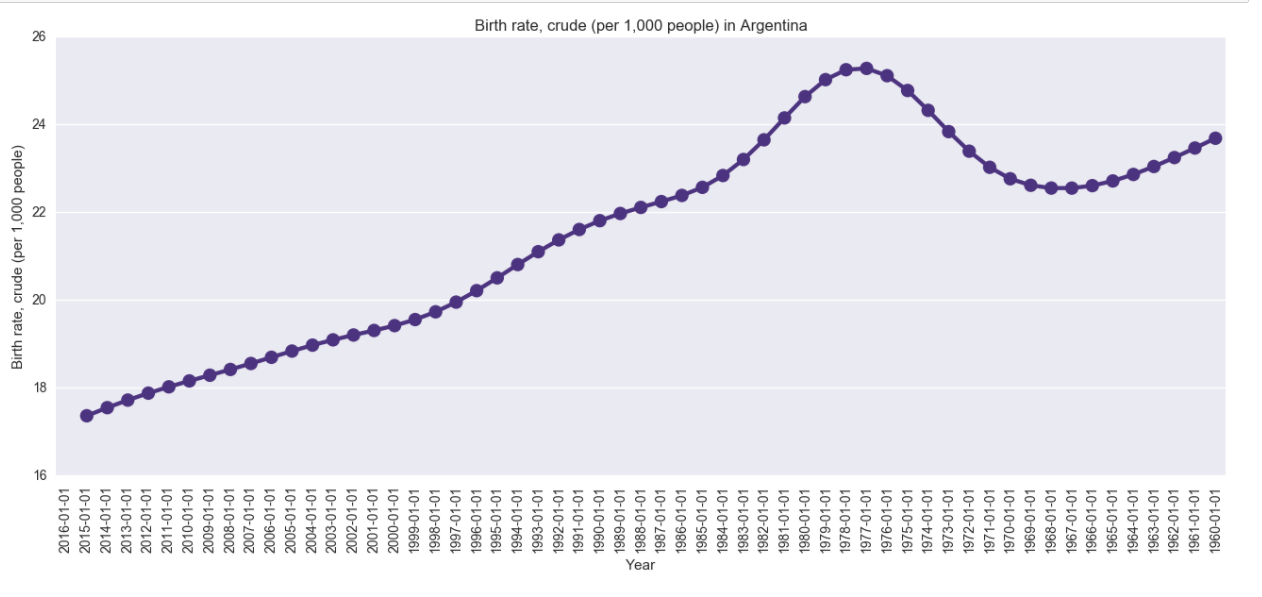
* The data for each country and respective counties and metro areas was present in row format, hence we had to clean it and transform it in column format to further save it as a dataframe.
* After creation of dataset, check for missing data and impute correct values
* **Missing Data Handling:**
* In this case, we had very little missing data. Only some value for indicators like Agricultural Land and Birth Rate was missing. Here we have imputed the value using bfill() method.
* We have also used Interpolate() to fill out missing values because most of the places only one of the year value was missing so it would be best if we fill it with values in between.
* Most of the places the values missing was the last year in dataset i.e. 2016 so we used bfill() to fill that value.
* Here is the EDA part for missing value and outlier analysis:
* For each country and for each indicator we have platted the graph to see the missing values as well as any outliers in the dataset.
* Agricultural Land Argentina:



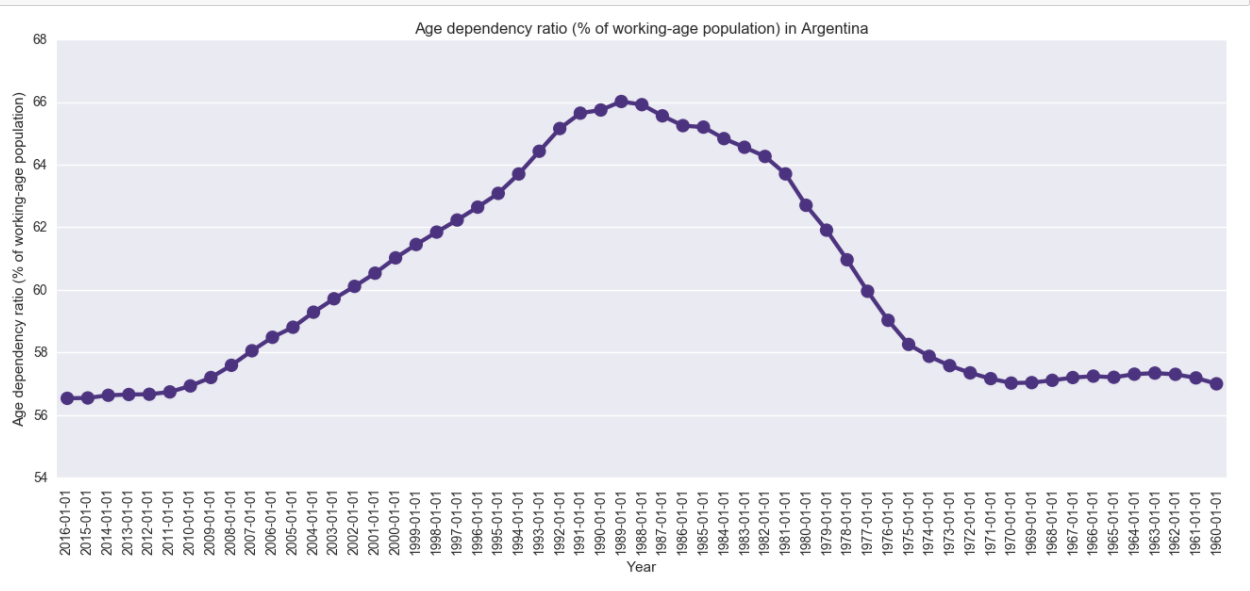
* Sample code for calculating for all the indicators is:



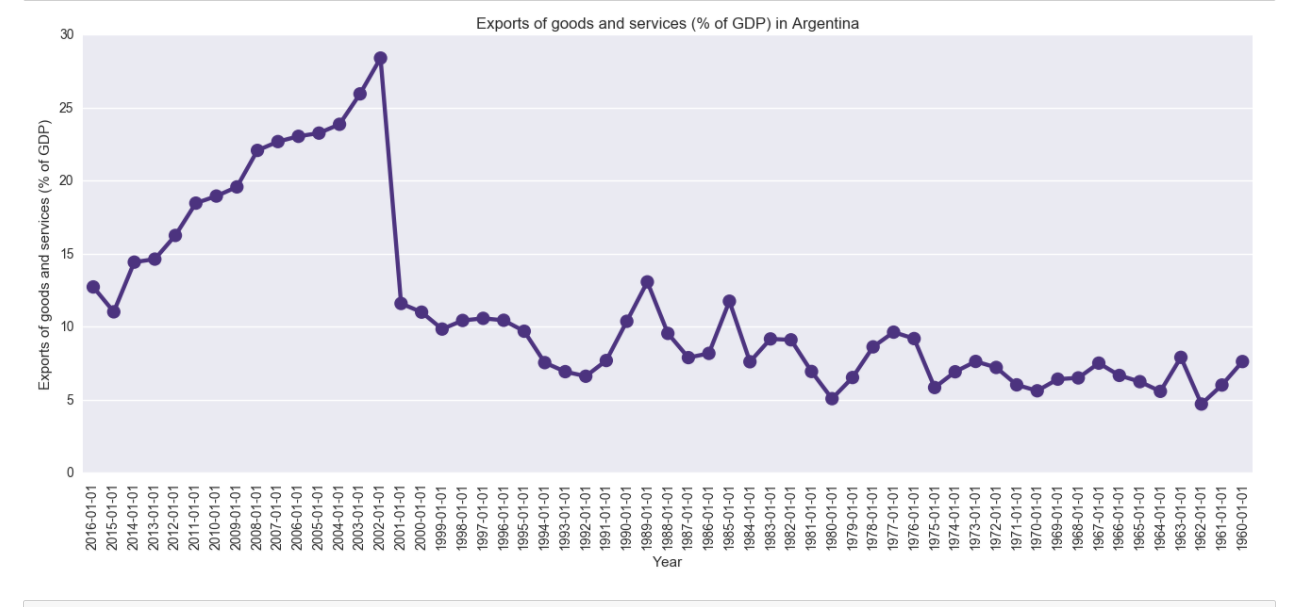
* Birth Rate Argentina:



