**POPULATION PROJECTION OF GOVERNMENT DATA**

**Submitted by: Under the Guidance of:**

Ajay Ravi Renu Sri Gandham

Logesh Naveen Kumar

Ravi Shankar

Kishore Rajan

Susil kumar

Leelavathi

***TABLE OF CONTENTS***

1.INTRODUCTION……………………………………………………………………………………………...2

2.LITERATURE SURVEY………………………………………………………………………………………2

3. PROBLEM DEFINITION……………………………………………………………………………………...2

4.DATA COLLECTION………………………………………………………………………………………………………………………..3

5.SYSTEM ANALYSIS…………………………………………………………………………………………………………………………3

6. SYSTEM ENVIRONMENT……………………………………………………………………………………………………...3

6.1 Hardware Requirements……………………………………………………………………………….3

6.2 Software Requirements………………………………………………………………………………..3

7.IMPLEMENTATION…………………………………………………………………………………………………………………….4-30

8.CONCLUSION………………………………………………………………………………………………………………………………31

9.REFERENCES…………………………………………………………………………………………………………………………………31

1. **INTRODUCTION:**

A Key competence of successful countries is the ability to plan flexibly for the short and long term. Planning must be based on good information about the present situation, on how important variables will change in the future and on how much faith to place in future predictions. One signiﬁcant variable that enters national and local planning is population. Much

effort is expended by researchers and governments in projecting the future of the population under a given set of assumptions.

To provide guidance about uncertainty, statistical ofﬁces develop variant projections based on alternative scenarios about the projection ‘drivers’. More recently researchers have attempted to be more precise about uncertainty and have made estimates of the probability of outcomes. at the national scale, international migration estimates are subject to error, while for regions, internal migration data series suffer from gaps in time coverage or discontinuities in geographical coverage.

1. **LITERATURE SURVEY**

The literature survey deals with the topics and the researches that would help to understand the population projection. The objective of this literature survey is to analyze the related work to this project and mechanisms used in previous studies.

### **[Title:](Title: Future Population Growth - Our World in Datahttps://ourworldindata.org › future-population-growth)** [Future Population Growth - Our World in Data](Title: Future Population Growth - Our World in Datahttps://ourworldindata.org › future-population-growth)

[https://ourworldindata.org › future-population-growth](Title: Future Population Growth - Our World in Datahttps://ourworldindata.org › future-population-growth)

**Author:** M Roser · 2013

[World population growth](https://ourworldindata.org/world-population-growth) – This article is focusing on the history of population growth up to the present. We show how the world population grew over the last several thousand years and we explain what has been driving this change.

[Age Structure](https://ourworldindata.org/age-structure) – What is the age profile of populations around the world? How did it change and what will the age structure of populations look like in the future?

1. **PROBLEM DEFINITION**

A population projection gives a picture of what the future size and structure of the population by sex and age might look like. It is based on knowledge of the past trends, and, for the future, on assumptions made for three components: fertility, mortality and migration. Individual behaviour, certain public policy actions, scientific progress or unforeseen events (weather events, epidemics) in the coming years may have a lasting effect and significantly influence trends, which the projections do not take into account. Unfortunately, not understanding population projections can lead to [much larger problems](https://web.archive.org/web/20180922090906/https:/transportation.house.gov/uploadedfiles/documents/2013-04-17%20-%20costs%20of%20overbuilding%20and%20unneeded%20courthouses.pdf) than a rained out barbecue. Every population projection has a margin of error, just like weather forecasts do. The most commonly accepted margin of error for 10-year population projections at county level is 12 percent.

1. **DATA COLLECTION**

Taking the dataset and build a machine learning time series model which would help forecast/project the population estimate for subsequent year i.e 2020 based on the current and historical statistics information.

#### 5. SYSTEM ANALYSIS

#### System Analysis is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components. System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose. The objective of the system analysis activity is to develop structured system specification for the proposed system. The structured system specification should describe what the proposed system would do; independent of the technology.

1. **SYSTEM ENVIRONMENT**

**6.1 Hardware requirements:**

* Disk Space: 16 GB or more
* Processor: 1.4 GHz or more 32 bit/64 bit
* Memory: 4 GB or more

**6.2 Software requirements:**

**Python anaconda**

Download and install the software from <https://www.anaconda.com/products/individual>

**Power BI:** Download and install the software from url

<https://app.powerbi.com/SignupRedirect?ru=https%3A%2F%2Fapp.powerbi.com%2Fhome>

1. **IMPLEMENTATION**
2. **Importing libraries:**

Importing all the necessary libraries we need such as:

* **pandas:** pandas has been one of the most commonly used tools for Data Science and Machine learning, which is used for data cleaning and analysis.
* **glob:** glob is a standard python library it is used to return all file paths that match as specific pattern.
* **numpy:** numPy is a Python library used for working with arrays.
* **matplotlib:** matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
* **statsmodels:** statsmodels is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration.
* **Pylab:** PyLab is a procedural interface of the Matplotlib Module, an object-oriented plotting library of Python.
* **Seaborn:** Seaborn is an open-source Python library built on top of matplotlib. It is used for data visualization and exploratory data analysis.
* **Os:** The OS module in Python provides functions for interacting with the operating system.

**2. Load, Merge and sort All the Dataset:**

* **Loading the files:** Using the glob library we are reading all the files.
* **Merge:** Merging all the files using glob library.
* **Sort:** Sorting all the file paths using sorted keyword.

# **Reading Files individually and Combining into Single File:**

# Using pandas concat() we are reading and concating all the files using file paths and make it into a single csv file.

# **Removing Unnamed Column:**

# Removing unnecessary column name which is named as unnamed column from the combined data set using loc.

# **Removing Null Values:**

# Dropna is used for the removing all the null values in the dataset.

# **Extracting The Year From File Name and Appending Year to the List:**

# **Using os.path.basename we are extracting the year from the file name then we are reading all the files to find the count of the total rows and repeatly appending year upto the total row count.**

# **(os.path.basename(path) :** It is used to return the basename of the file . This function basically return the file name from the path given.)

# **Insert A List To The Year Column:**

# Inserting the year list into the year column on the data set.

# **Displaying All the Rows With Year Column.**

# **Exploratory Data Analysis (EDA):**

EDA is a phenomenon under data analysis used for gaining a better understanding of data aspects like: 

* Main features of data
* Variables and relationships that hold between them
* Identifying which variables are important for our problem

## **Descriptive Statistics:**

Descriptive statistics is a helpful way to understand characteristics of your data and to get a quick summary of it. Pandas in python provide an interesting method **describe()**. The describe function applies basic statistical computations on the dataset like extreme values, count of data points standard deviation etc. Any missing value or NaN value is automatically skipped. describe() function gives a good picture of distribution of data. 

# **Pandas DataFrame info() Method:**

The **info()** method prints information about the DataFrame.

The information contains the number of columns, column labels, column data types, memory usage, range index, and the number of cells in each column (non-null values).

# **Pandas DataFrame shape Property:**

The **shape** property returns a tuple containing the shape of the DataFrame.

The shape is the number of rows and columns of the DataFrame.

**Pandas DataFrame dtypes Property:**

The **dtypes** property returns data type of each column in the DataFrame.

# **Implementing**

# Taking one year value and display the all ages values in it.

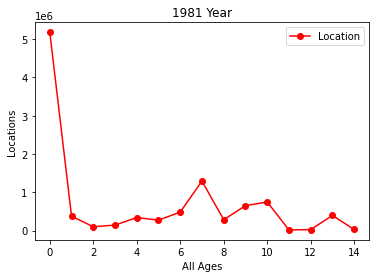
# **Line Chart:**

A line chart or line plot or line graph or curve chart is a type of chart which displays information as a series of data points called 'markers' connected by straight line segments. It is a basic type of chart common in many fields.

 A line chart is often used to visualize a trend in data over intervals of time.  This variation is usually plotted in a two-dimensional XY plane. Thus, the line graph is also called a **linear graph**.

X-Axis = All Ages

Y-Axis = Locations



1. **Pie Chart:**

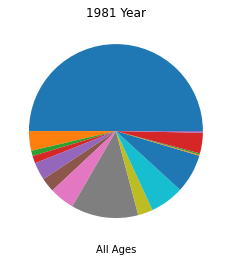
A pie chart is a circular statistical graphic, which is divided into slices to illustrate numerical proportion. In a pie chart, the arc length of each slice is proportional to the quantity it represents.

It contains different segments and sectors in which each segment and sector of a pie chart forms a specific portion of the total(percentage). The sum of all the data is equal to 360°.

**Formula**:

**(Given Data / Total value of Data) × 360°**

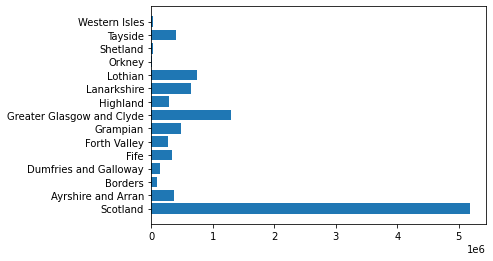
**The total value of the pie is always 100%.**



**13.Bar Chart:**

**Bar charts** are the pictorial representation of data (generally grouped), in the form of vertical or horizontal rectangular bars, where the length of bars are proportional to the measure of data. They are also known as bar charts. Bar graphs are one of the means of data handling in statistics.

Bar charts consist of multiple price bars, with each bar illustrating how the price of an asset or security moved over a specified time period. Each bar typically shows open, high, low, and closing (OHLC) prices, although this may be adjusted to show only the high, low, andclose (HLC).



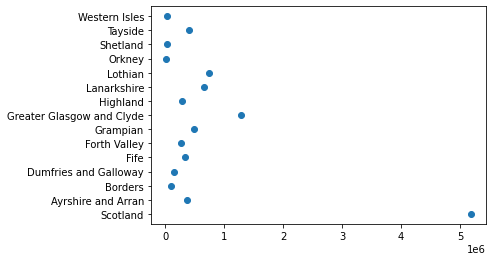
**14. Scatter Plot**:

**Scatter plots** are the graphs that present the relationship between two variables in a data-set. It represents data points on a two-dimensional plane or on a **Cartesian system.**

The independent variable or attribute is plotted on the X-axis, while the dependent variable is plotted on the Y-axis. These plots are often called **scatter graphs** or **scatter diagrams**.