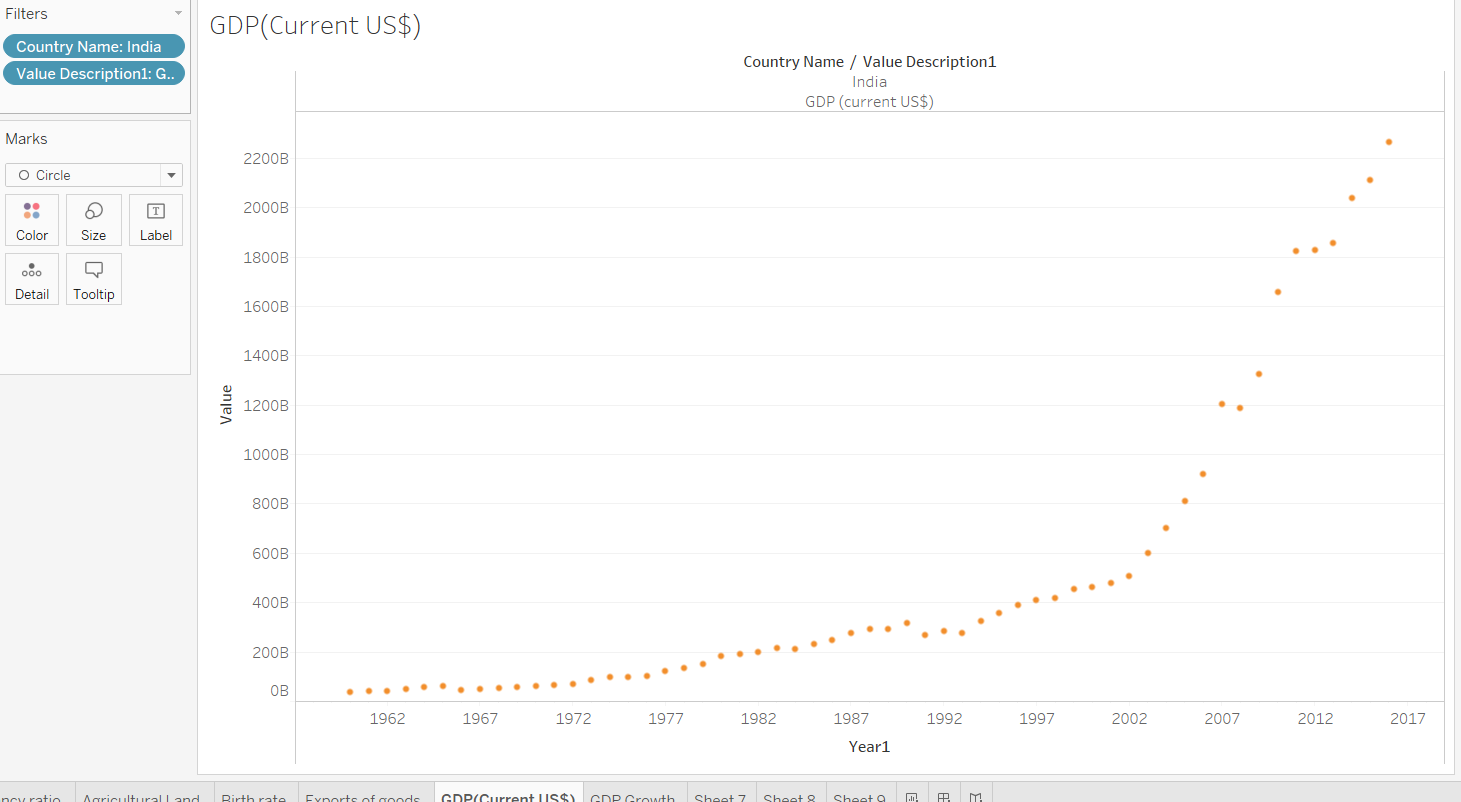
* Age dependency Ratio:

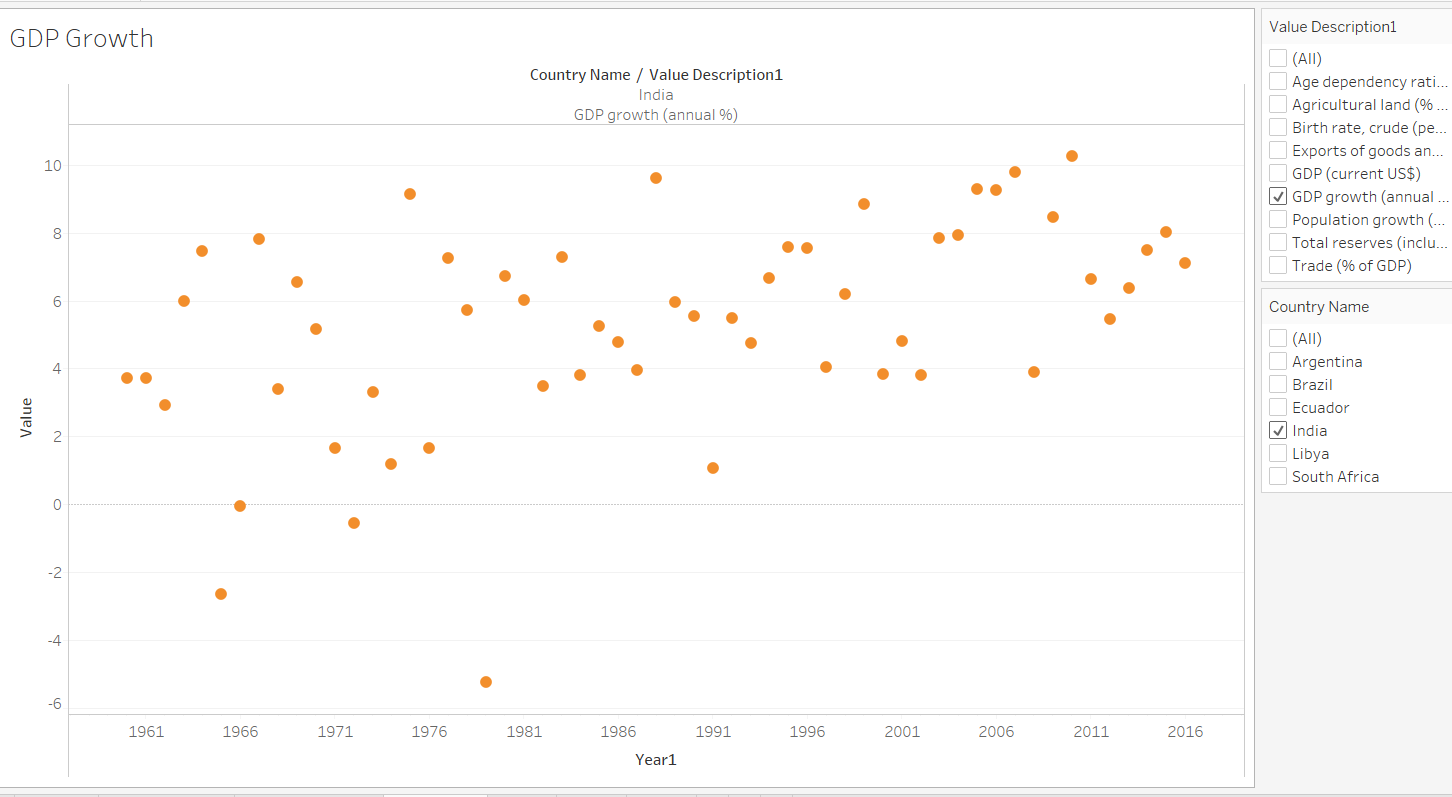


* Exports of goods and services:



* We have also done analysis in tableau using scatter plots to see the outliers.