X-Axis = All Ages

Y-Axis = Locations

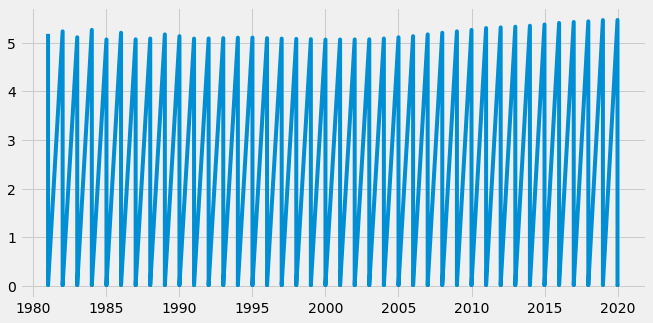


**15. Plotting Graph:**

**1. The Graph for ALL AGES has been ploted.**

X-Axis = Years(1980-2020)

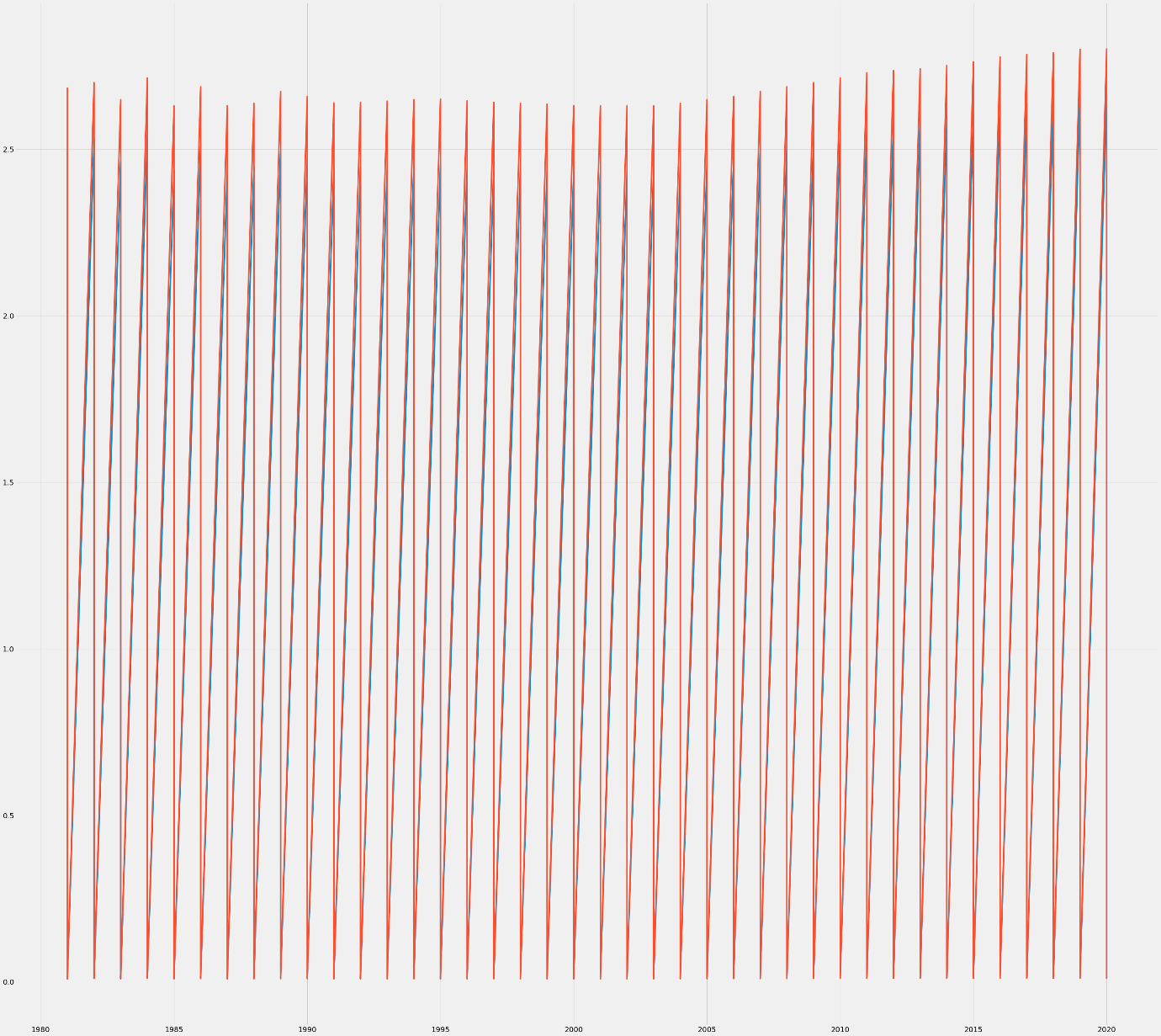
Y-Axis = All ages



**2. The Graph for COMPARING MALE ALL AGES and FEMALE ALL AGES has been ploted.**

X-Axis = Years(1980-2020)

Y-Axis =Male & Female All ages

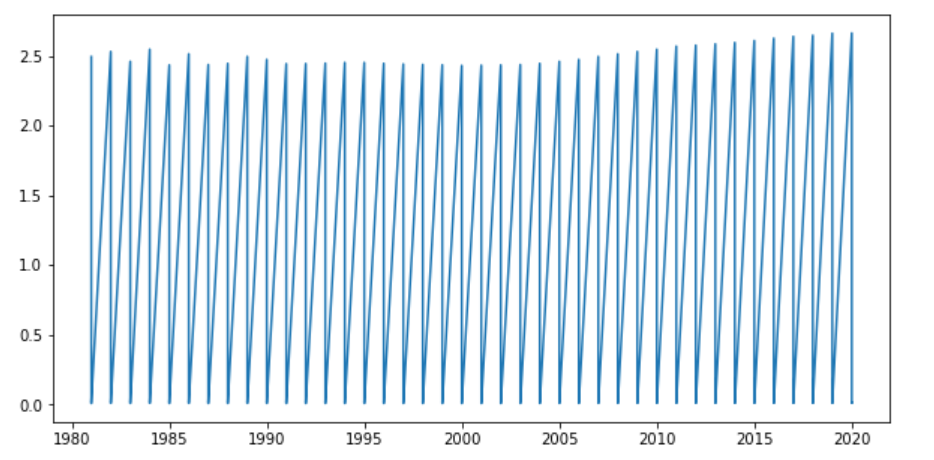


**3. A seperate Graph for MALE ALL AGES and FEMALE ALL AGES has been ploted.**

**MALE ALL AGES Graph:**

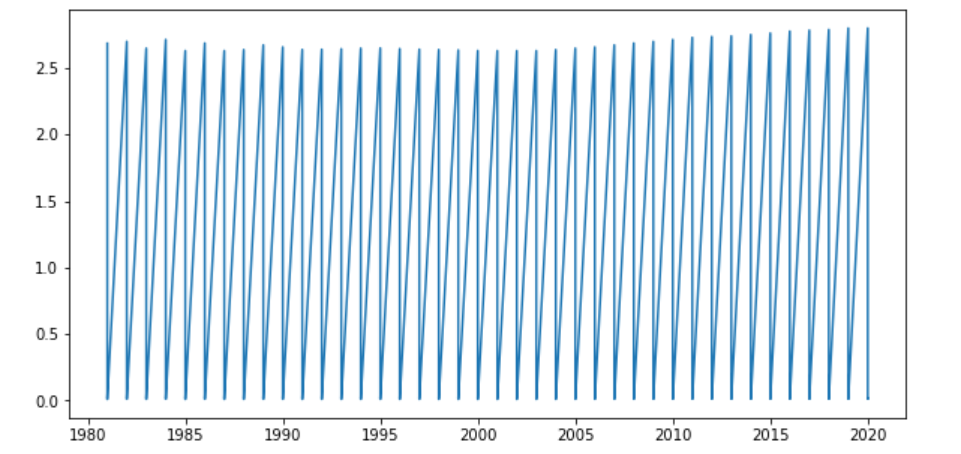
X-Axis = Years(1980-2020)

Y-Axis = Male All ages



**FEMALE ALL AGES Graph:**

X-Axis = Years(1980-2020) , Y-Axis = Female All ages

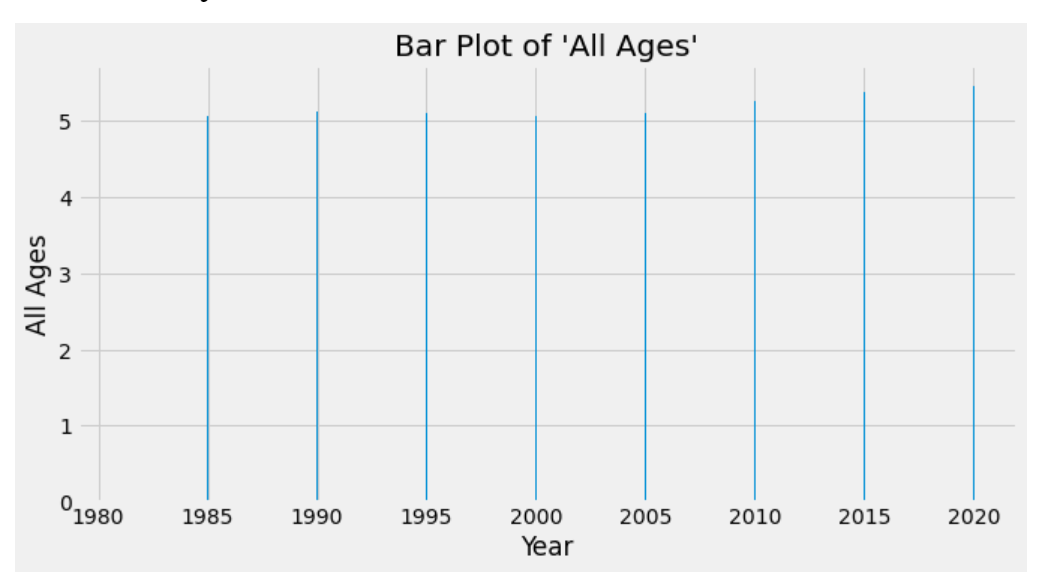


**BAR PLOT:**

**4.The Bar Plot for ALL AGES has been ploted.**

X-Axis = Years(1980-2020)

Y-Axis = All ages



1. **Fetching one Location from all files:**

Using loc() , fetching a one Location from all files in DataFrame.