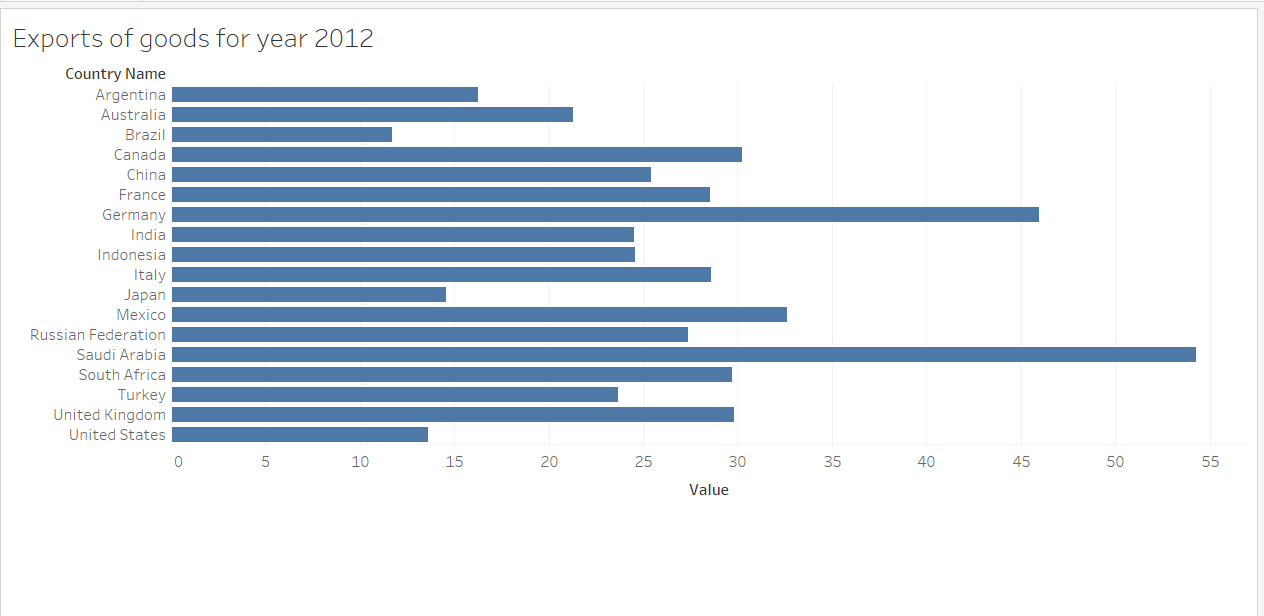


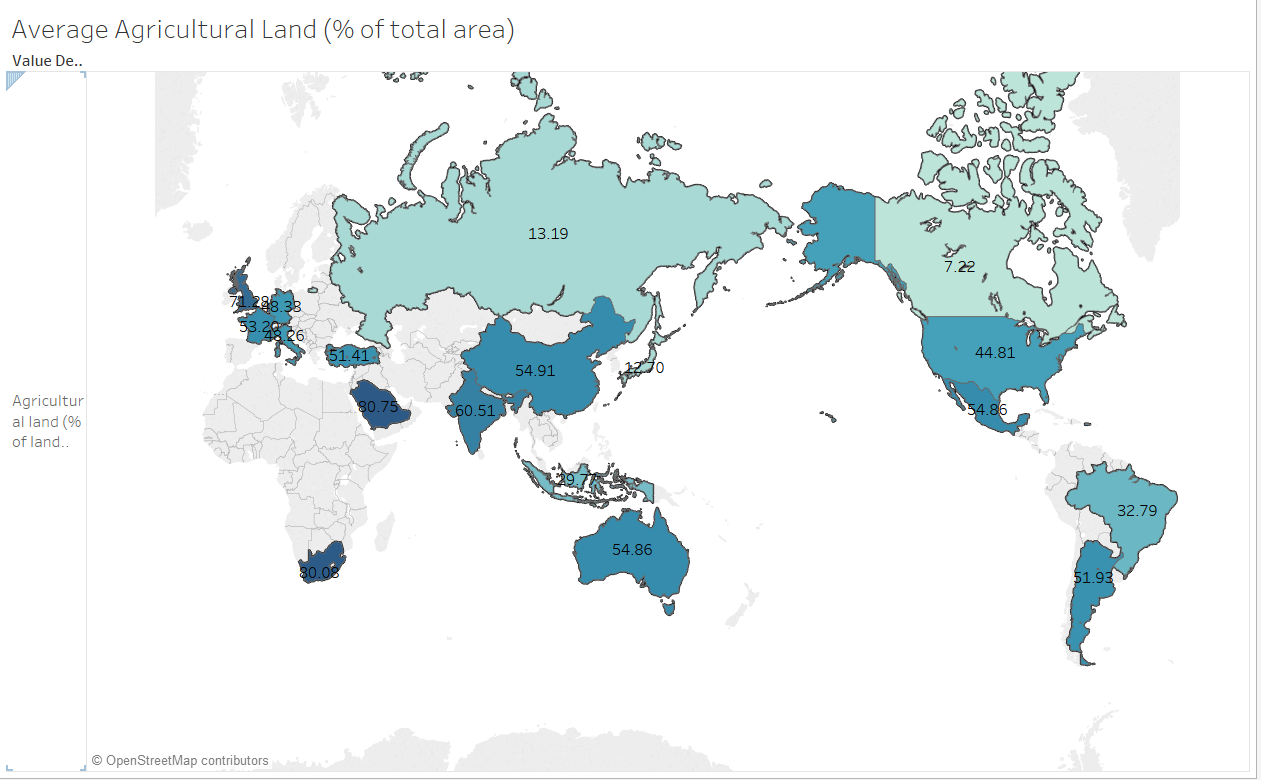
## Step 2: Exploratory Data Analysis

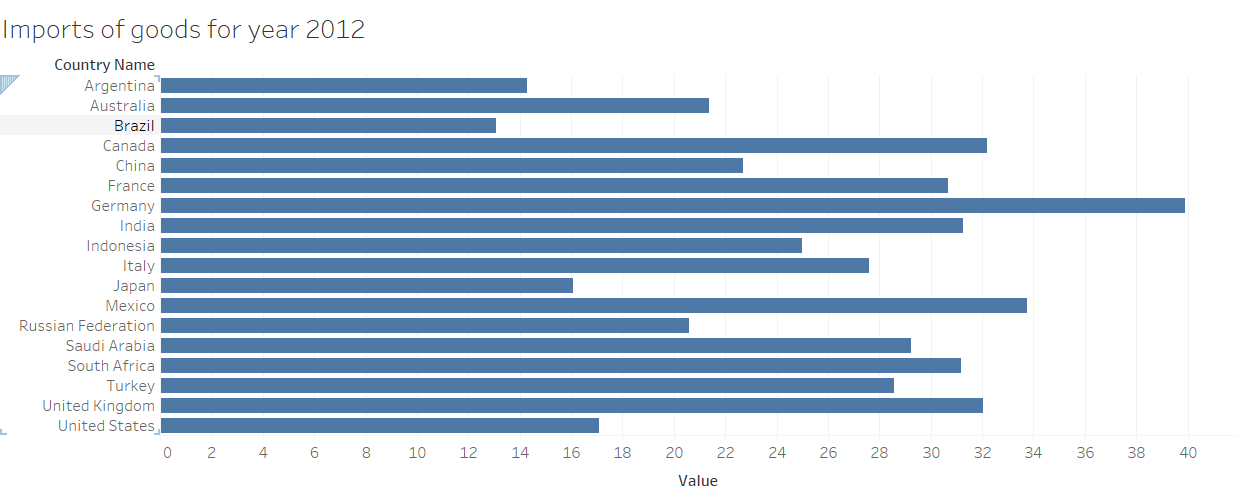
Here we have summarized some important variables and tried to figure out the correlations between them. This helped us to gain some key insights on the trends and impacts of variables on other variables. Based on the conclusions it was further decided which variables to consider for creating machine learning models.

1. Compare different countries based on Exports of goods, Agricultural Land, Birth Rate and etc.: for different years.

* We have applied a filter for every year from 1960 here we are showing values and graphs for 2012 only for demo purpose:

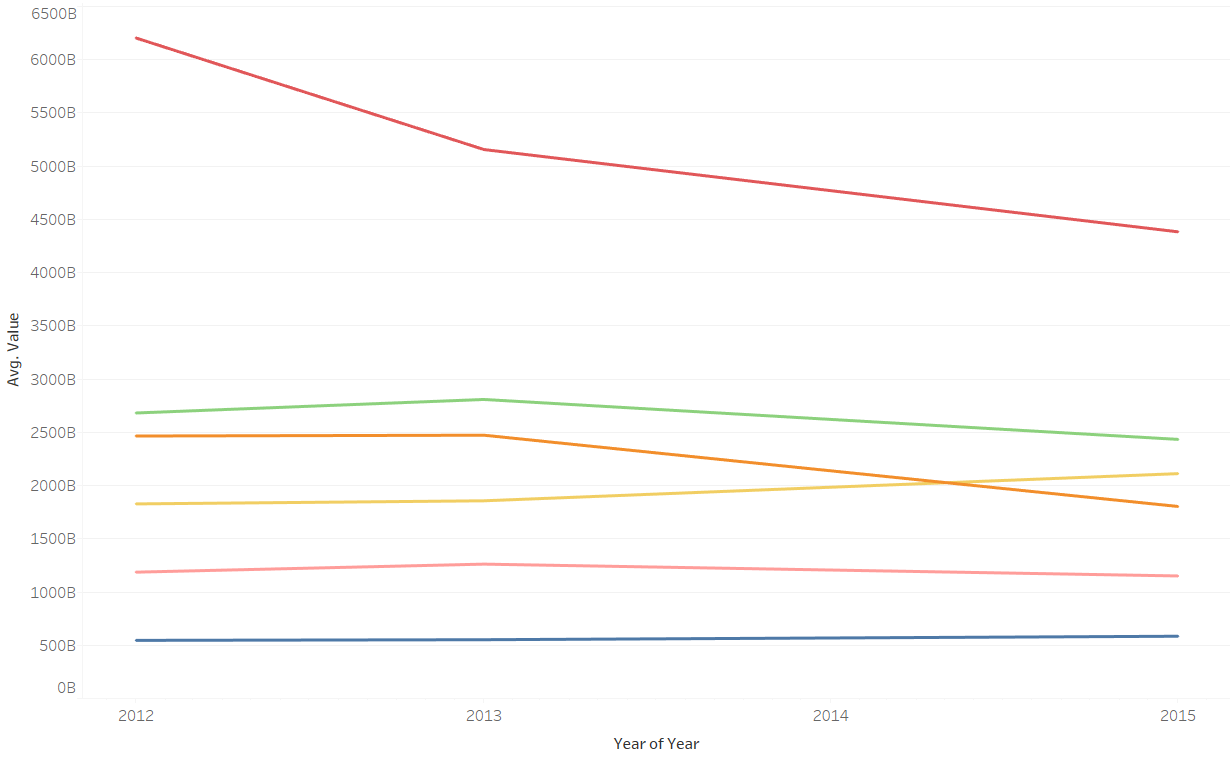






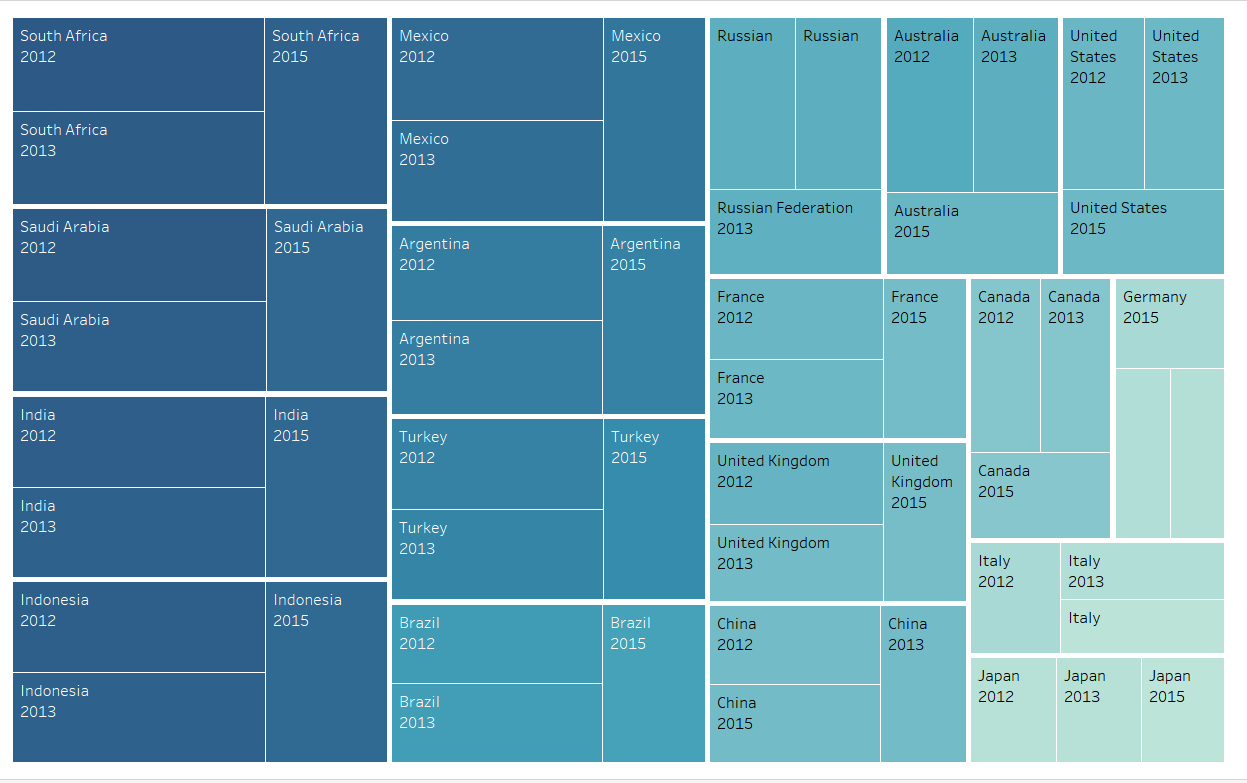
Here we can see that G20 countries imports more goods than they export

* GDP trends over the years in



Countries like India and Argentina are growing at a decent rate

* Birth Rate Distribution

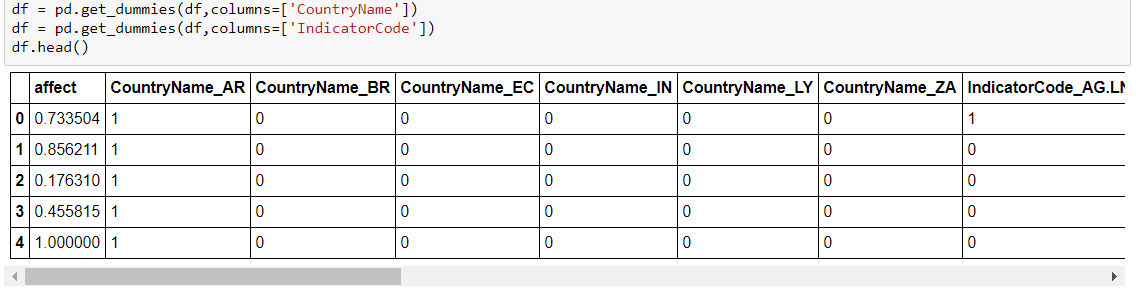


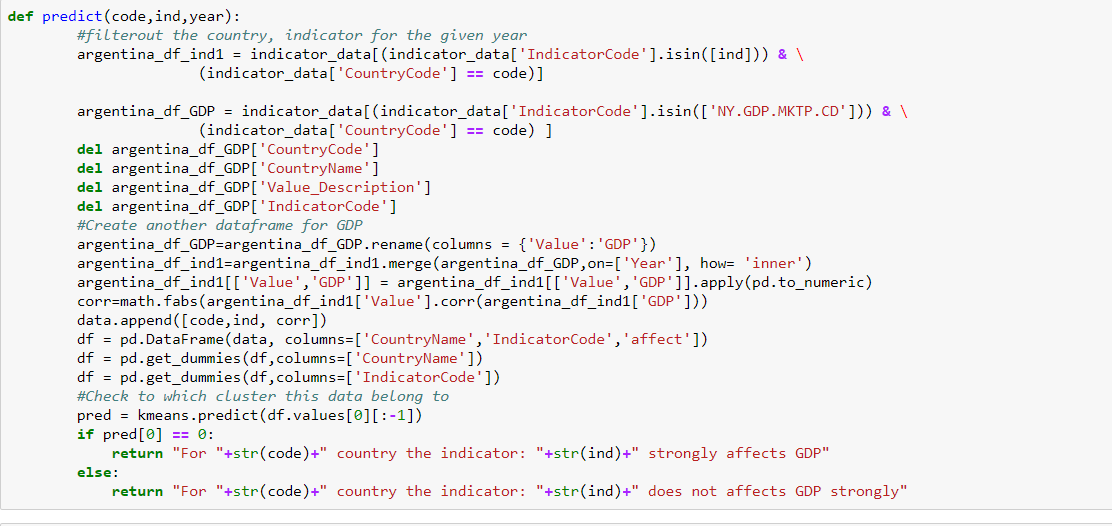
We can see that countries with high population are because they have a higher birth index rate

## Step 3: Clustering

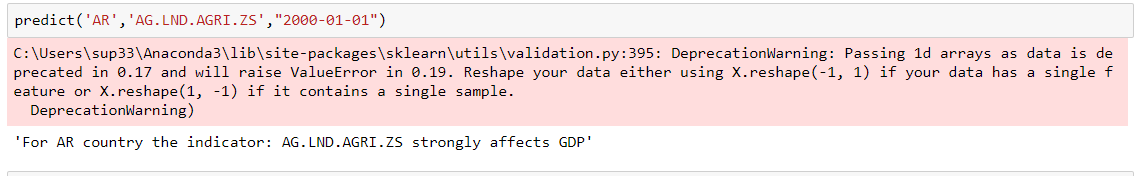
### Clustering Algorithm

* The one use case to develop further in future is clustering where user can select the country and indicator and there he will see if the selected indicator is strongly or weakly affecting the GDP of the country.
* Since this is a time series data we have option of doing clustering with different method.
* We have applied K-means clustering before applying k-menas we determined correlation between indicators and then we applied clustering
* The clustering algorithm is giving a 2 cluster one is strongly associated and other is weakly associated with the gdp of the country.
* Here is a code sample for the clustering.
* Also, we have transformed data in to the required format for the clustering
* Sample:





**Result**:



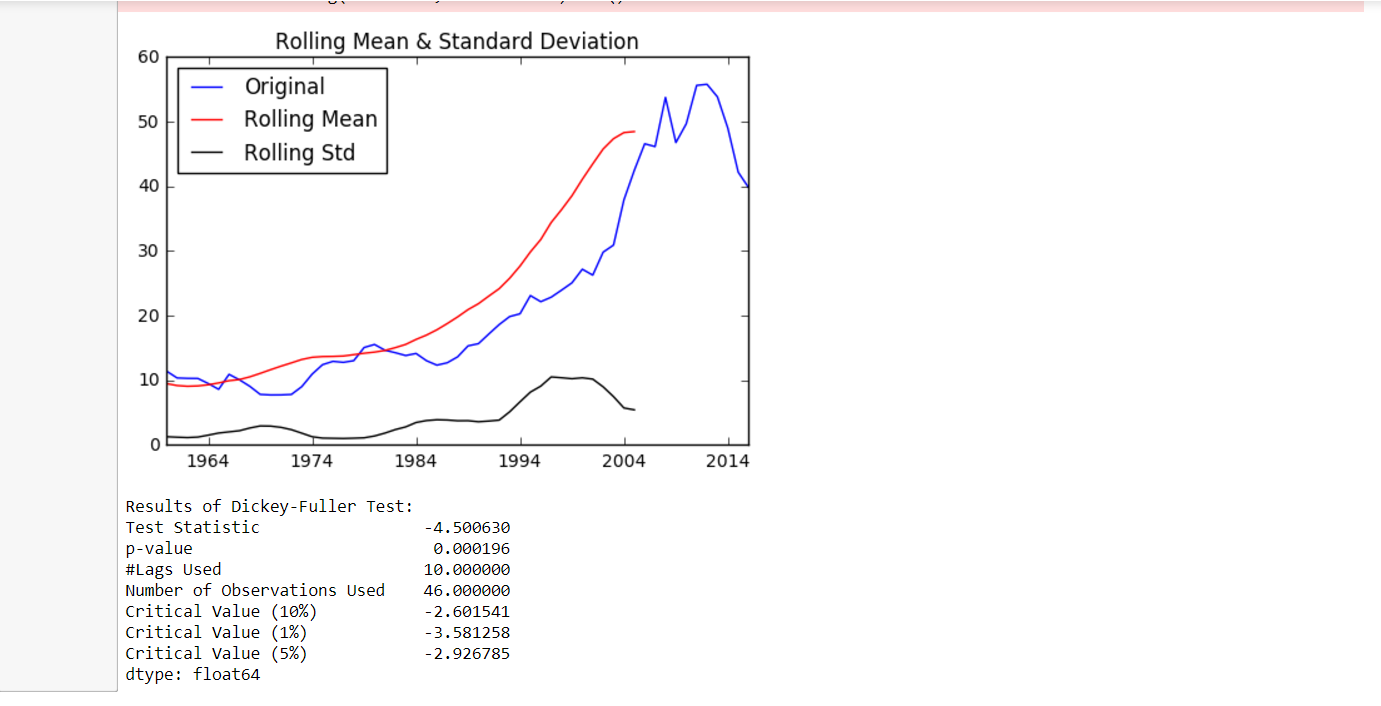
## Step 4: Decomposing Time Series

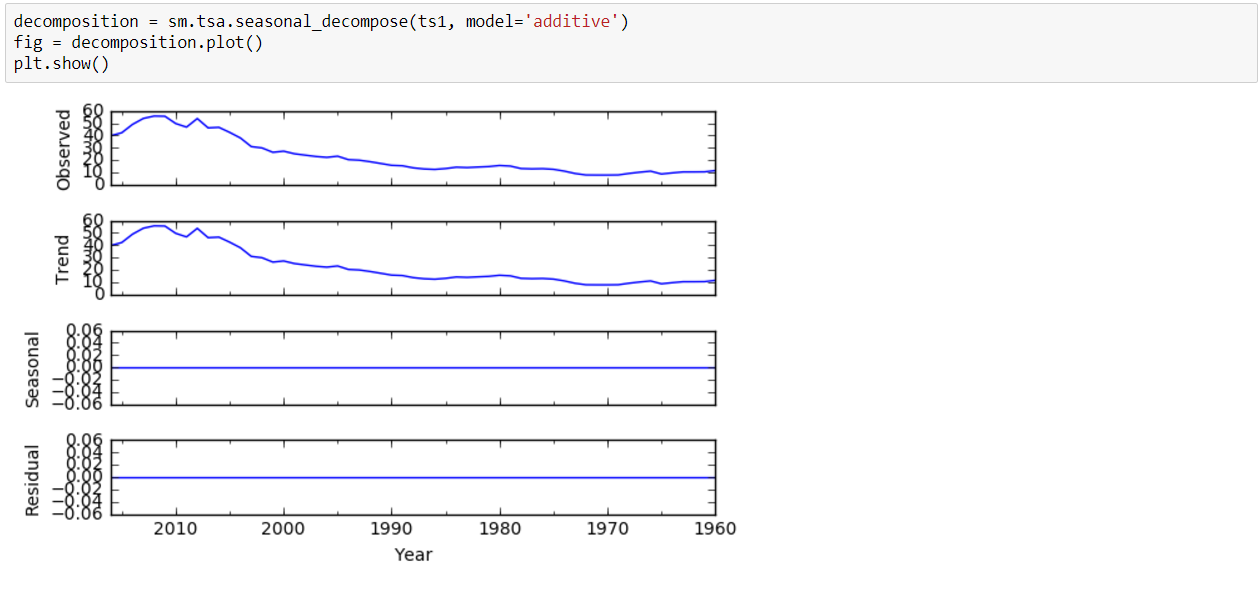
The decomposition of time series is a statistical method that deconstructs a time series into several components, each representing one of the underlying categories of patterns. Here we have decomposed the time series’ based on rates of change. It seeks to construct, from an observed time series, a number of component series (that could be used to reconstruct the original by additions or multiplications) where each of these has a certain characteristic or type of behavior. This is an important technique for seasonal data adjustment. Following are the components in which time series is decomposed:

* **Trend**: at time t reflects the long-term progression of the series. A trend exists when there is an increasing or decreasing direction in the data
* **Cyclical**: at time t describes repeated but non-periodic fluctuations. The duration of these fluctuations is usually of at least two years.
* **Seasonal**: at time *t*, reflects seasonality (seasonal variation). A seasonal pattern exists when a time series is influenced by seasonal factors. Seasonality is always of a fixed and known period (e.g., the quarter of the year, the month, or day of the week)

**Noise**: at time t, that describes random, irregular influences. It represents the residuals or remainder of the time series after the other components have been removed

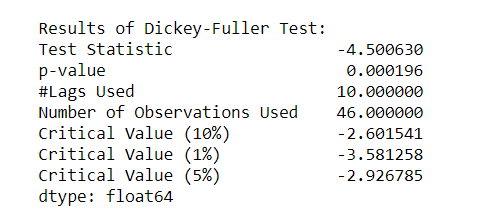






* Conclusion from above time series is this time series is not stationary.
* We have tested the stationarity with **Dickey-Fuller Test.**

Here is the result of the test:



* As seen from the test critical value is more than test statistics .

## Step 5: Making Time Series Stationary

Once we decompose the available time series data, we can see that the values of indicators vary a period of time, which is expected. But in order to do further detailed analysis we need to make the time series stationary. A TS is said to be stationary if its statistical properties such as mean, variance remain constant over time. Stationary time series is defined using very strict criterion. However, for practical purposes we can assume the series to be stationary if it has constant statistical properties over time:

* constant mean
* constant variance
* an auto covariance that does not depend on time

As per the decomposition done above it is clear that all our time series have a clear trend. We go ahead and check the stationarity using following two methods:

* Plotting rolling statistics:

Here we can plot the moving average or moving variance and see if it varies with time.

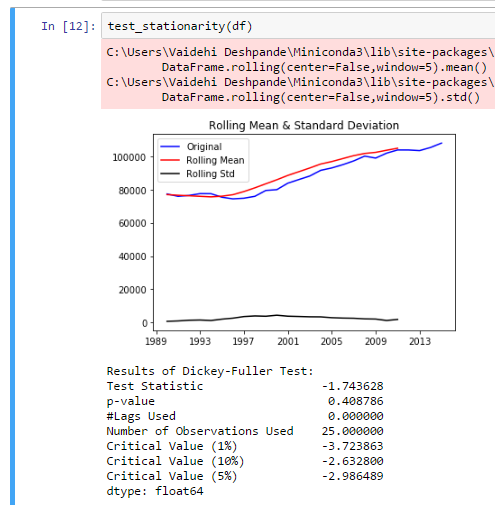
* Dickey-Fuller Test:

This is one of the statistical tests for checking stationarity. Here the null hypothesis is that the TS is non-stationary. The test results comprise of a Test Statistic and some Critical Values for difference confidence levels. If the ‘Test Statistic’ is less than the ‘Critical Value’, we can reject the null hypothesis and say that the series is stationary.

Below is the screenshot of the function that we have created to determine the stationarity of the time series.

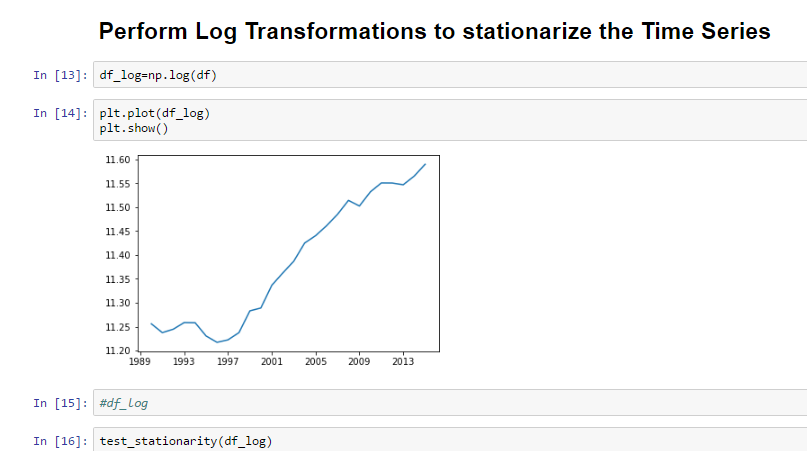


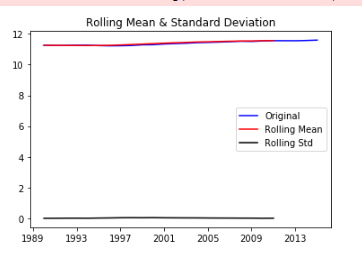
* We first check the stationarity of our original time series data frame



* As seen above, this time series is not stationary because of following reasons:
* Mean and variance are not constant over a period of time
* The “Test Statistic” value is more than all three Critical Values

In order to make the time series stationary we take logarithmic transformations and then again perform stationarity check. Below screenshots explain the same:

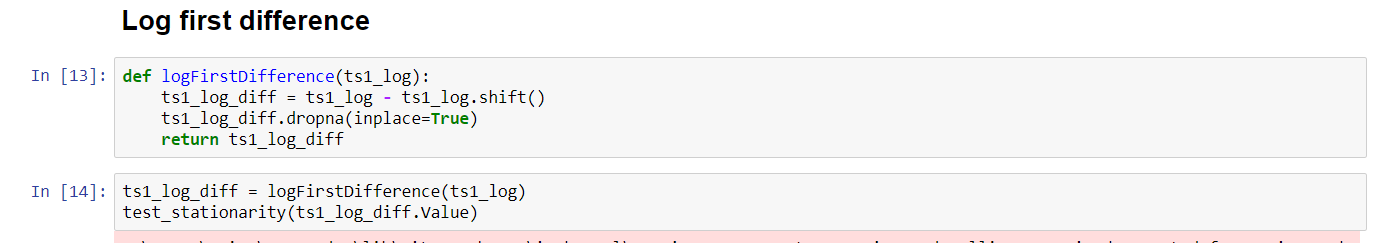


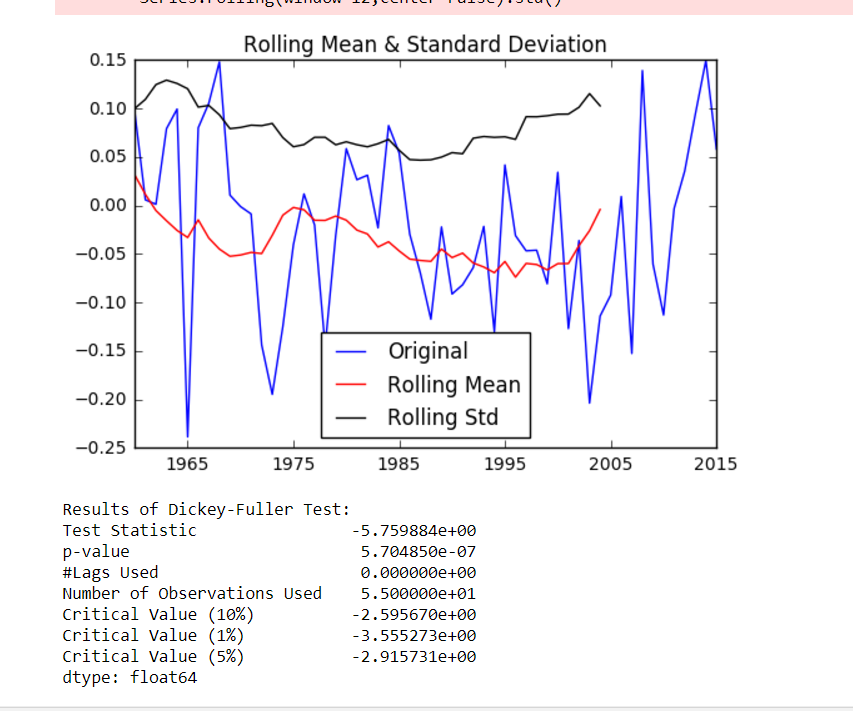


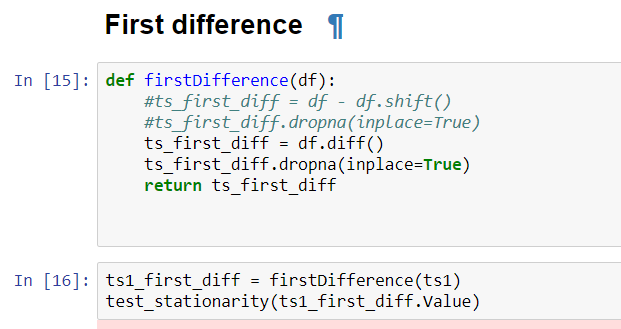
As seen in above graphs after taking log transform the mean and standard deviation are almost constant.

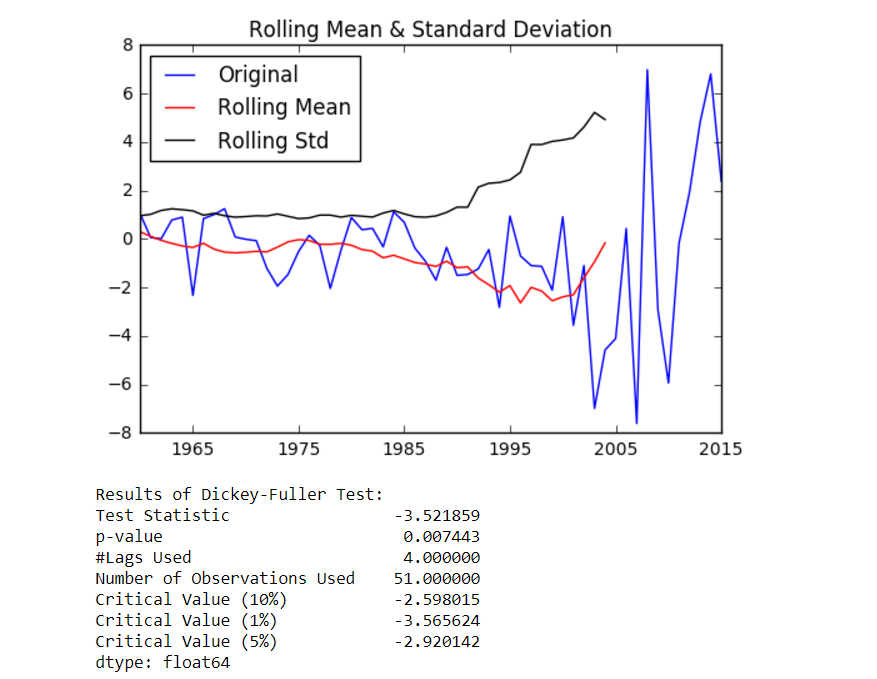
Hence, we can now consider that this time series is now stationary and go ahead with building regression models.