**Time Series Decomposition:**

Decomposition is a statistical task in which the Time Series data is decomposed into several component or extracting seasonality, trend from a series data.

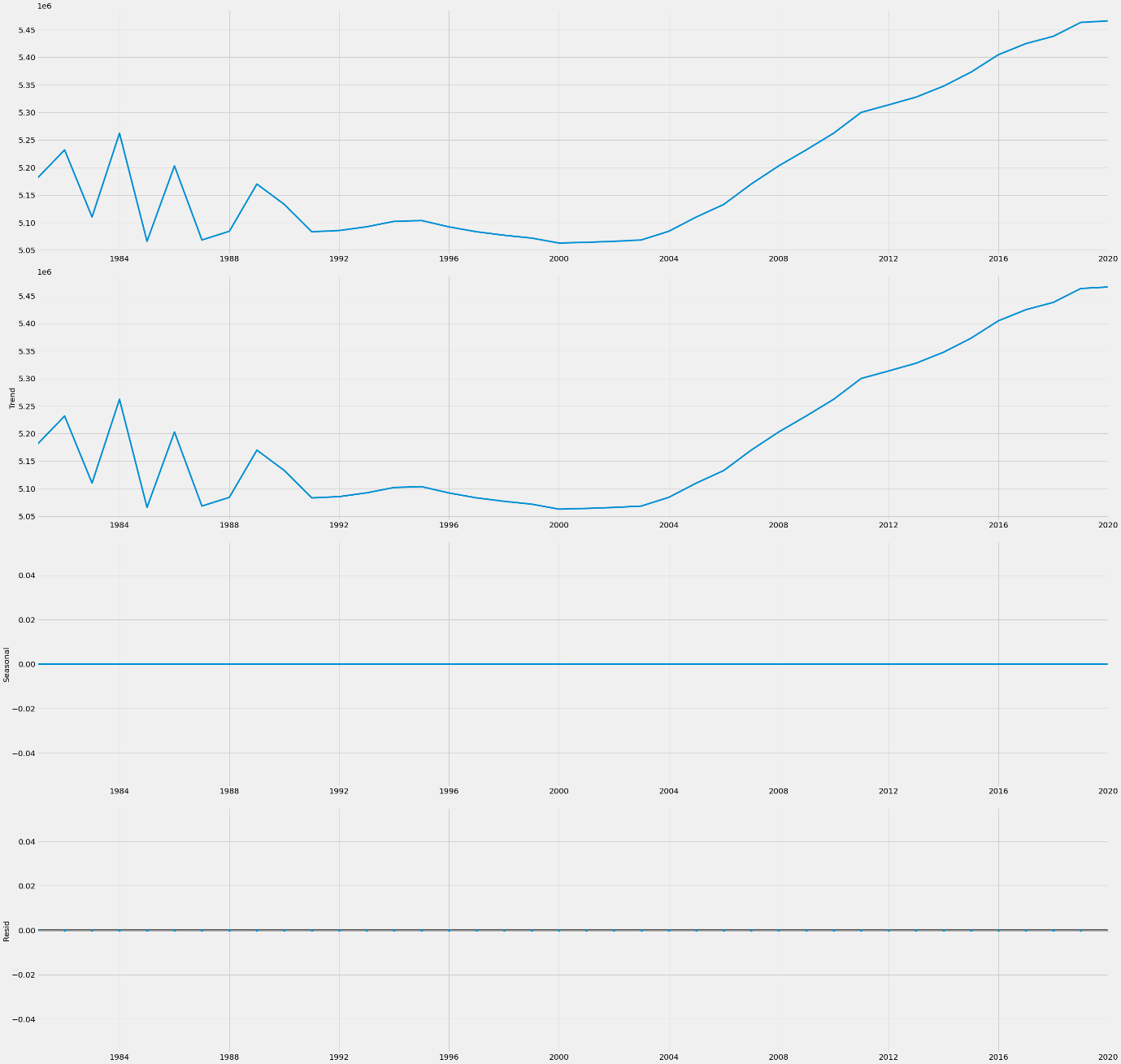
**Seasonality**: describes the periodic signal in your time series.  
 **Trend**: describes whether the time series is decreasing, constant, or increasing over time.  
 **Noise**: describes what remains behind the separation of seasonality and trend from the time series. In other words, it’s the variability in the data that cannot be explained by the model

**17. The Decomposition:**

In seasonal\_decompose we have to set the model. We can either set the model to be **Additive** or **Multiplicative.**

If it is linear then we will select the **Additive** model. Otherwise, if the trend and seasonal variation increase or decrease over time then we use the **Multiplicative** model.

**Note :** Our index here has Yearly frequency. If DataFrame does not have a frequency then we need to set a period.



**Moving average:**

we can calculate the moving average **using .** **rolling() method**.

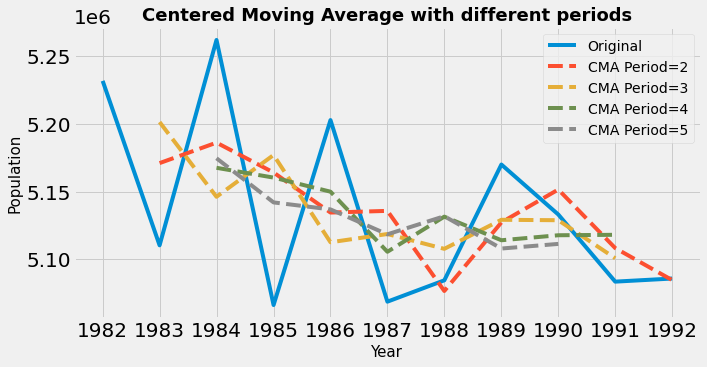
This method provides rolling windows over the data, and we can use the mean function over these windows to calculate moving averages.

The size of the window is passed as a parameter in the function

**Moving Average Period:**

The length of a moving average period, or simply **moving average period**, means how many bars are used for calculating the moving average.

When you are selecting a moving average period length, you are deciding how far back to the history you want to look



**Cheking Time series of given data is Stationary or Not :**

**Dickey-Fuller Test:**

Dickey–Fuller test tests the [null hypothesis](https://en.wikipedia.org/wiki/Null_hypothesis) that a [unit root](https://en.wikipedia.org/wiki/Unit_root) is present in an [autoregressive](https://en.wikipedia.org/wiki/Autoregressive) time series model. The [alternative hypothesis](https://en.wikipedia.org/wiki/Alternative_hypothesis) is different depending on which version of the test is used, but is usually [stationarity](https://en.wikipedia.org/wiki/Stationarity_(statistics)) or [trend-stationarity](https://en.wikipedia.org/wiki/Trend-stationary_process)

**Null Hypothesis (H0)**: If failed to be rejected, it suggests the time series has a unit root, meaning it is non-stationary. It has some time dependent structure.

**Alternate Hypothesis (H1)**: The null hypothesis is rejected; it suggests the time series does not have a unit root, meaning it is stationary. It does not have time-dependent structure

**Interpret this result using the p-value from the test:**

A p-value below a threshold (such as 5% or 1%) suggests we reject the null hypothesis (stationary), otherwise a p-value above the threshold suggests we fail to reject the null hypothesis (non-stationary).

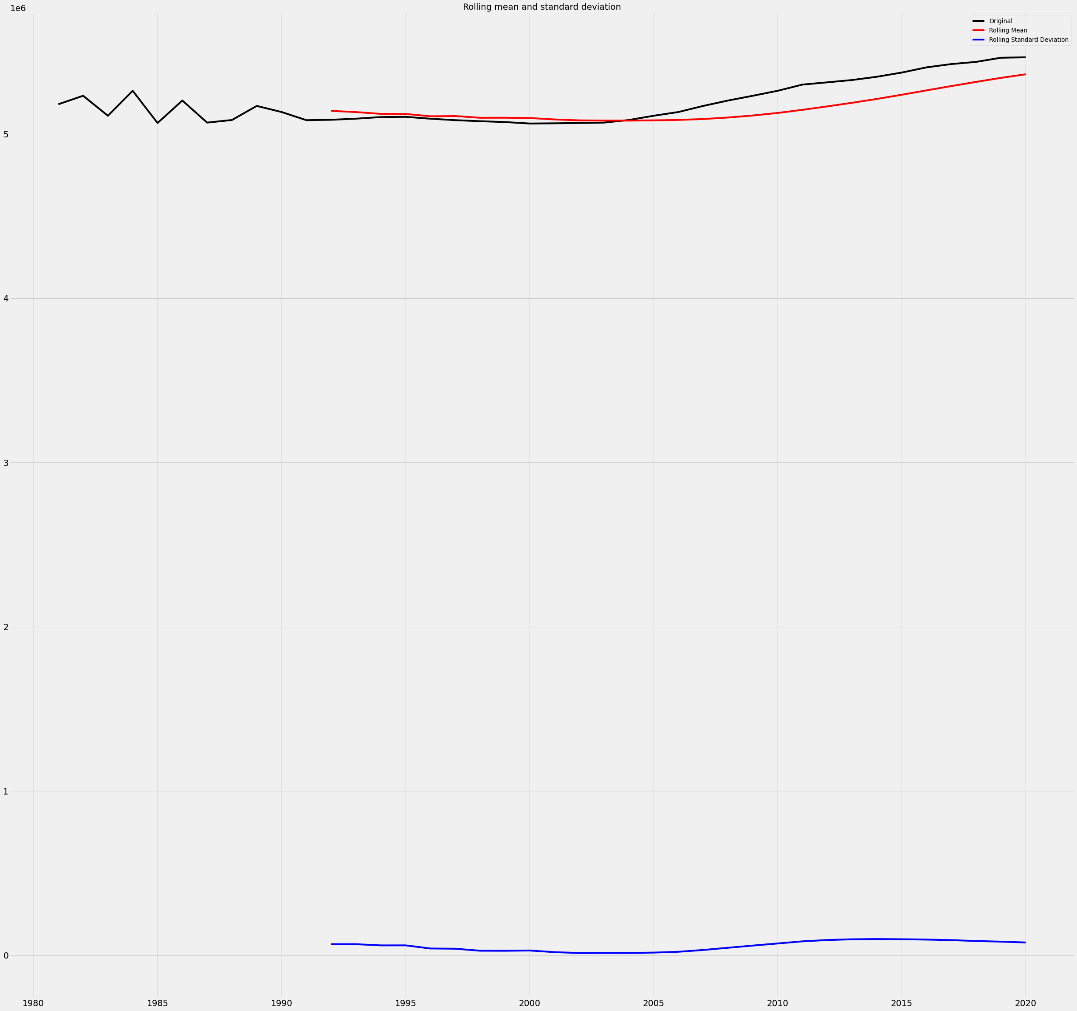
**p-value > 0.05**: Fail to reject the null hypothesis (H0), the data has a unit root and is non-stationary.

**p-value <= 0.05**: Reject the null hypothesis (H0), the data does not have a unit root and is stationary

**Rolling Mean and Standard Deviation:**

**Rolling Mean:** A rolling mean is an average for a window of data, where the window is a series of sequential values from the data

**Rolling standard Deviation:** Rolling Standard Deviation is a statistical measurement of market volatility. It makes no predictions of market direction, but it may serve as a confirming indicator. You specify the number of periods to use, and the study computes the standard deviation of prices from the moving average of the prices.



**Stats models**:

[stats models](https://www.statsmodels.org/stable/about.html#about-statsmodels) is a Python module that provides classes and functions for the estimation of many different statistical models, as well as for conducting statistical tests, and statistical data exploration. An extensive list of result statistics are available for each estimator. The results are tested against existing statistical packages to ensure that they are correct.

Stats models supports specifying models using R-style formulas and pandas DataFrames. You can also use numpy arrays instead of formulas.

**Augmented Dickey Fuller Test (ADF Test):**

Augmented Dickey Fuller test (ADF Test) is a common statistical test used to test whether a given Time series is stationary or not. It is one of the most commonly used statistical test when it comes to analysing the stationary of a series.

The statsmodel package provides a reliable implementation of the ADF test via the adfuller() function in statsmodels.tsa.stattools. It returns the following outputs:

1. The p-value
2. The value of the test statistic
3. Number of lags considered for the test
4. The critical value cutoffs.

## **The P-value:**

The P-value is a statistical number to conclude if there is a relationship between any two factors.

We test if the true value of the coefficient is equal to zero (no relationship). The statistical test for this is called Hypothesis testing.

* A low P-value (< 0.05) means that the coefficient is likely not to equal zero.
* A high P-value (> 0.05) means that we cannot conclude that the explanatory variable affects the dependent variable.
* A high P-value is also called an insignificant P-value.

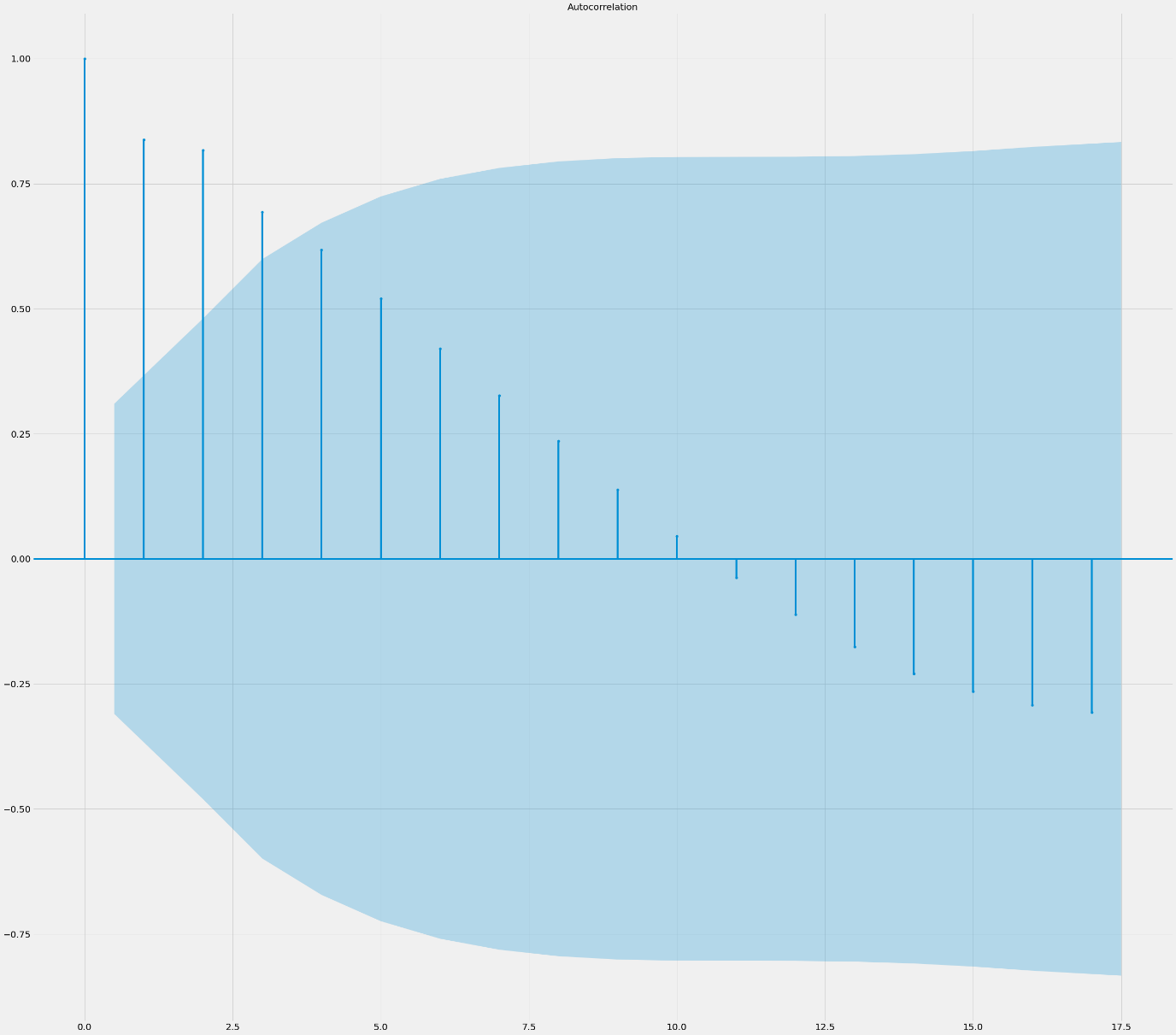
# **Autocorrelation in Python:**

Autocorrelation measures the degree of similarity between a time series and a lagged version of itself over successive time intervals.

It’s also sometimes referred to as “serial correlation” or “lagged correlation” since it measures the relationship between a variable’s current values and its historical values.

When the autocorrelation in a time series is high, it becomes easy to predict future values by simply referring to past values.

Autocorrelation is calculated as follows:

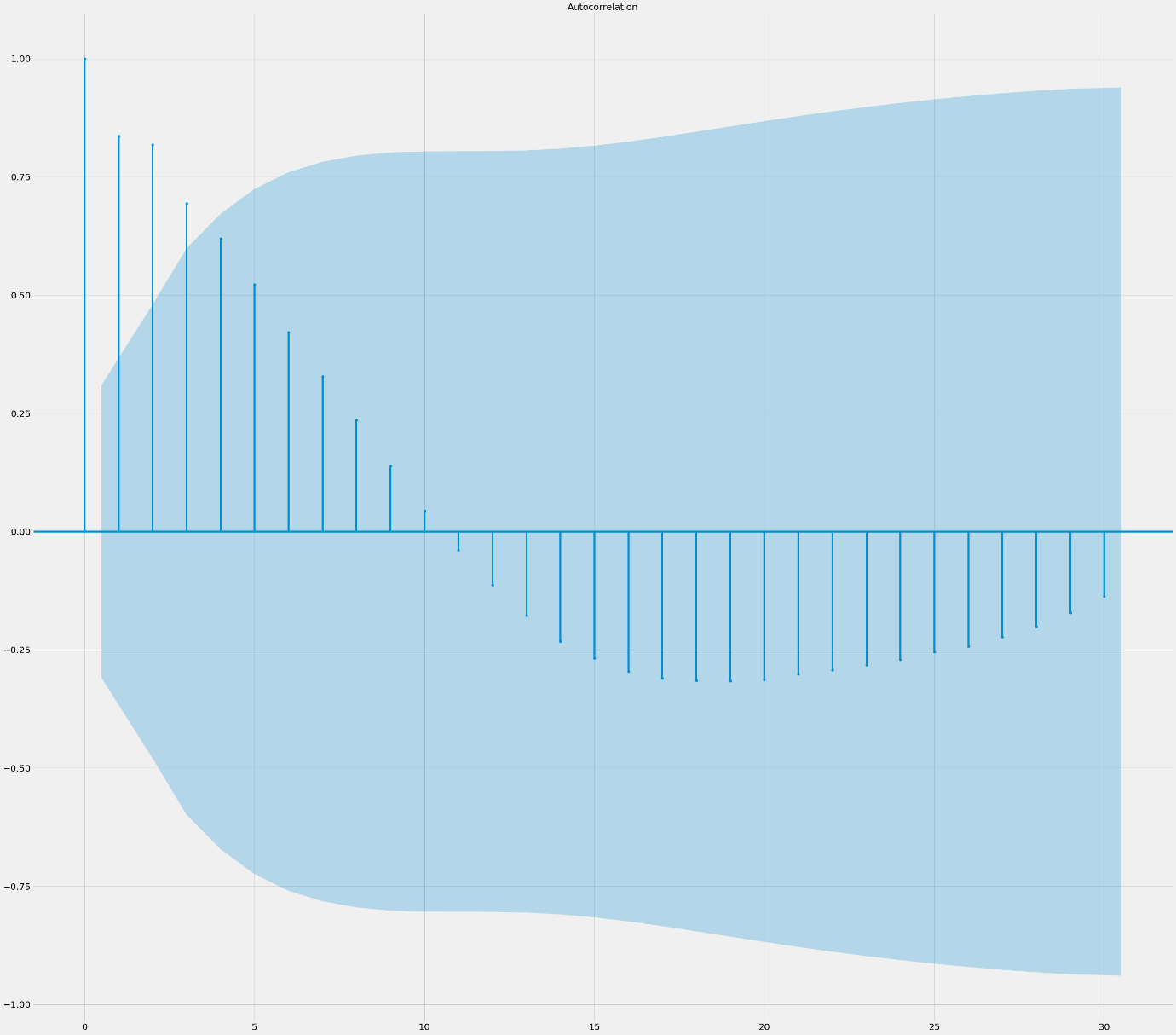


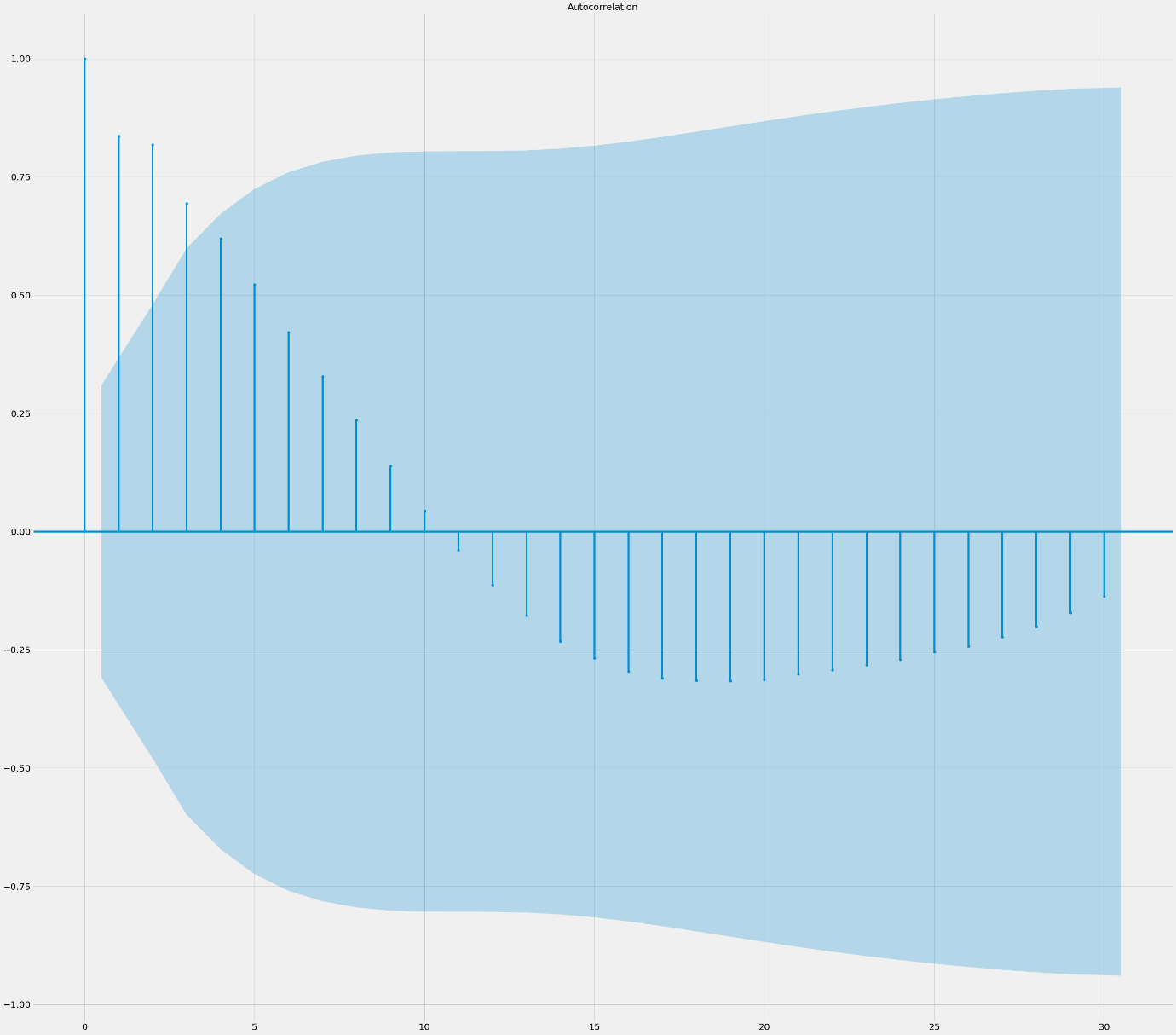
**Set number of lags in Python pandas autocorrelation\_plot:**

The autocorrelation\_plot function is a high level function.

If one does not need to use pandas methods. There is a statsmodels function plot\_acf in which you can set the lags argument.

When lag is given as 30, the graph is plotted as follows:

****

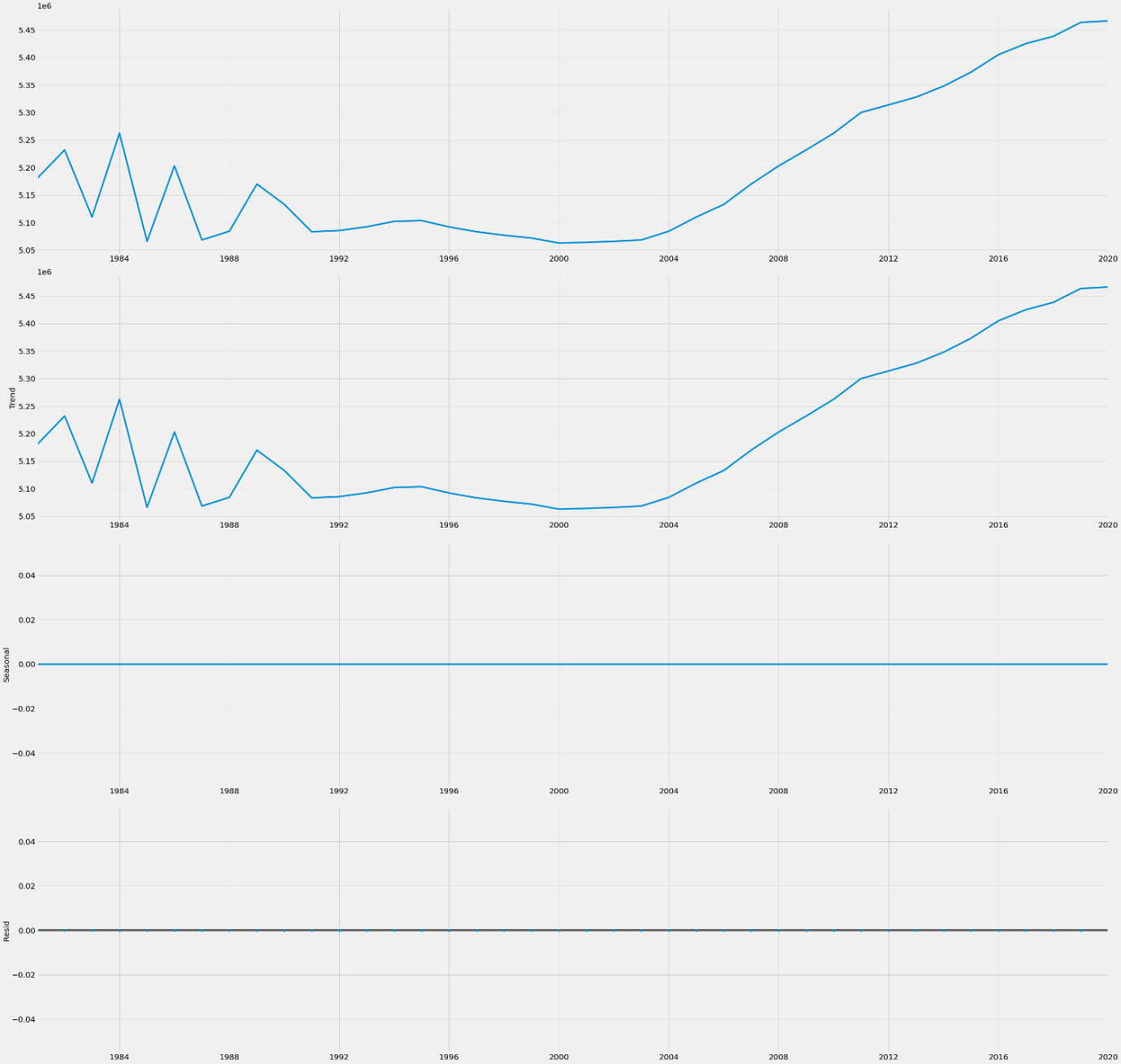
****

**Seasonal Decomposition:**

### The stats model library in Python has a [seasonal decompose](https://www.statsmodels.org/stable/generated/statsmodels.tsa.seasonal.seasonal_decompose.html) function that does just this. Given a time series of data, the function splits into separate trend, seasonality, and residual (noise) components.

After loading and reformatting the data, the date and metric will be fed into this function to parse out the separate pieces.

Seasonal decomposition is plotted as follows:



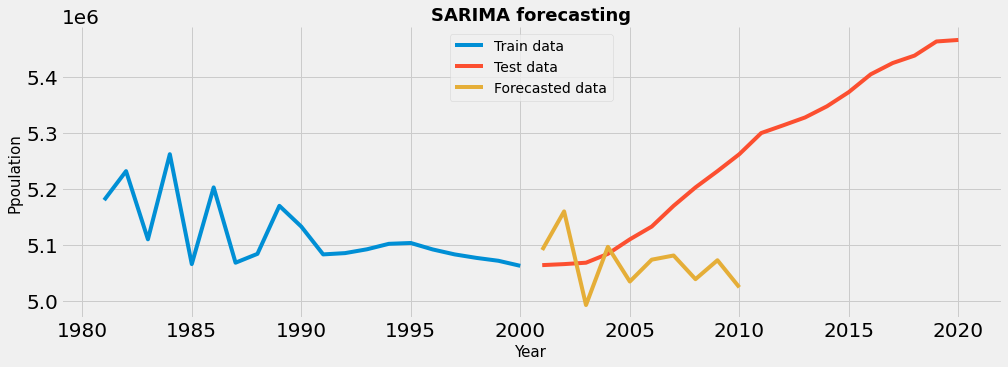
**SARIMA Using Python – Forecast Seasonal Data:**

SARIMA is Seasonal ARIMA, or simply put, ARIMA with a seasonal component. As mentioned above, ARIMA is a statistical analysis model that uses time-series data to either better understand the data set or to predict future trends.

It consists of 3 components –

| **Component** | **Explanation** |
| --- | --- |
| **Autoregressive** | A model that uses the dependent relationship between an observation and some number of lagged observations. |
| **Integrated** | The use of differencing of raw observations (e.g. subtracting an observation from observation at the previous time step) in order to make the time series stationary. |
| **Moving Average** | A model that uses the dependency between an observation and a residual error from a moving average model applied to lagged observations. |

The SARIMA model is plotted as follows:



# **ARIMA Model:**

The Autoregressive Integrated Moving Average (ARIMA) model uses time-series data and statistical analysis to interpret the data and make future predictions. The ARIMA model aims to explain data by using time series data on its past values  to make predictions.

* The ARIMA model uses statistical analyses in combination with accurately collected historical data points to predict future trends and business needs.
* The ARIMA model is typically denoted with the parameters (*p, d, q*), which can be assigned different values to modify the model and apply it in different ways.
* Some of the limitations of the model are its dependency on data collection and the manual trial-and-error process required to determine parameter values that fit best.

**ARIMA (*p, d, q*)**

* The parameter ***p***is the number of autoregressive terms or the number of “lag observations.” It is also called the “lag order,” and it determines the outcome of the model by providing lagged data points.
* The parameter ***d***is known as the degree of differencing. it indicates the number of times the lagged indicators have been subtracted to make the data stationary.
* The parameter ***q*** is the number of forecast errors in the model and is also referred to as the size of the moving average window.

The parameters take the value of integers and must be defined for the model to work. They can also take a value of 0, implying that they will not be used in the model. In such a way, the ARIMA model can be turned into:

* ARMA model (no stationary data, ***d*= 0**)
* AR model (no moving averages or stationary data, just an autoregression on past values, ***d*= 0, *q*= 0**)
* MA model (a moving average model with no autoregression or stationary data, ***p*= 0, *d* = 0)**

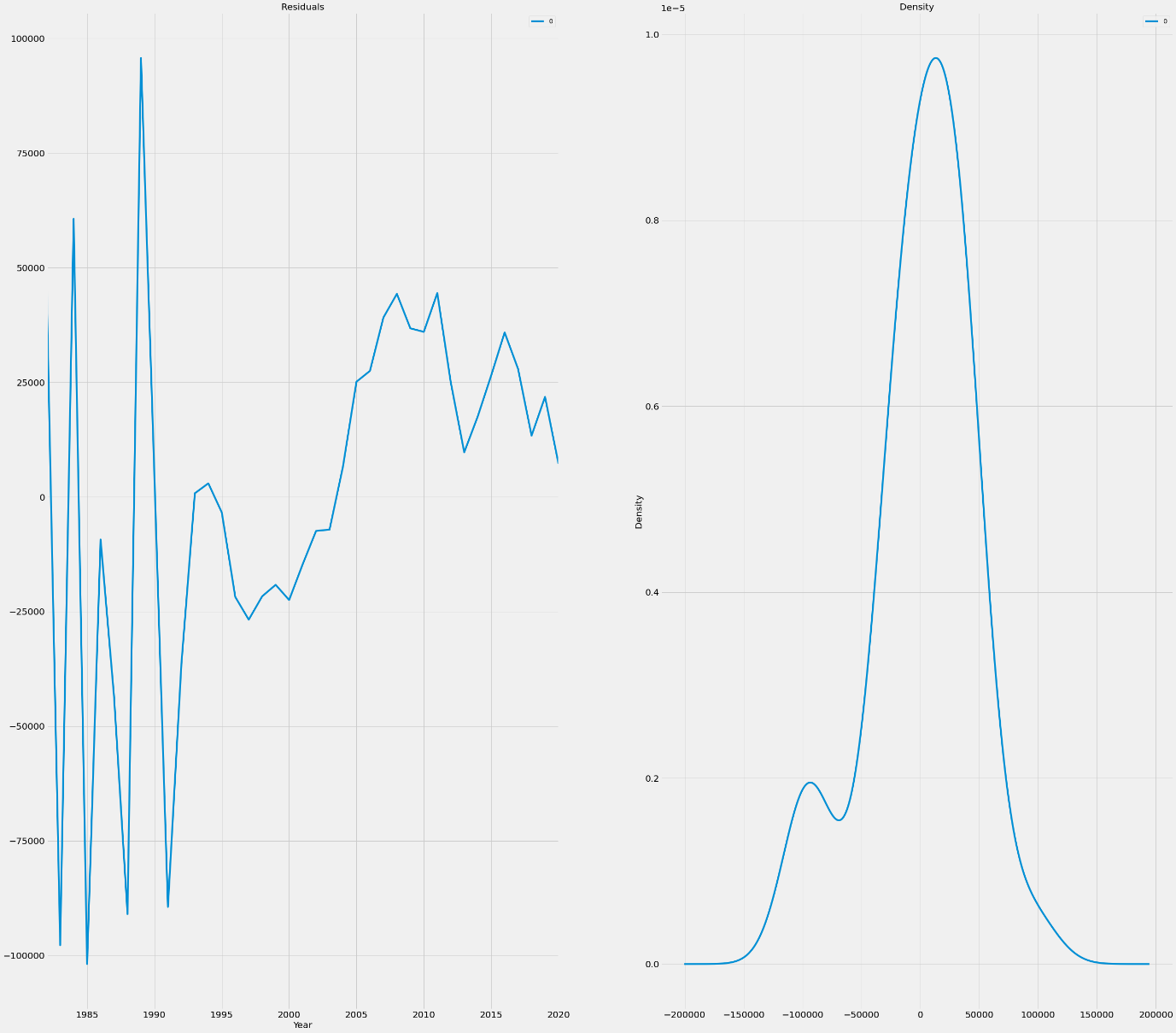
Therefore, ARIMA models may be defined as:

* ARIMA(1, 0, 0) – known as the **first-order autoregressive model**
* ARIMA(0, 1, 0) – known as the **random walk model**
* ARIMA(1, 1, 0) – known as the **differenced first-order autoregressive model**, and so on.

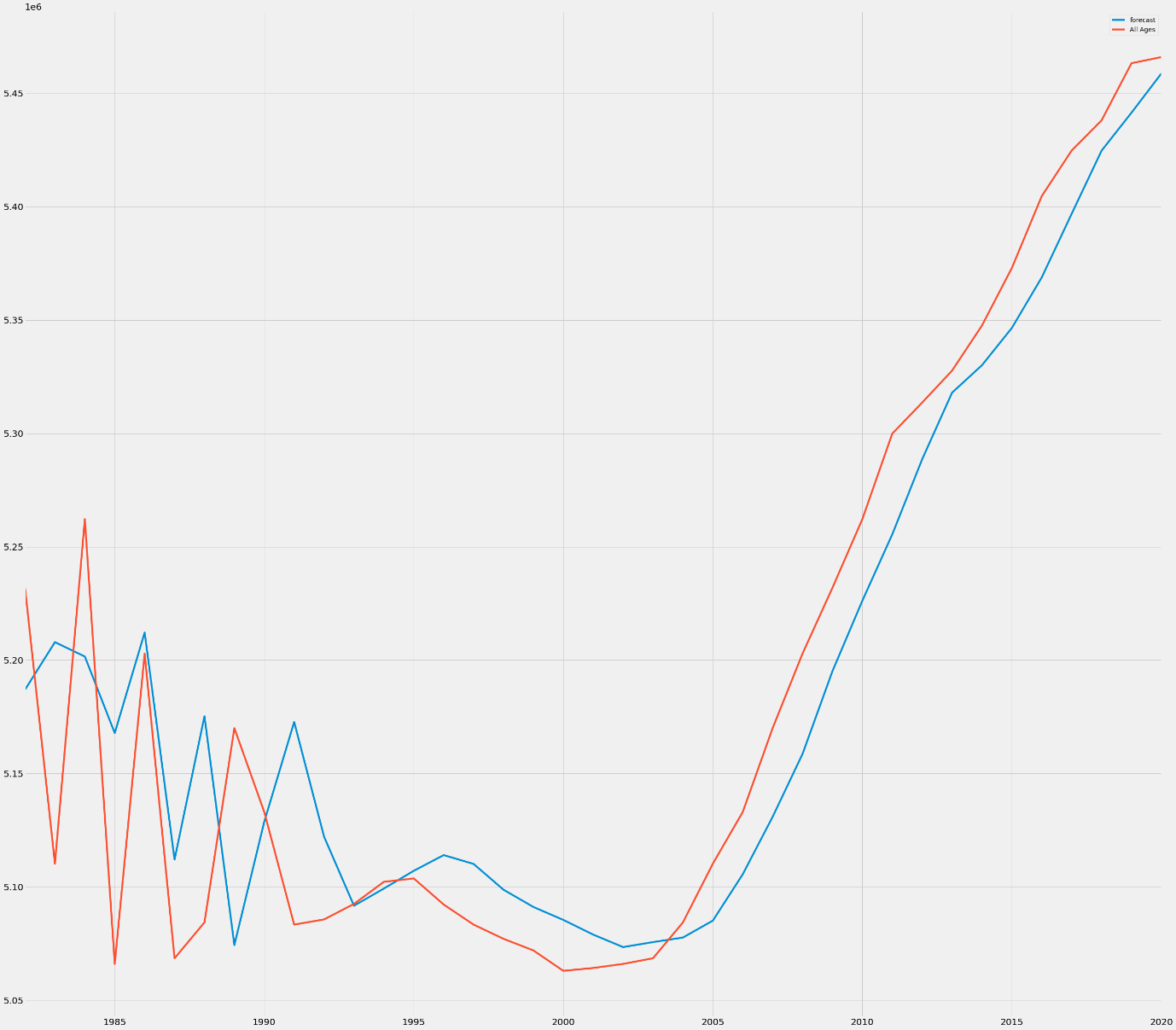
Once the parameters (*p, d, q*) have been defined, the ARIMA model aims to estimate the coefficients **α** and **θ**, which is the result of using previous data points to forecast values.

# **Residual Error & Density :**

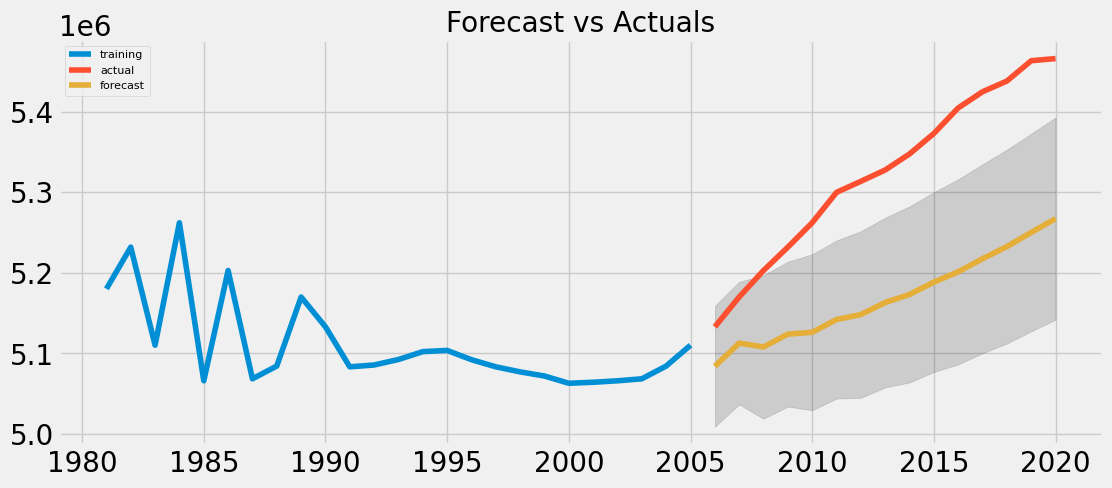
In [statistics](https://en.wikipedia.org/wiki/Statistics) and [optimization](https://en.wikipedia.org/wiki/Mathematical_optimization), **errors** and **residuals** are two closely related and easily confused measures of the [deviation](https://en.wikipedia.org/wiki/Deviation_(statistics)) of an [observed value](https://en.wikipedia.org/wiki/Observed_value) of an [element](https://en.wikipedia.org/wiki/Elementary_event) of a [statistical sample](https://en.wikipedia.org/wiki/Sample_(statistics)) from its "[true value](https://en.wikipedia.org/wiki/True_value)" (not necessarily observable). The **error** of an [observation](https://en.wikipedia.org/wiki/Observation) is the deviation of the observed value from the true value of a quantity of interest (for example, a [population mean](https://en.wikipedia.org/wiki/Population_mean)). The **residual** is the difference between the observed value and the [*estimated*](https://en.wikipedia.org/wiki/Estimation) value of the quantity of interest (for example, a [sample mean](https://en.wikipedia.org/wiki/Sample_mean)). The distinction is most important in [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis), where the concepts are sometimes called the **regression errors** and **regression residuals** and where they lead to the concept of [studentized residuals](https://en.wikipedia.org/wiki/Studentized_residual).



# **Actual Data versus Forecasting Data:**

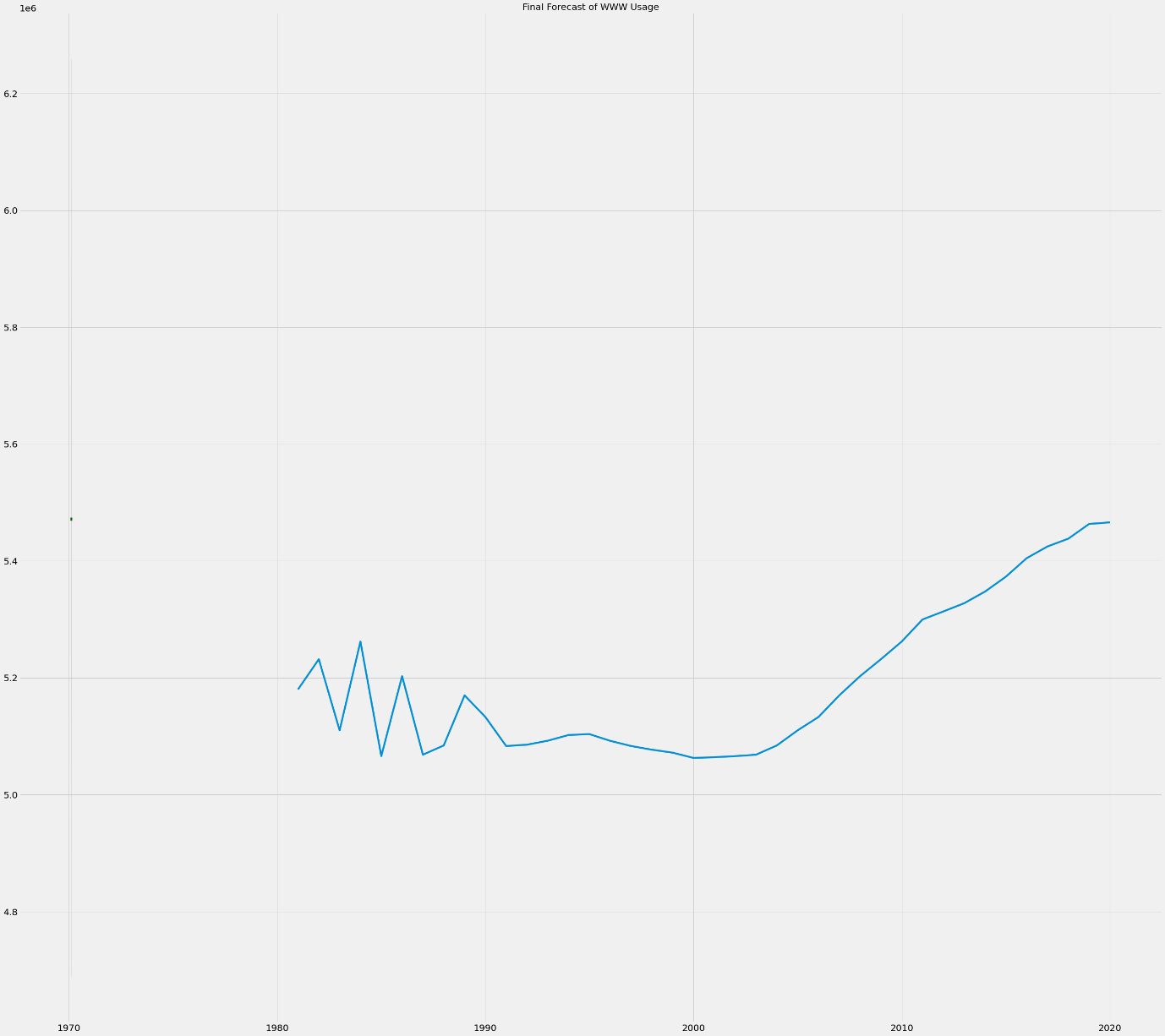


# **Forecasting Data versus Actual Data**



**AUTO ARIMA MODEL:**

# Forecast Graph using Auto Arima

****

# **aic, bic, hqic value using Arima Decomposition:**

**1. Akaike Information Criterion(AIC)**

AIC = -2ln(L) + 2k

where L is the value of likelihood function

k is the number of estimated parameters

**2. Bayesian Information Criterion(BIC)**

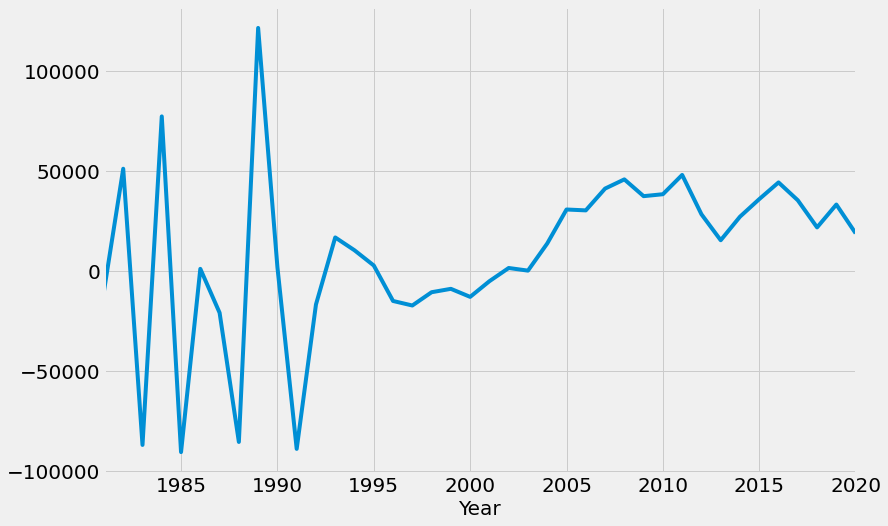
BIC = -2ln(L) + ln(N)k

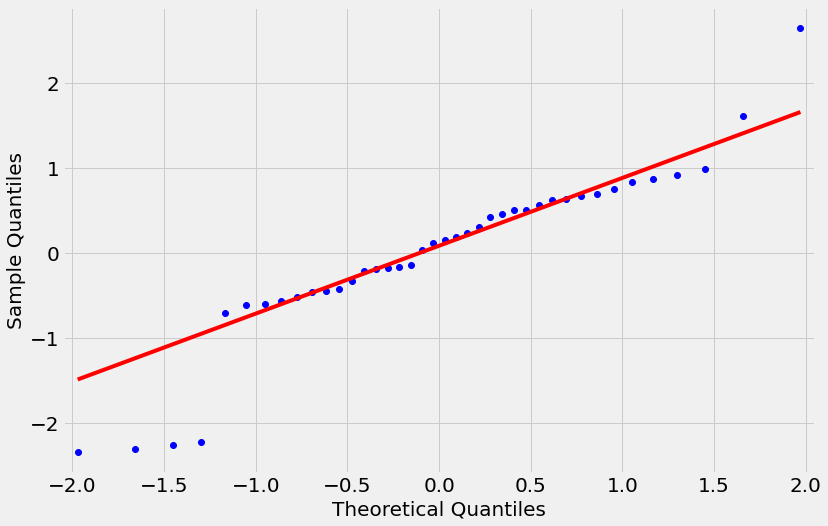
where L is the value of likelihood function

N is the number of observations

k is the number of estimated parameters

# **Residual Plot:**



**Residual Graph:**

**Autocorrelation**

Autocorrelation is the measure of the degree of similarity between a given time series and the lagged version of that time series over successive time periods. It is similar to calculating the correlation between two different variables except in Autocorrelation we calculate the correlation between two different versions Xt and Xt-k of the same time series.

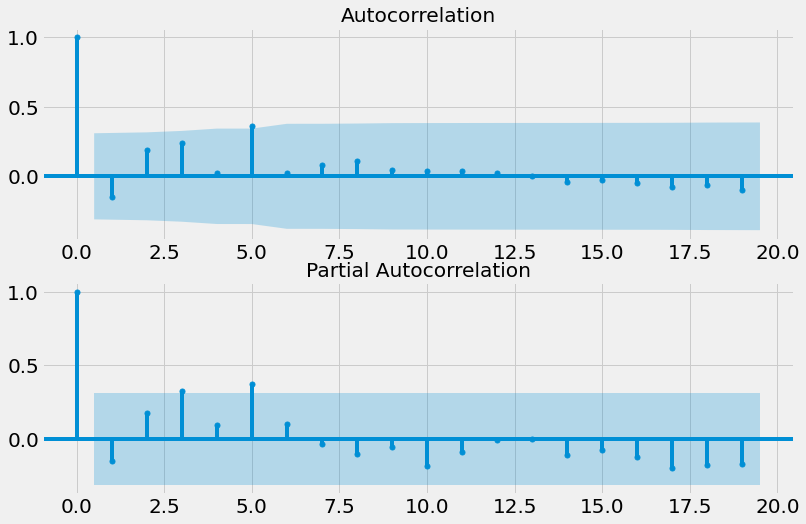
An autocorrelation test is used to detect randomness in the time-series. In many statistical processes, our assumption is that the data generated is random. For checking randomness, we need to check for the autocorrelation of lag 1.To determine whether there is a relation between past and future values of time series, we try to lag between different values.

**Partial Autocorrelation**

In time series analysis, the **partial autocorrelation function** (**PACF**) gives the partial correlation of a stationary time series with its own lagged values, regressed the values of the time series at all shorter lags. It contrasts with the autocorrelation function, which does not control for other lags.

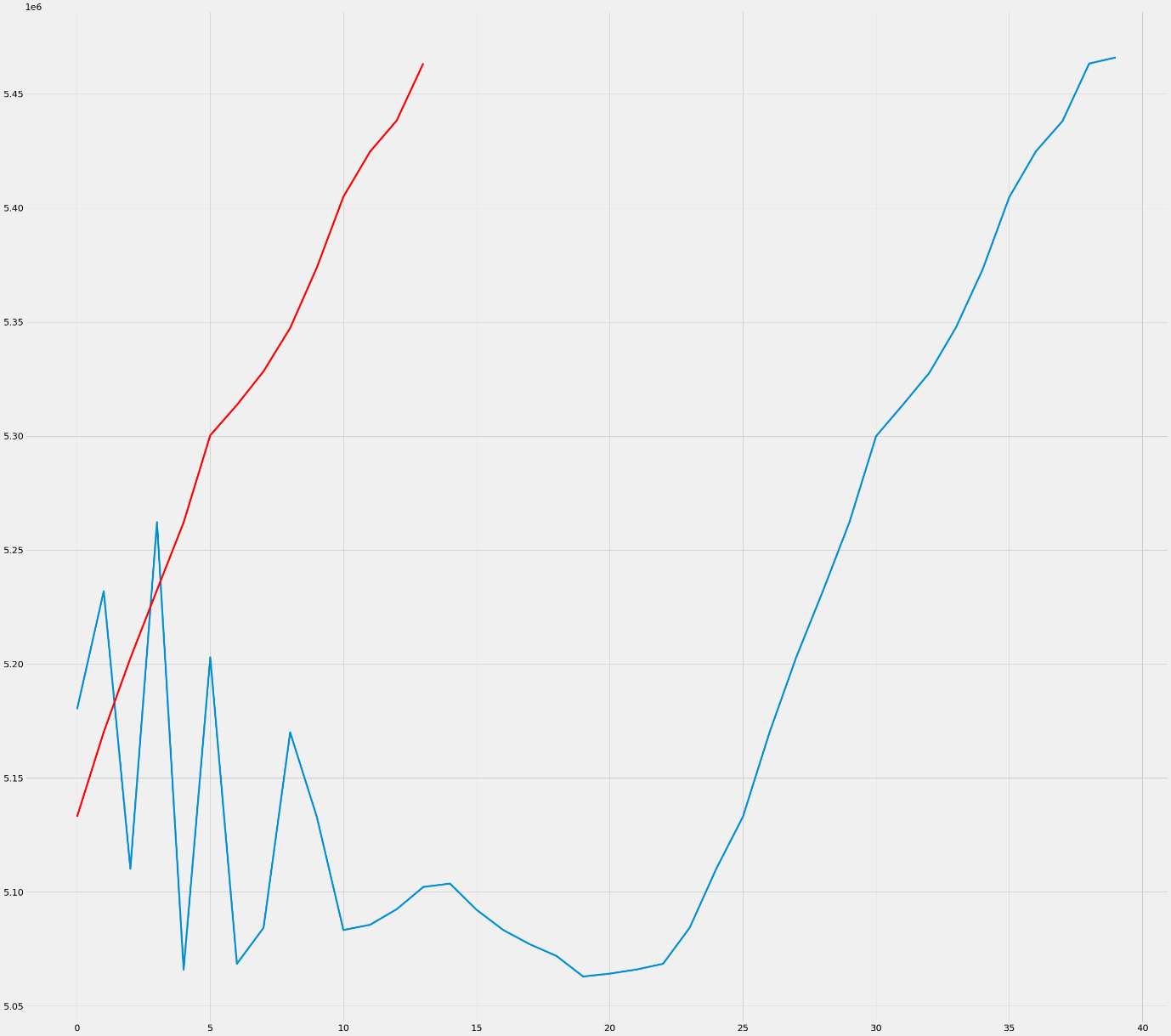
This function plays an important role in data analysis aimed at identifying the extent of the lag in an autoregressive model, The use of this function was introduced as part of the Box jekins approach to time series modelling, whereby plotting the partial autocorrelative functions one could determine the appropriate lags **p** in an AR (**p**) model or in an extended ARIMA (**p**,**d**,**q**) model.

Graphical representation of **Autocorrelation** and **Partial autocorrelation:**

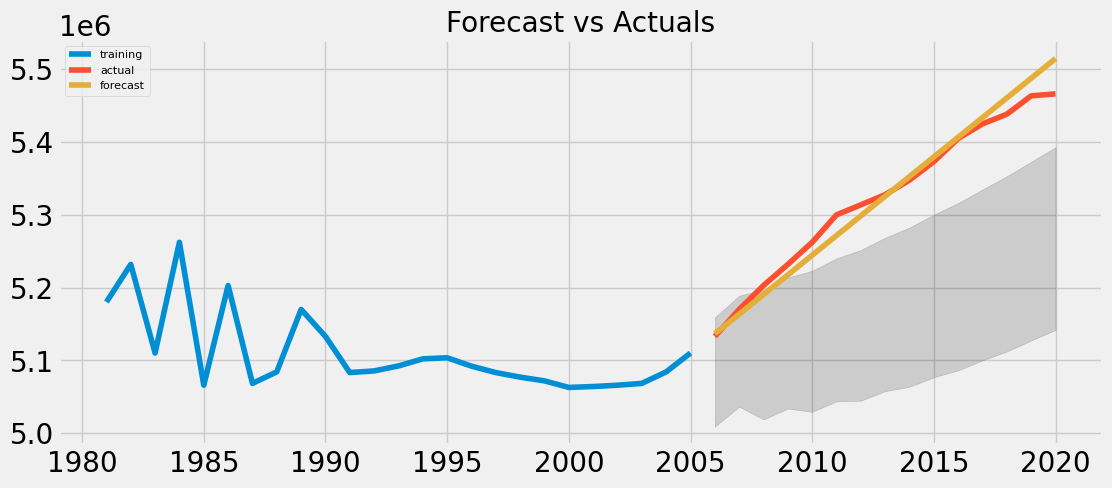


The above graph represents **Autocorrelation** and **Partial autocorrelation of population projecton.**

# **Train, Test, Walkford Validation & Forecast Evoluation**

****

**Final Graph Actual vs Forecast:**

****

1. **CONCLUSION**

We summarise the conclusions of our survey of population projection developments. The activity of evaluating population forecasts has increased in recent years. This reﬂects the growing availability of time series of data to be assessed. Reﬂection on past errors is a great help in improving methods and assumptions for the future. Analysis of past errors has been used to develop new ways of coping with future uncertainty. However, the purpose of the projection should be borne in mind when comparing errors between projections. We will learn a great deal from scenario projections, although most will be far from the most likely future.

1. **REFERENCES**

[1] Abello A, Kelly S, King A. 2002. Demographic Projections with DYNAMOD-2. NATSEM Technical Paper No. 21. National Centre for Social and Economic Model-ling, University of Canberra. <http://www.natsem>. canberra.edu.au/publication.jsp?titleID=TP21[accessed 11 June 2005].

[2] Abernethy VD. 2001. Comment on Bermingham’s summary of the UN’s year 2000 ‘Replacement Migration: Is it a Solution to Declining Population and Aging?’ Population and Environment 22(4): 365–375. DOI: 10.1023/A:1006741420885.