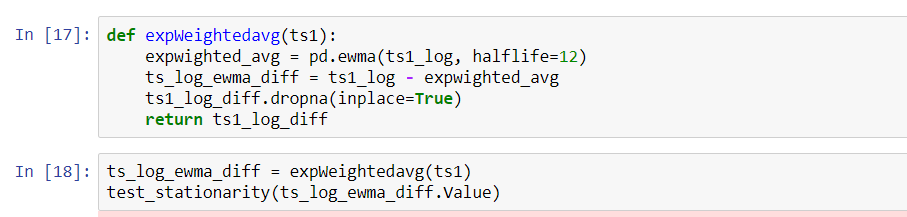
* If after the log transform time series does not show stationary nature then you have to take log first difference .
* Also we can take first difference of original series .
* Below are the functions we have developed to check for that values and see the result

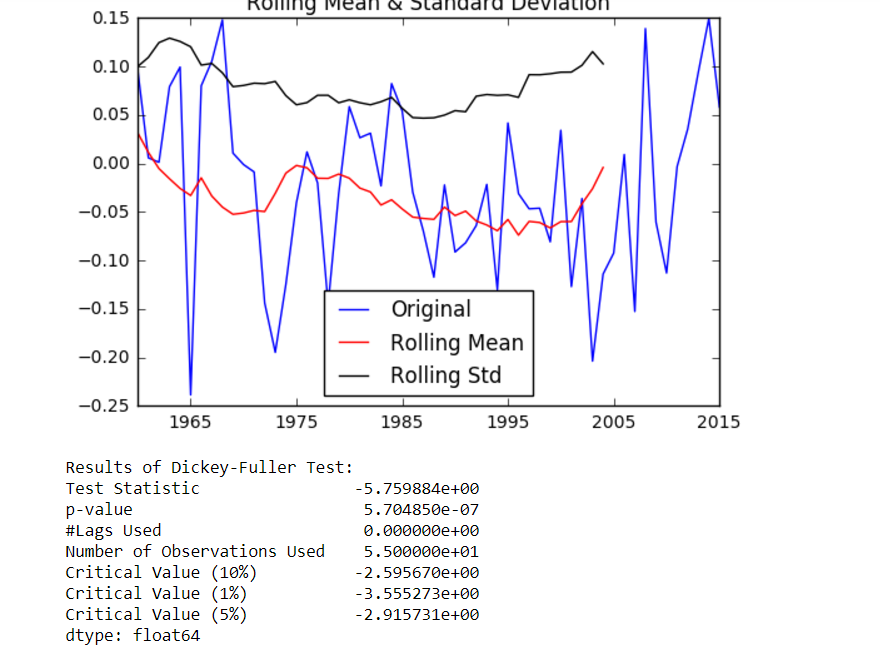












## Step 6: Building Models

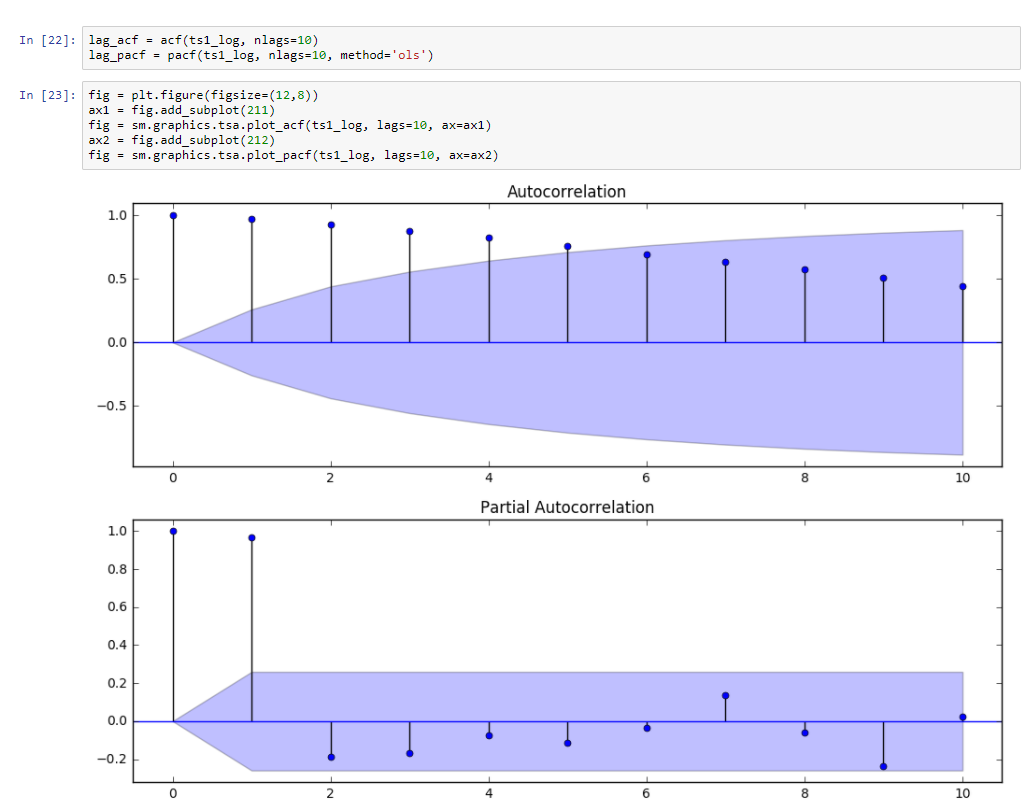
In Time Series Data Analysis, we use special type of regression models known as Auto Regression (AR) models. An autoregressive model is when a value from a time series is regressed on previous values from that same time series. In addition to AR, we also have Moving Average (MA) models and a combination of both ARMA models. The predictors depend on the parameters (p,d,q) of the ARIMA model:

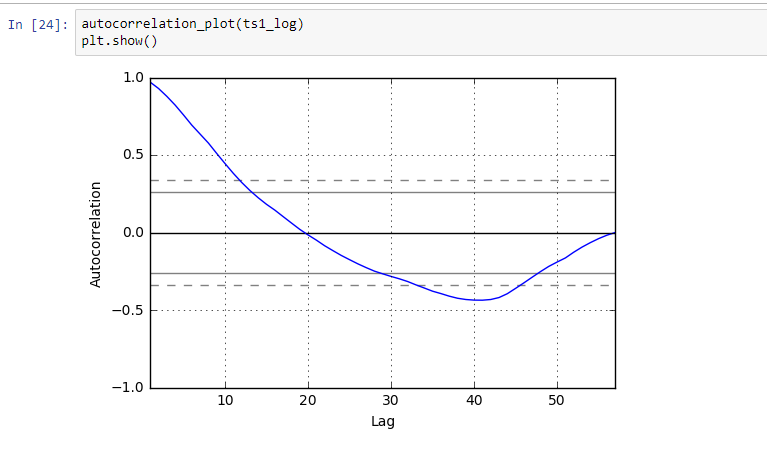
* Number of AR (Auto-Regressive) terms (p): AR terms are just lags of dependent variable. For instance if p is 5, the predictors for x(t) will be x(t-1)….x(t-5).
* Number of MA (Moving Average) terms (q): MA terms are lagged forecast errors in prediction equation. For instance if q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average at ith instant and actual value.
* Number of Differences (d): These are the number of nonseasonal differences, i.e. in this case we took the first order difference. So either we can pass that variable and put d=0 or pass the original variable and put d=1. Both will generate same results.

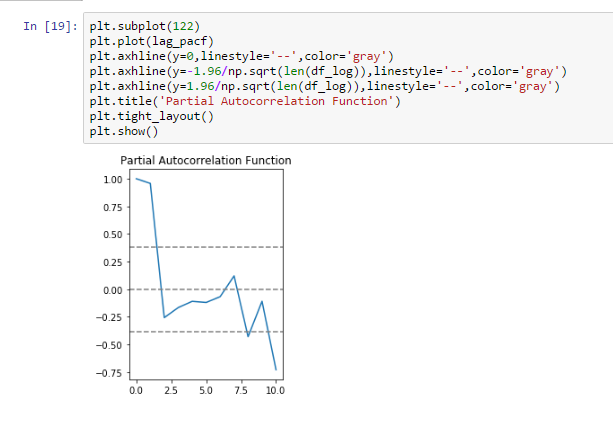
Since, we have used Python for our project we imported the statsmodels package which contains all these models.

Below are the steps taken to build tuned models:

**Plotting ACF PACF graphs:**



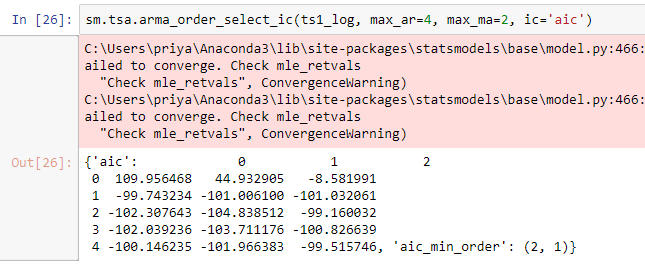




* Build Models:

Once the values of p and q are determined, we go ahead and build the AR, MA and ARMA models and then determine the best model based on value of AIC. The lower the value of AIC the better is the model.

* Also to determine the order of the models we are using python function which will give us possible order of p,d,q. Here is below how we are doing it:



## Step 7: Choose the best model

We have calculated AIC (Akaike information criterion) to measure the strength of the model.

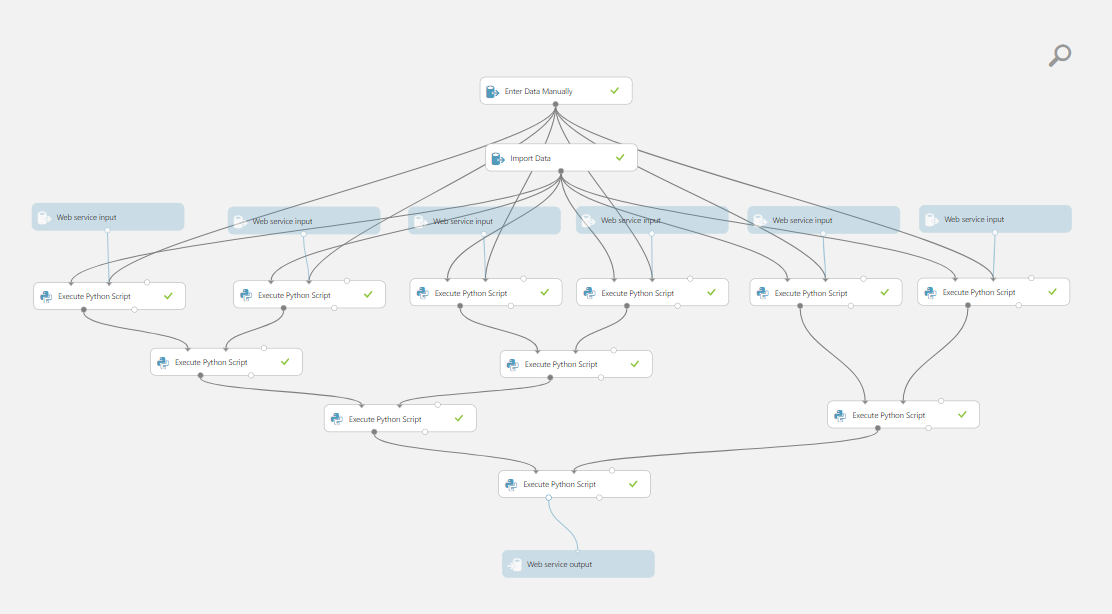
Definition of AIC:

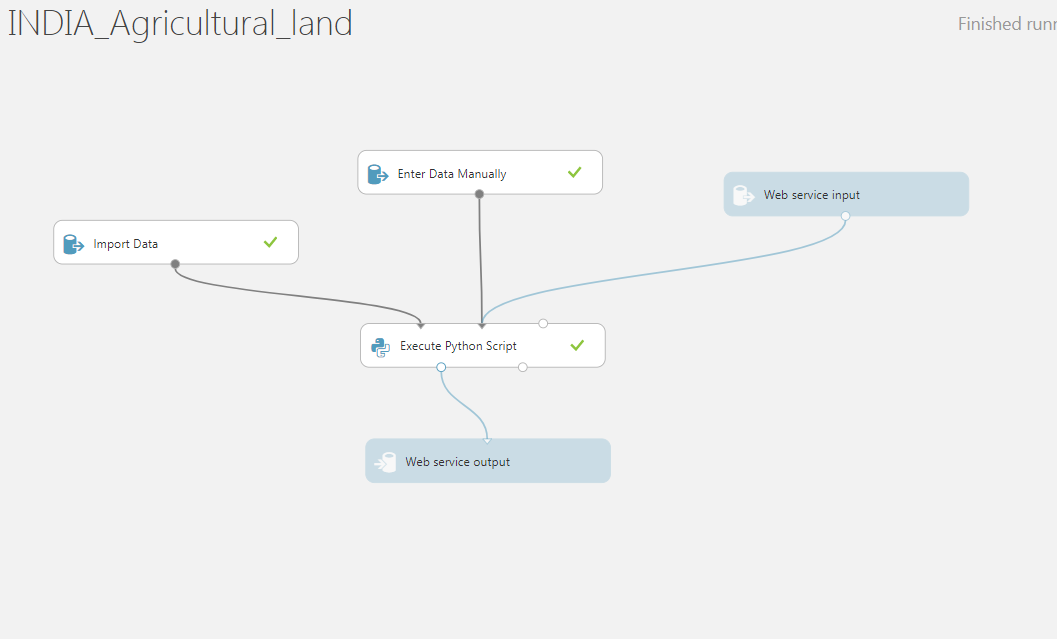
The Akaike information criterion (AIC) is a measure of the relative quality of statistical models for a given set of data. Given a collection of models for the data, AIC estimates the quality of each model, relative to each of the other models. Hence, AIC provides a means for model selection.

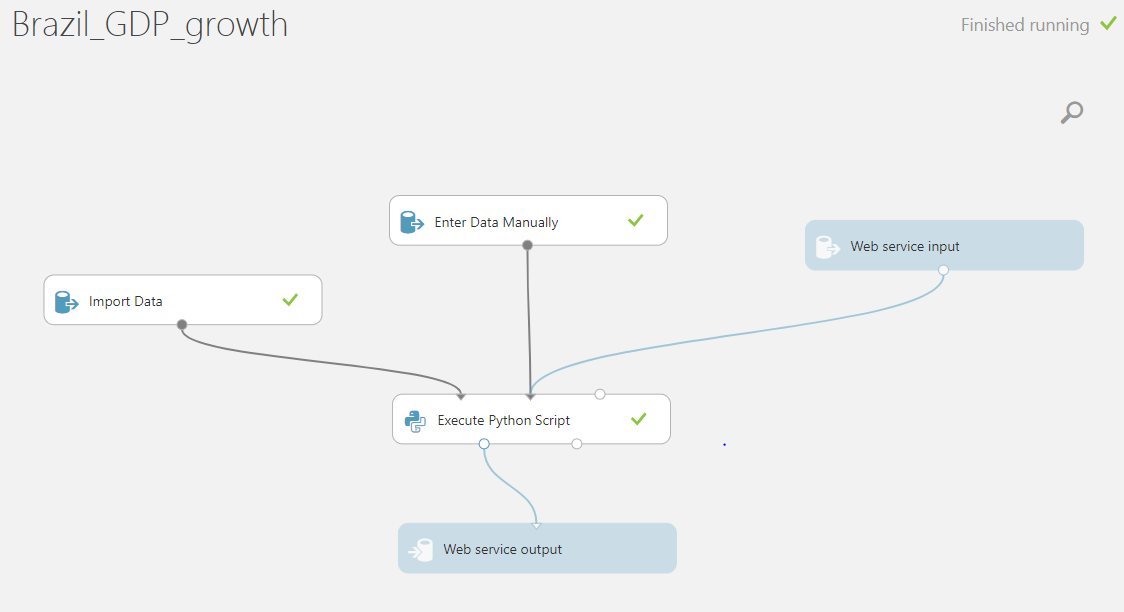
Hence, using following piece of code for all indicators and states we find out the best model based on AIC value:

## Step 8: Deploy the best model in Azure

After finding the best model using python code through the value of AIC, we have further deployed it in Azure Machine Learning Studio to generate the REST API. Since the AR/ARMA models were not available in the studio, we have used blocks of python code to implement them. Below are the models created for each indicator for each country:







+

## 

## 

## Step 10: Run Docker Image

Commands:

1. Docker hub link:

<https://hub.docker.com/r/prashantvksingh/ads_final/>

Tag: team6finalproject

1. Pull the docker image from docker hub:

**docker pull prashantvksingh/ads\_final:team6finalproject**

1. Run the Image: Please give your access key, secret key and bucket name to upload the files to Amazon S3.

Copy the command below and make necessary changes as instructed:

|  |
| --- |
| **docker run -e AWS\_ACCESS\_KEY="AKIAJL2ZPNORFYQ74FVQ" -e AWS\_SECRET\_KEY="LijYsR9NdNIJd/j2XBHPBAAKttphXOO6w1/a+Wcn" prashantvksingh/ads\_final:team6finalproject /usr/src/ADS\_Final/run.sh** |

Replace accesskey, secretkey and bucket with your credentials.

Here, use your credentials in the command, your access key, secret key and bucket name followed by our docker image name.

For environmental variable: accesskey, secretkey and bucket. These are case sensitive, just use as given in the command.

Note: There should be no space in your credentials, otherwise the code will give error message to enter correct credentials. Please do not remove double quotes(“accesskey=<enter your accesskey>”) also.

1. After successful execution files are uploaded to S3 bucket as shown in below screenshots:

