PREDICTION OF WATER CONSUPTION IN A RESERVOIR MANAGEMENT SYSTEM USING ML ALGORITHM

# A report on project phase - 1

***in partial fulfilment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

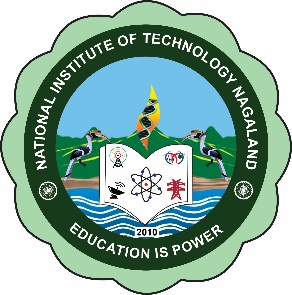
***In***

**ELECTRONICS AMD INSTRUMENTATION ENGINEERING**

***By***

**ASHISH MISHRA**

**Registration No. 2017107008**

****

**DEPARTMENT OF**

**ELECTRONIC AND INSTRUMENTATION ENGINEERING**

**NATIONAL INSTITUTE OF TECHNOLOGY, NAGALAND**

**DIMAPUR 797103**

**March 2021**

PREDICTION OF WATER CONSUPTION IN A RESERVOIR MANAGEMENT SYSTEM USING ML ALGORITHM

# B. TECH PROJECT – PHASE 1

Submitted by

**ASHISH MISHRA**

*in partial fulfilment for the award of the degree*

*of*

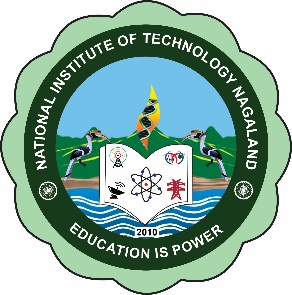
**BACHELOR OF TECHNOLOGY**

In

**ELECTRONICS AND INSTRUMENTATION ENGINEERING**

Under the supervision of

**Dr. D GANGA**

****

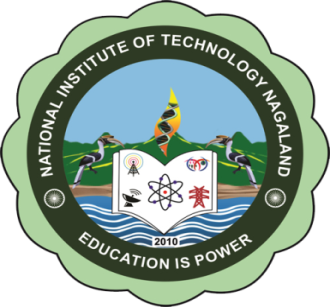
**DEPARTMENT OF**

**ELECTRONIC AND INSTRUMENTATION ENGINEERING**

**NATIONAL INSTITUTE OF TECHNOLOGY, NAGALAND**

**DIMAPUR 797 103**

**MARCH 2021**



**राष्ट्रीय प्रौद्योगिकी संस्थान, नागालैंड**

**NATIONAL INSTITUTE OF TECHNOLOGY, NAGALAND**

**(An Institute of National Importance under Ministry of HRD, Govt Of India)**

**Chumukedima, Dimapur Nagaland - 797 103**

**BONAFIDE CERTIFICATE**

Certified that this Project titled “PREDICTION OF WATER CONSUPTION IN A RESERVOIR MANAGEMENT SYSTEM USING ML ALGORITHM” is the Bonafide work of ASHISH MISHRA (2017107008) and who carried out the work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other student.

**Dr. D. GANGA**

Project Guide

**Asst. Professor**

**Dept. Of Electrical and Electronics Engineering**

**NIT NAGALAND**

**Dimapur- 797103**

**Mrs. A. PRASANNA LAKSHMI**

Project Coordinator

**Asst. Professor**

**Dept. Of Electronics and Instrumentation Engineering**

**NIT NAGALAND**

**Dimapur- 797103**



**Dr. R. KUMAR**

**Head of the Department**

**Professor**

**Dept. Of Electronics and Instrumentation Engineering**

**NIT NAGALAND**

**Dimapur- 797103**

**Dr. M. KANNAN**

**External Examiner**

**Professor**

**Dept. Of Electronics**

**Engineering, MIT Campus**

**Anna University**

**Chennai, TN**

#### **ACKNOWLEDGEMENT**

I wish to place on record my deep sense of gratitude to   
my honorific Guide **Dr. D. Ganga**, Assistant professor, National Institute of Technology Nagaland for his / her supervision, valuable guidance and moral support leading to the successful completion of the work. Without his / her continuous encouragement and involvement, this project would not have been a reality.

I would like to extent my sincere thanks to **Dr. R. Kumar**, Head of the Department of Electronics and Instrumentation Engineering, NIT Nagaland for continuous support. I would also like to thank all my friends who have developed me to gain a sense of dutifulness, perfection and sincerity in the effort.

Last but not least, I would like to owe my sincere and incessant gratitude to my parent for rendering their full support and continuous love and care during the course of this work, and the almighty God for the immense blessing on me.

**ASHISH MISHRA**

2017107008

Dept of EIE

NIT Nagaland

**ABSTRACT**

Now a day’s water crisis has become a very big problem and due to limited availability of water resources it has become more challenging to supply sufficient water to all. This is why water management has become an essential need now-a-days. Moreover water prediction is also a basic necessity besides all this.

In our project work we are making water predictions for the real time data collected through the reservoir management systems. This can help us to solve the problem of water scarcity and keep a check over our water resources.

Using machine learning algorithms we can easily predict the values for a certain span

Here we are using time series analysis under the machine learning for solving our problem statement. Under time series analysis, we are using ARIMA model to make predictions.

Using ARIMA model we are able to easily predict the water consumption for up to two years.

**CONTENTS**

**LIST OF CONTENTS PAGE NO.**

**ABSTRACT**

**CHAPTER 1**

**INTRODUCTION**

1.1: PROBLEM STATEMENT 08

1.2: NEED OF SMART RESERVOIR MANAGEMENT SYSTEM 10

1.3: SMART WATER MANAGEMENT SYSTEM 11

1.4: OBJECTIVE 11

**CHAPTER 2**

**TECHNOLOGY & SOFTWARE PACKAGE**

2.1: MACHINE LEARNING 12

2.2: JUPYTER NOTEBOOK 13

2.3: CSV FILES 13

2.4: LANGUAGE: PYTHON 14

2.5: LIBRARY USED 15

**CHAPTER 3**

**TIME SERIES ANALYSIS**

3.1: WHY TIME SERIES 16

3.2: TIME SERIES ANALYSIS 16

3.3: COMPONENT OF TIME SERIES 16

3.4 LIMITATIONS 17

3.5 STATIONARITY 18

3.6 ARIMA MODEL 19

**CHAPTER 4**

**SOURCE CODE, GRAPHS AND WATER PREDICTIONS**

4.1 DATA PRE-PROCESSING 21

4.2 TESTING THE DATASET 25

4.3ESTIMATING TRENDS 28

4.4LOG DIFFERENCING METHOD 29

4.5DECOMPOSING TIME SERIES INTO ITS COMPONENTS 33

4.6 PLOTTING THE ACF AND PACF GRAPHS 37

4.7 BUILDING THE ARIMA MODEL 39

4.8 PREDICTION 42

4.9 PREDICTED VALUES IN A LOGARITHMIC FORM INSIDE AN ARRAY 44

**CHAPTER 5**

**RESULT AND CONCLUSION**

5.1: RESULT 48

5.2: CONCLUSION 48

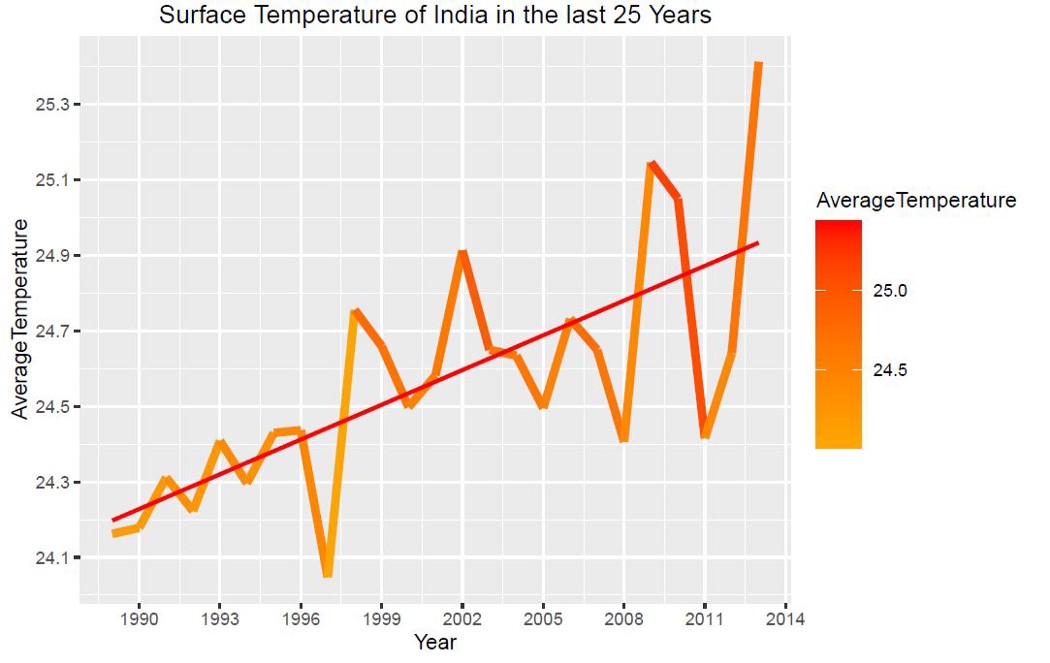
**CHAPTER-1**

**INTRODUCTION**

**1.1: PROBLEM STATEMENT**

India today as a population of over 1.3 billion people making it second largest country before china in the next 5 year india is expected to surpass china to become the most populous county in the world with this enormous population there’s obviously a huge demand for an essential resource water. India is now facing huge water crisis due to number of pressing factor around 40 percent of india population which amounts to close to 600 million people are predicted to have no access to drinking water by the year 2030 if changes are not made to the country’s water management. To India’s water problem is the fact that groundwater in india is running out fast 40 percent of country’s drinking water supply comes from groundwater which was predicted to run out for 21 major Indian cities by this year.

India's average temperature is constantly increasing, due to which water reservoirs are constantly drying up.

****

Another source of water for the country is through surface water things like reservoirs lakes and rivers however many of these water bodies have been dying up in many regions of the country due to more frequent heat waves and more delayed monsoon periods for example: Chennai’s largest water reservoir Red Hill Water Reservoir is continuously drying up due to which there is going to be a huge water shortage in the in Chennai.



Example-There was not a single drop of water in 3100 villages out of the total 8522 villages of Marathwada in Maharastra. To meet the shortage of water in Latur, 500000 liters of water was being sent to Latur every third day from Miraj place, located 342 km from Latur.

****

To meet the shortage of water in Latur, water was being transported from various parts of Maharashtra and India by tank train to Latur.

****

We can not produce water so we need to conserve water resources. We need to build proper infrastructure for Reservoir management system & water distribution system. we need systems which should be smart & by using technology like Machine Learning, Deep Learning, Artificial Intelligence , Internet Of Things we can predict scarcity of water & we can convert whole manual system into smart & fully automatic system.

**1.2:NEED OF SMART RESERVOIR MANAGEMENT SYSTEM:**

Sometimes there may be an excess  of water at some place and simultaneously there may be water shortage at some other place. This reservoir management system will help to overcome this problem. Using the machine learning algorithm, we can estimate the drought at specific places and accordingly do the arrangements beforehand.

High-technology solutions for the water sector include digital meters and sensors, supervisory control and data acquisition systems, as well as geographic information systems (GIS).Smart technology can change conventional water and wastewater systems into instrumented, interconnected, and intelligent systems.

1-Instrumented: the ability to detect, sense, measure, and record data.

2-Interconnected: the ability to communicate and interact with system operators and managers.

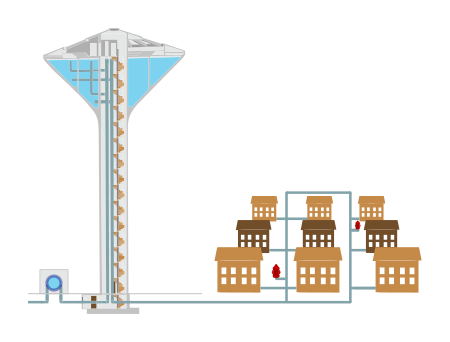
3-Intelligent: the ability to analyse the situation, enable quick responses, and optimize troubleshooting solutions.

Some of the main advantages of smart reservoir management systems are a better understanding of the water system, detection of leaks, conservation, and monitoring of water quality. The implementation of smart water system technologies enables public services companies to build a complete database for the identification of the areas where water losses or illegal connections occur. Other advantages of smart reservoir management systems are economic benefits to water and energy conservation, while the efficiency of the system can improve customer service. The wireless datatransmission allows the customers to analyse their water consumption towards preserving and reducing the water bill as well as keeping a track of their water requirements.

**1.3: SMART WATER MANAGEMENT SYSTEM**

Smart reservoir management systems can provide a more resilient and efficient water supply system, reducing costs and improving sustainability. Smart technology can change conventional water management systems into  an instrumented and intelligent systems. Moreover, acquisition and analysis of the usage data can result in a better forecasting of the future demands as well as will help us to optimize the water usage in particular sectors where the water demand is high.

By Reservoir Management System we mean that there may be a single reservoir or a collection of multiple tanks in such. For multiple tanks all these tanks together will send their data to a common node and all these tanks together will form a Reservoir Management System.

****

**1.4: OBJECTIVE**

In phase-1 we are going to predict water consumption using machine learning algorithm. Using this prediction we can forecast scarcity of water in many areas. We will analyze the data and then after we can say which area is consuming more water and we can take several decisions. In the first phase we have thought of working on the Machine Learning Algorithms for analyzing and predicting the future forecast of the water consumption and supply needs. Using this we will easily be able to put forward the statistics of water consumption by various sectors and also predict a future forecast for them.

We have one variable that is quantity of water in gallons. We need to build a forecast to determine how much gallons of water are going to consumption at the month level in the future. So here we have real month and date from 2012 to 2018 and we have the quantity of water in gallons. So now we have this kind of data and we need to analyse what will be the th quantity of water if we have to do it for next 2 years.

**CHAPTER-2**

**TECHNOLOGY & SOFTWARE PACKAGE**

**2.1: MACHINE LEARNING**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs** that can access data and use it to learn for themselves.

The process of learning begins with observations or data, such as examples, direct experience, or instruction, in order to look for patterns in data and make better decisions in the future based on the examples that we provide. **The primary aim is to allow the computers learnautomatically** without human intervention or assistance and adjust actions accordingly.

But, using the classic algorithms of machine learning, text is considered as a sequence of keywords; instead, **an approach based on semantic analysis mimics the human ability to understand the meaning of a text.**

Machine learning algorithms are often categorized as supervised or unsupervised.

**(A) Supervised machine learning algorithms** can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system is able to provide targets for any new input after sufficient training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

**(B)Unsupervised machine learning algorithms** are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

**(C)Semi-supervised machine learning algorithms** fall somewhere in between supervised and unsupervised learning, since they use both labeled and unlabeled data for training – typically a small amount of labeled data and a large amount of unlabeled data. The systems that use this method are able to considerably improve learning accuracy. Usually, semi-supervised learning is chosen when the acquired labeled data requires skilled and relevant resources in order to train it / learn from it. Otherwise, acquiring unlabeled data generally doesn’t require additional resources.

**(D)Reinforcement machine learning algorithms** is a learning method that interacts with its environment by producing actions and discovers errors or rewards. Trial and error search and delayed reward are the most relevant characteristics of reinforcement learning. This method allows machines and software agents to automatically determine the ideal behavior within a specific context in order to maximize its performance. Simple reward feedback is required for the agent to learn which action is best; this is known as the reinforcement signal.

Machine learning enables analysis of massive quantities of data. While it generally delivers faster, more accurate results in order to identify profitable opportunities or dangerous risks, it may also require additional time and resources to train it properly. Combining machine learning with AI and cognitive technologies can make it even more effective in processing large volumes of information.

**2.2:JUPYTER NOTEBOOK**

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and explanatory text. Uses include: data cleaning and transformation, numerical simulation, statistical modelling, machine learning and much more.JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plug-in that add new components and integrate with existing ones.

#### **2.3: CSV FILES**

CSV stands for Comma Separated Values. A CSV file is a plain text file that stores tables and spreadsheet information. The contents are often a table of text, numbers, or dates. CSV files can be easily imported and exported using programs that store data in tables.

These files serve a number of different business purposes. They help companies export a high volume of data to a more concentrated database, for instance.

They also serve two other primary business functions:

1-CSV files are plain-text files, making them easier for the website developer to create.

2-Since they're plain text, they're easier to import into a spreadsheet or another storage database, regardless of the specific software you're using.

3-To better organize large amounts of data.

**2.4: LANGUAGE: PYTHON**

Python is a high-level programming language designed to be easy to read and simple to implement. It is [open source](https://techterms.com/definition/opensource), which means it is free to use, even for commercial applications. Python can run on Mac, Windows, and Unix systems and has also been ported to [Java](https://techterms.com/definition/java) and .NET virtual machines. Python is considered a scripting language, like [Ruby](https://techterms.com/definition/ruby) or [Perl](https://techterms.com/definition/perl) and is often used for creating Web [applications](https://techterms.com/definition/application) and dynamic Web content. It is also supported by a number of 2D and 3D imaging programs, enabling users to create custom [plug-ins](https://techterms.com/definition/plugin) and extensions with Python. Examples of applicatio

Python was created in 1991 by Dutch programmer Guido Van Rossum. It is an interpreted language. This means that it has an interpreter to execute the programme directly, as opposed to depending more complicated machine languages. In fact, Van Rossum wants Python to eventually as understandable and clear as plain English. He has also made the language open source, which means that anyone can contribute to it, and he hopes that it will become as powerful as competing languages.“Readability” is a key factor in Python’s philosophy. As such, it aims to limit code blocks (blocks of source code text) and have white space instead, for a clearer, less busy appearance.

Python is used in virtually every industry and scientific field that you can imagine, including

1-Data Science.

2-Machine Learning.

3-Web Development.

4-Computer Science Education.

5-Computer Vision and Image Processing.

6-Game Development.

7-Medicine and Pharmacology.

8-Biology and Bioinformatics.

9-Neuroscience and Psychology.

10-Astronomy.

11-Other areas such as robotics, autonomous vehicles, business, meteorology, and graphical user interface (GUI) development.

2.5LIBRARY USED

1-NUMPY

2-PANDAS

3-STATSMODELS

4-MATPLOTLIB

**1-NUMPY LIBRARY**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding.

**2-PANDAS LIBRARY**

Pandas is an open-source, BSD-licensed Python library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc. In this tutorial, we will learn the various features of Python Pandas and how to use them in practice.

**3-STATSMODELS**

As its name implies, statsmodels is a Python library built specifically for statistics. Statsmodels is built on top of Numpy, Scipy , and matplotlib, but it contains more advanced functions for statistical testing and modeling that you won't find in numerical libraries like NumPy or SciPy.

**4-MATPLOTLIB**

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

**CHAPTER-3**

**TIME SERIES ANALYSIS**

**3.1: WHY TIME SERIES**

In time series analysis we have just one variable that is time. Under supervised learning we have linear regression, in linear regression we have an independent variable and we have a dependent variable. We do mapping of function that how one variable is related to another and then we can go ahead with analysis part but in time series analysis we just have one variable that is time. For this project we need only one variable that is quantity of water in gallons. So we are using time series analysis.

**3.2: TIME SERIES ANALYSIS**

Time series is a set of observation taken at specified times usually at equal intervals. On x- axis we have a time the y-axis the magnitude of the data. For this project we are taking date on x-axis and quantity of water (total gallons) in y-axis.

We can apply time series analysis in many fields for example

1- Business forecast

2-understand past behaviour

3- Plan future

4- Evaluate current accomplishment

Many times past data decides what is going to happen in future. When some festival is there and we are selling chocolates then sales will increase during a festival so we need to think about the seasonality. Time series helps us to plan the future operations so we can analyse the past and then we can forecast your future using this algorithm that is time series analysis.

**3.3: COMPONENT OF TIME SERIES**

Time series have

1- Trend

2- Seasonality

3- Irregularity

4- Cyclic pattern

**1-Trend**

Trend is basically a movement of relative higher or lower values over a long period of time. So when the time series analysis shows a General pattern that is up then we call all it and uptrend, and if trend exhibits a lower pattern that is down trend. If there was no trend we call it as a horizontal Trend or stationary Trend. For example: there is a new township that has been constructed and people are going to come and live over there. A guy comes up and he opens up a hardware shop there so people will be coming up and they will buy stuff from there. Now once all these houses are settled up or it is been occupied then need of hardware reduces so the trend may go down so let's say the sales were up in the first year and it goes down in next 6 month. It is a trend. In trend for some amount of time selling was high then it got down but this should not be a pattern. Trend is something that happens for some time and then disappear.

**2- Seasonality**

Seasonality is basically upward or downward trend but it should be a repeating pattern within a fixed time period. For example: Christmas happens every year on 25 December and we have business of chocolates. So we observe that every year on year chocolates are served more and more in the last week of December. This is because of Christmas. We are seeing this across the year that is from past two years or four years or six years or 10 years and so on. So it is a repeating pattern within a fixed time period.

**3- Irregularity**

It is also called as noise. These are erratic in nature or we can say unsystematic. It is also called as Residual so this happens basically for short duration and it is non repeating.

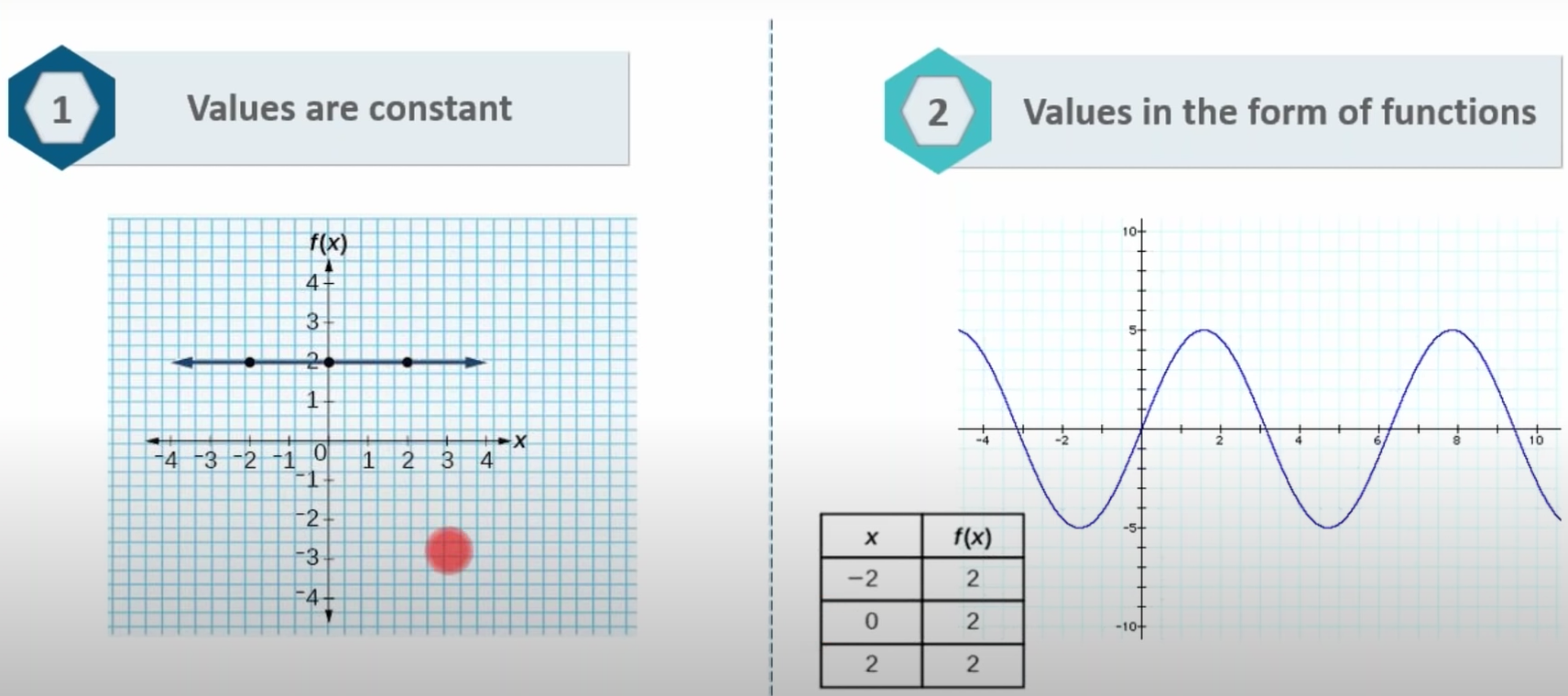
Example- There is a flood in your town out of nowhere in one year now a lot of people are buying medicines and Boat for relief but after sometime when everything is settled up then sales of those boat have gone down so this is something that no one could have predicted it's going to happen erratically. We don't know how much selling is going to happen.

**4- Cyclic**

Cyclic is basically repeating up and down movements you can go over more than a year so they don't have a fixed pattern so they can happen anytime in two years then 4th year then maybe in 6 month. So they keep on Repeating and they are much harder to predict.

**3.4 LIMITATIONS**

We cannot apply time series analysis when the values are constant. If we have values in form of functions like sinx or cosx, here also we cannot apply time series analysis.we can apply time series here also but there is no meaning to apply time series here because we can easily protect the value using the function.

****

**3.5 STATIONARITY**

Time series required data to be stationarity so any type of statistical model that we will apply on time series that data should be stationary. If time series has a particular behaviour over time there is a very high probability that it will follow the same in the future.

Stationarity have following criteria

1- Constant mean - mean should be constant according to the time

2- Constant variance - Variance should be equal on equal time interval

3- Auto covariance - It does not depends on a time

Mean is basically a average. Variance is just the distance from the mean. So each point distance from the mean should be equal. Auto covariance that should not depend on time or it should be equal. For example: We are standing at time t, and previous time period are T-1 or T-2. T-1, T-2, T should not have any kind of correlation between them which is basically dependent on your time. That is auto covariance.

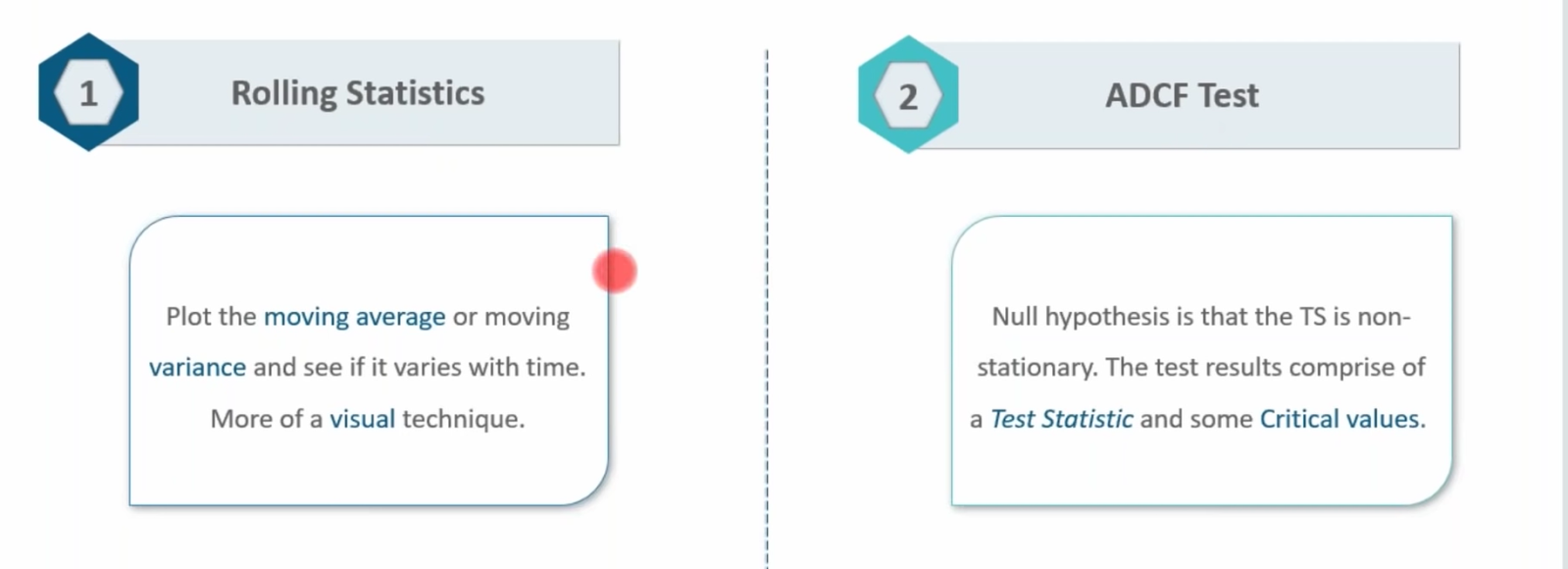
When above three conditions are met then we can say that series is stationary and then we can apply time series analysis. To check stationary in Python we have to popular test first is rolling statistics and second is ADCF(augmented dickey fuller) test.

**Rolling Statistics**

In rolling statistics we can plot the moving average and see if it varies with time. By moving average or moving variance our mean is that at any instance T take the average or variance of a time window. We cannot deploy this type of of stuff on production it is quite useful for POC purpose.

**Dickey fuller Test**

It is another statistical test for checking stationarity. Here we have the null hypothesis. Which time series is non stationary and we perform this test we will get a result which comprises of a test statistic and some critical values for different confidence level. If the test statistic is less than some critical value we can reject the null hypothesis and say the series is stationary.

****

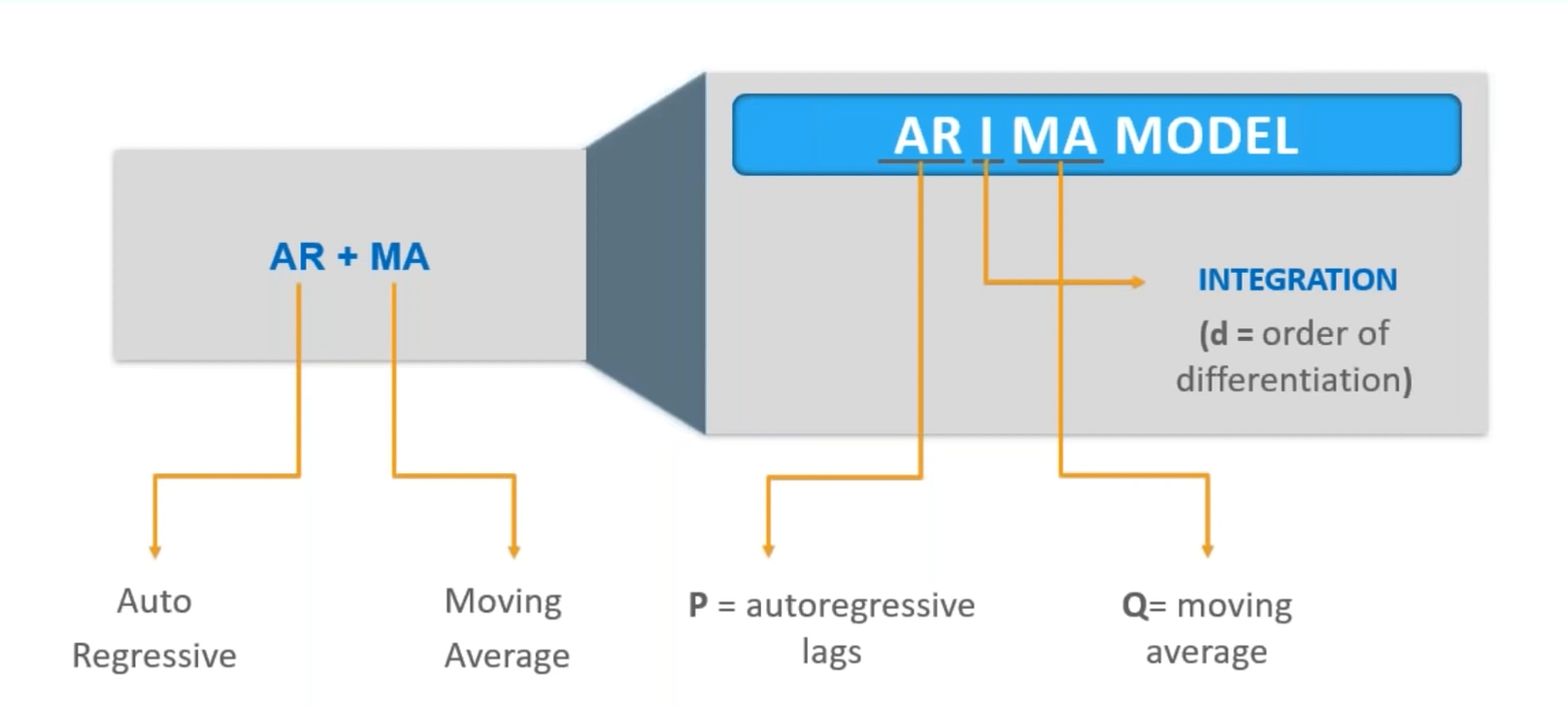
**3.6 ARIMA MODEL**

It is one of the best models to work with time series data. So this is basically the combination of two models that is

AR + MA

After combining these two models we will get the ARIMA model. AR Model stands for auto regressive. MA model stands for or moving average. AR is the separate model and MA is the separate model and integration binds it together. Integration part is indicated by I.

AR is nothing but the correlation between the previous time period to the current. Lets we are standing at the time period T and there are previous time periods like T-1 and T-2 and T-3. And if we found any correlation between T-3 and T that is nothing but the autoregressive part. There is always some kind of noise or irregularity attached in a time series so we need to figure out that noise in fact we need to average that out. Now whenever we try to average it out the cross and drop set of present in that noise smoothen out and we can have average forecast of that noise.

****

ARIMA model has three parameters it has P Q and D. P basically refers to your auto regressive lags. Q stands for moving average. D stands for order of differentiation.

We have each parameter for each of the models, If we take the integration by just one order then the value of D would be one if we differentiate it in the order of 2 to then we have the value OF D equals to 2. If we want to predict the value of P then we will be using PS EF graph that is partial autocorrelation graph. To predict value of q we need to plot ACF plot that is autocorrelation plot. Order of differentiation defines the value of D.

**CHAPTER 4**

**SOURCE CODE, GRAPHS AND WATER PREDICTIONS**

**4.1: DATA PRE-PROCESSING**

Before analyzing and working on any dataset we need to sort our dataset as per our needs as well as filter the unwanted data and attributes, this is called data pre -processing. Data pre-processing must essentially be performed in the very beginning.

## 4.1 .1: IMPORT THE RELEVANT LIBRARIES

import pandas as pd

import numpy as np

import matplotlib.pylab as plt

%matplotlib inline

from matplotlib.pylab import rcParams

rcParams["figure.figsize"] = [6.4, 4.8]

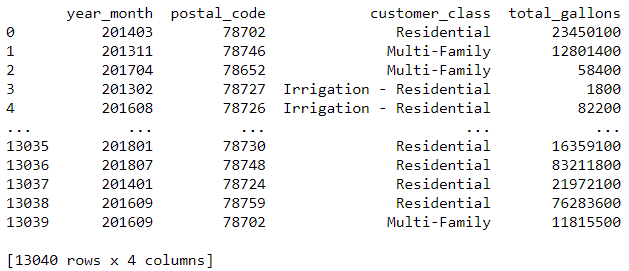
from datetime import datetime

## 4.1.2 :PRINTING THE ORIGINAL DATASET

data = pd.read\_csv(r"C:/Users/Piyus/Desktop/Review 2/austin\_water\_residential\_water\_consumption\_1.csv")

data.shape

print(data)



**4.1 .3: SETTING THE DATES TO STANDARD DATE STRING FORMAT:**

def timedate(x):

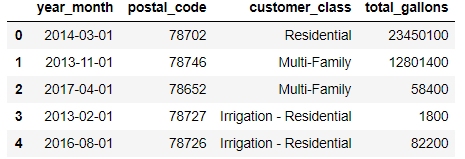
x = str(x)

x = x[0:4]+ str('-') + x[4:]+ str('-') + str('01')

return x

data['year\_month'] = data['year\_month'].apply(lambda x: timedate(x))

data.head()



For separating the continuous date format in our dataset we use Lambda function.

Using the Lambda function we select the whole year\_month column as a string format.

We further use an iterative loop which separates the year, month and day with a dash.

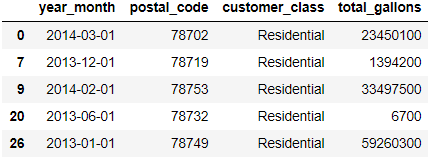
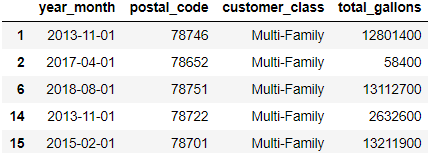
**4.1 .4 SEGREGATING THE DATASET AND DROPPING THE IRRELEVANT COLUMNS:**

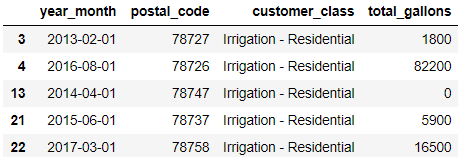
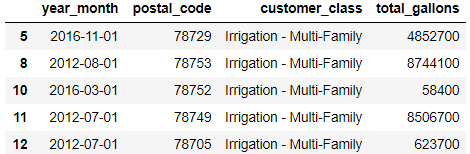
Residential = data[data["customer\_class"] == "Residential"]

d\_res = Residential.copy(deep = True)

d\_res.head()

# Similarly we can write the code for segregating all the for attributes

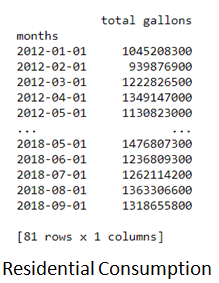
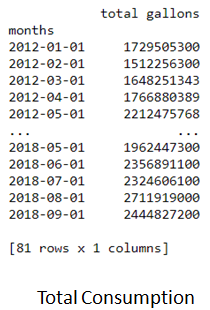


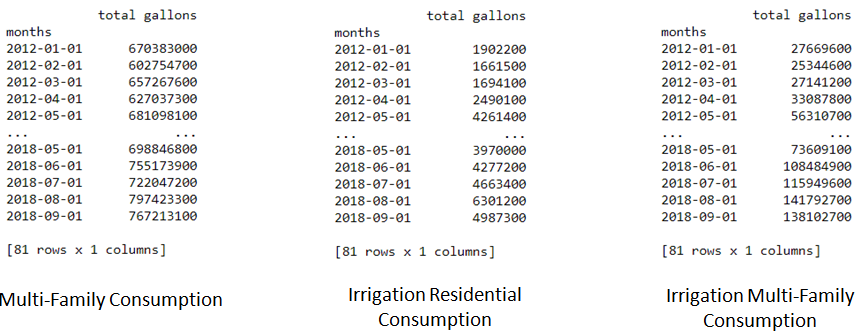


**4.1 .5: DROPPING OUT THE IRRELEVANT COLUMNS:**

d\_res = d\_res.drop(['postal\_code', 'customer\_class'], axis = 1)

d\_res.style.hide\_index()





**4.1 .6 PARSING THE DATES TO STANDARD DATE TIME FORMAT AND PLOTTING THE GRAPHS:**

data = pd.read\_csv(r"C:/Users/Piyus/Desktop/Review 2/austin\_water\_residential\_water\_consumption\_1.csv")

data.shape

# parsing string to a date time type

data['months']=pd.to\_datetime(data['months'], infer\_datetime\_format=True)

data=data.set\_index(['months'])

print(data)

# We can write the same code for all the four attributes as well

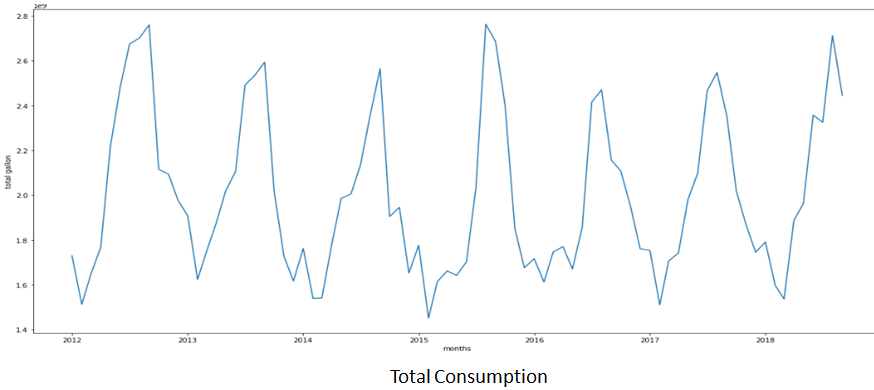
# Plotting Graph

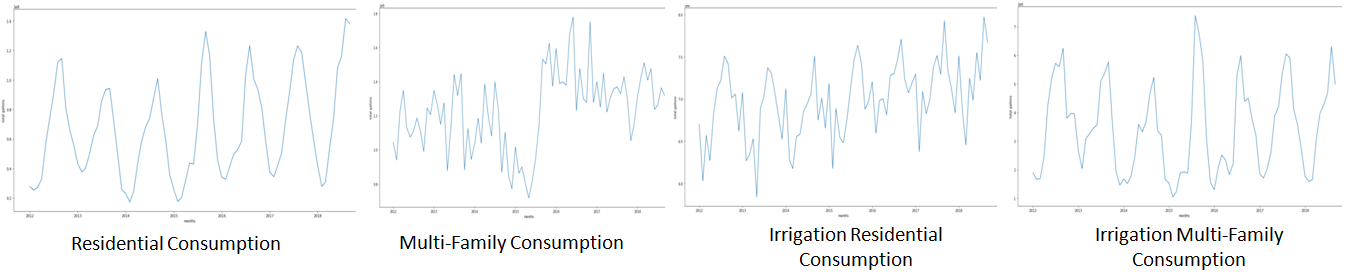
plt.figure(figsize=(20,10))

plt.xlabel("months")

plt.ylabel("total gallon")

plt.plot(data)





**4.2 TESTING THE DATASET**

* + 1. **ROLLING STATISTICS TEST**

Plotting the Rolling Mean and Standard Deviation, which has window of 12.

By looking below plot, we conclude that, it is very much stationary because mean and variance is almost constant .

# plotting rolling stastics

plt.figure(figsize=(20,10))

original=plt.plot(data, color='blue', label='Original')

mean\_6=plt.plot(rolmean, color='red', label='Rolling Mean')

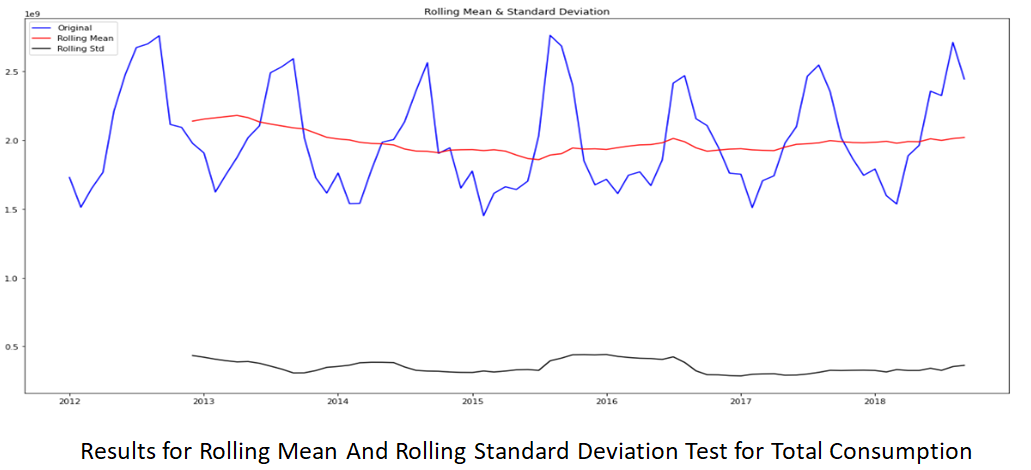
std\_6=plt.plot(rolstd, color='black', label='Rolling Std')

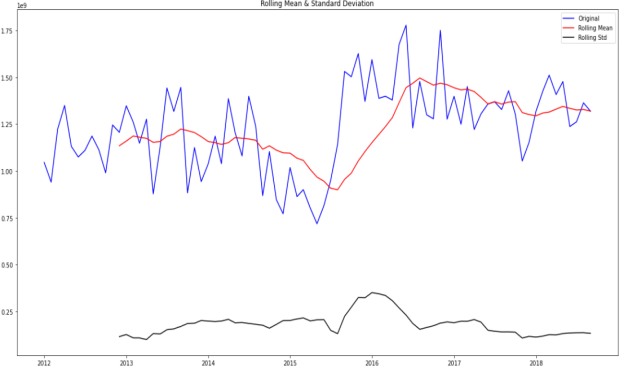
plt.legend(loc='best')

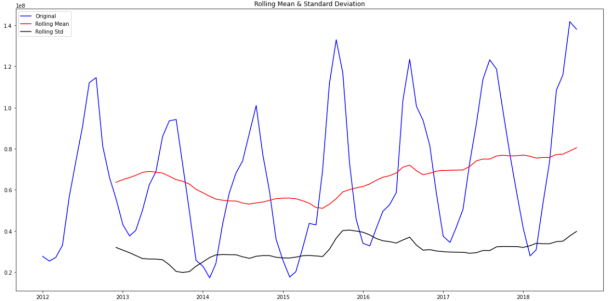
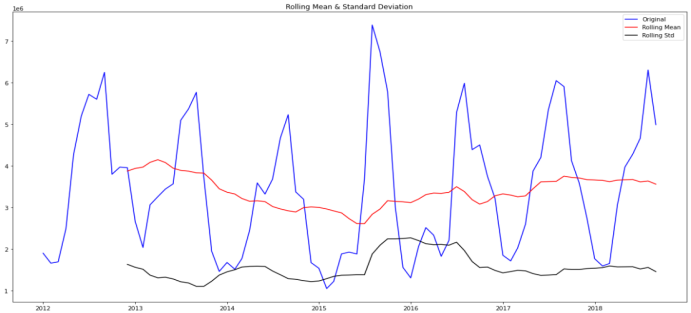
plt.title('Rolling Mean & Standard Deviation')

plt.show(block=False)

# We can write the same code for all the four attributes as well.







**4.2 .3 DICKEY FULLER TEST**

The Dickey–Fuller test tests the null hypothesis that a unit root is present in an autoregressive model.Here we can see the p value coming out for our original dataset is very small, so it is very close to a stationery dataset.

# Performing Dickey Fuller Test

from statsmodels.tsa.stattools import adfuller

print('Results For Dickey-Fuller Test: ')

dftest=adfuller(data['total gallons'], autolag='AIC')

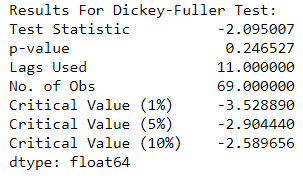
dfoutput=pd.Series(dftest[0:4], index=['Test Statistic','p-value','Lags Used','No. of Obs'])

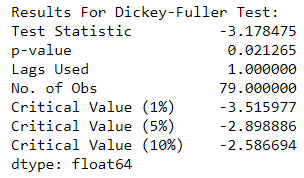
for key,value in dftest[4].items():

dfoutput['Critical Value (%s)'%key] = value

print(dfoutput)

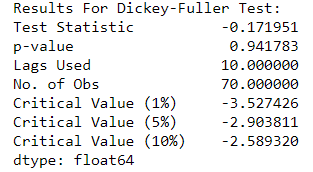
# We can write the same code for all the four attributes as well











**4.3:ESTIMATING TRENDS**

We took log transformation to make our Time series stationary and plotted visual for it, red line represents the average values. We found graph again to be looking similar to stationery. After this use another log method differencing, to make our time series stationary.

# Estimating Trends

plt.figure(figsize=(20,10))

data\_log=np.log(data)

plt.plot(data\_log)

plt.figure(figsize=(20,10))

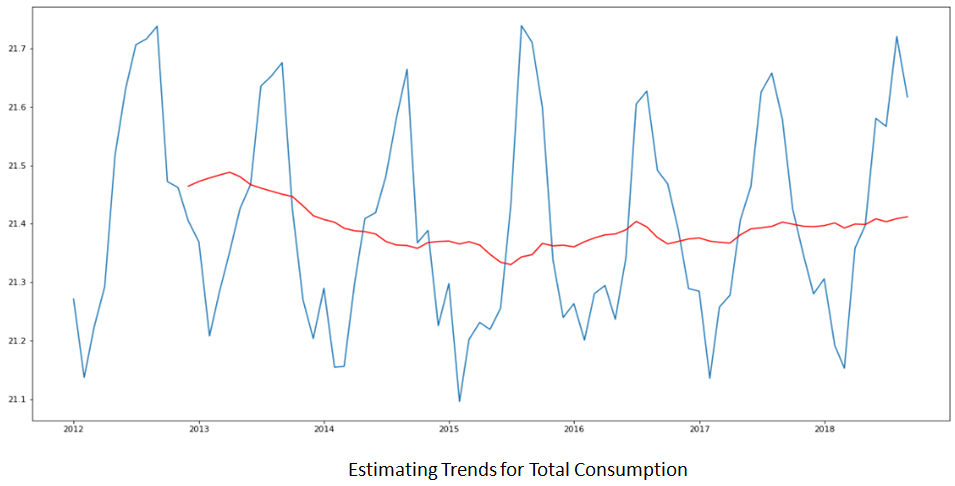
MAvg=data\_log.rolling(window=12).mean()

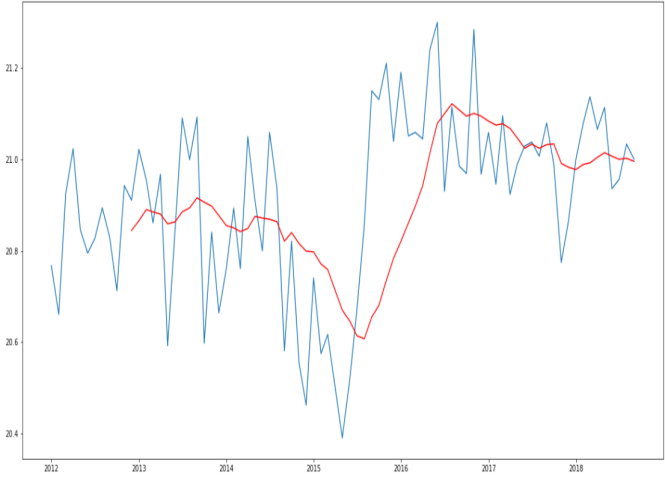
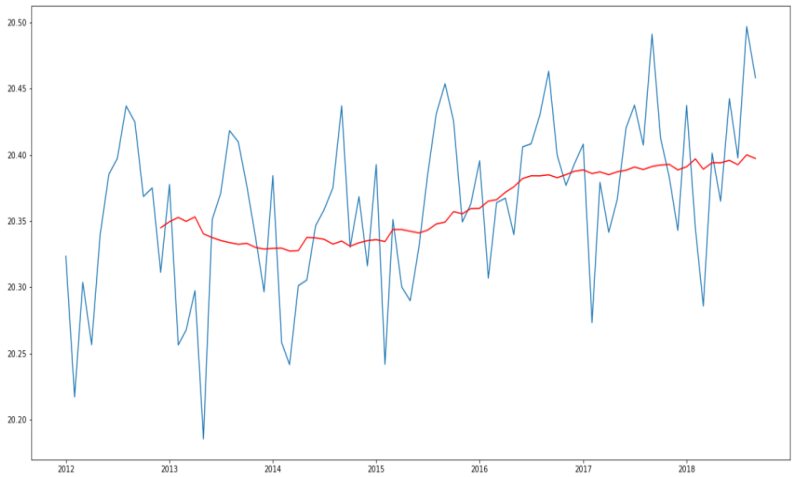
MStd=data\_log.rolling(window=12).std()

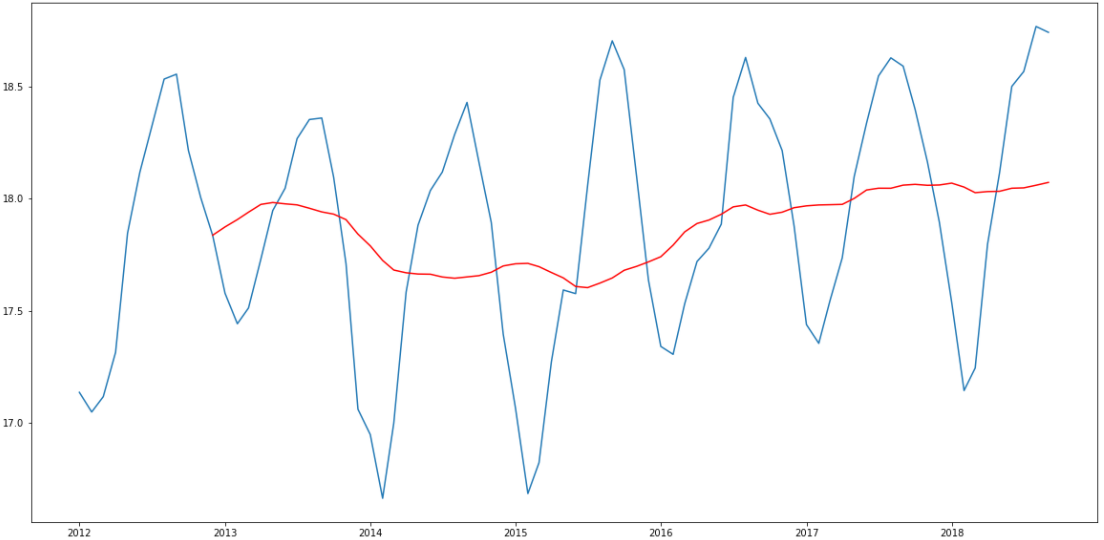
plt.plot(data\_log)

plt.plot(MAvg, color='red')

# We can write the same code for all the four attributes as well







**4.4LOG DIFFERENCING METHOD**

Using another log method differencing, to make our time series stationary. Dropping out the null values after applying the differencing method.

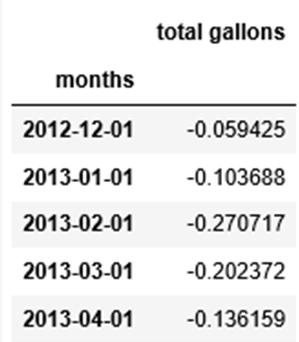
data\_log\_diff=data\_log-MAvg

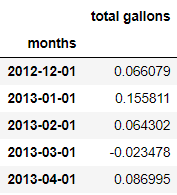
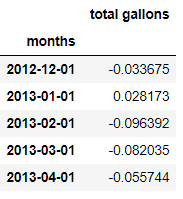
data\_log\_diff.head(12)

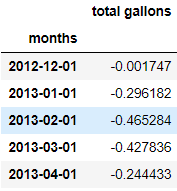
data\_log\_diff=data\_log\_diff.dropna()

data\_log\_diff.head()

# We can write the same code for all the four attributes as well.







# ADCF Test

def stationarity(timeseries):

rolmean=timeseries.rolling(window=12).mean()

rolstd=timeseries.rolling(window=12).std()

plt.figure(figsize=(20,10))

actual=plt.plot(timeseries, color='blue', label='original')

mean\_6=plt.plot(rolmean, color='red', label='Rolling Mean')

std\_6=plt.plot(rolstd, color='black', label='Rolling Std')

plt.legend(loc='best')

plt.title('Rolling Mean & Standard Deviation')

plt.show(block=False)

print('Dickey-Fuller Test: ')

dftest=adfuller(timeseries['total gallons'], autolag='AIC')

dfoutput=pd.Series(dftest[0:4], index=['Test Statistic','p-value','Lags Used','No. of Obs'])

for key,value in dftest[4].items():

dfoutput['Critical Value (%s)'%key] = value

print(dfoutput)

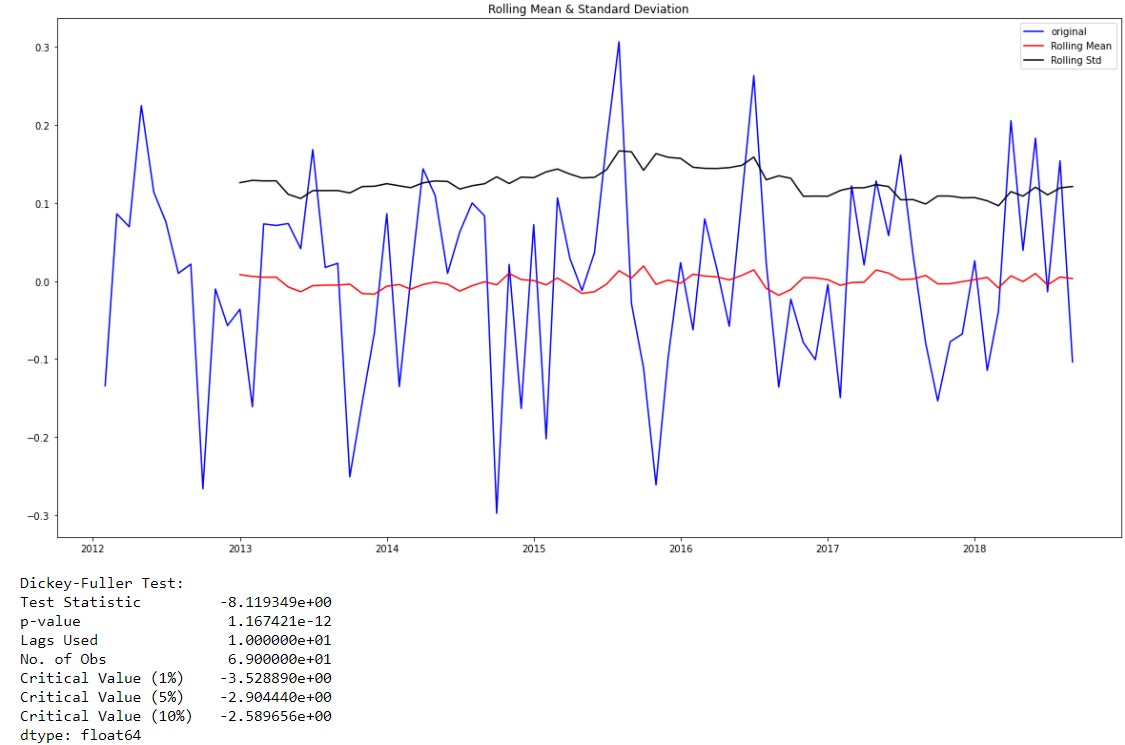
exp\_data\_diff=data\_log-exp\_data

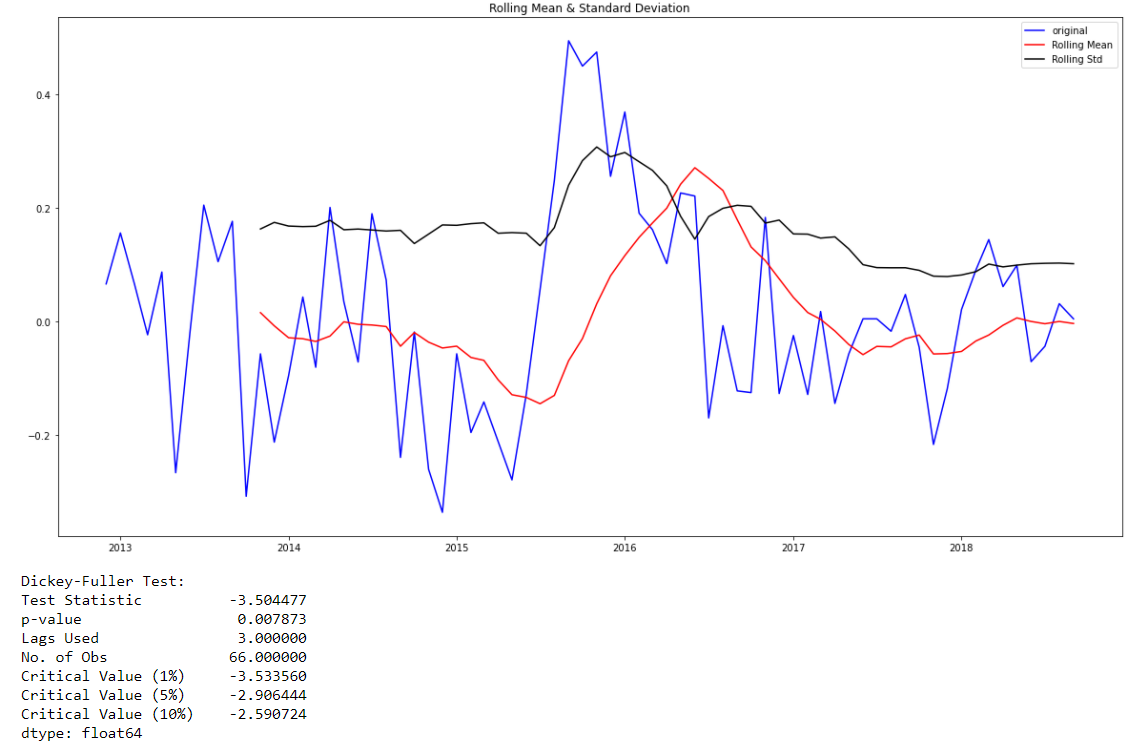
stationarity(exp\_data\_diff)

data\_shift=data\_log-data\_log.shift()

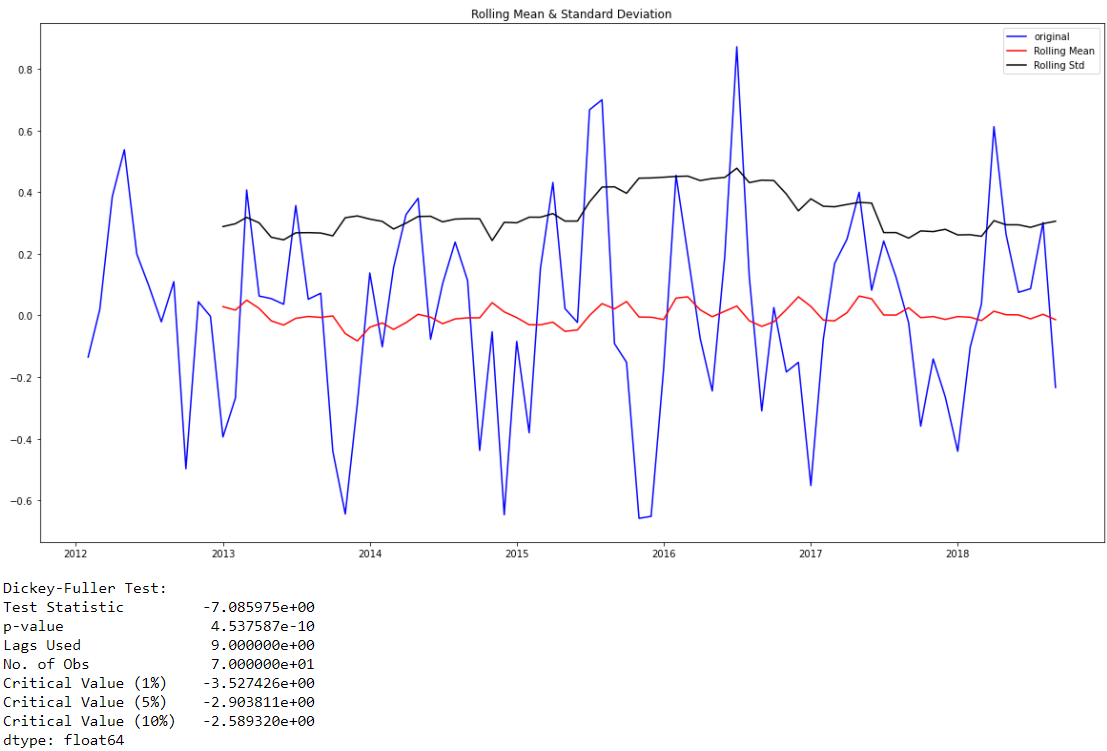
data\_shift=data\_shift.dropna()

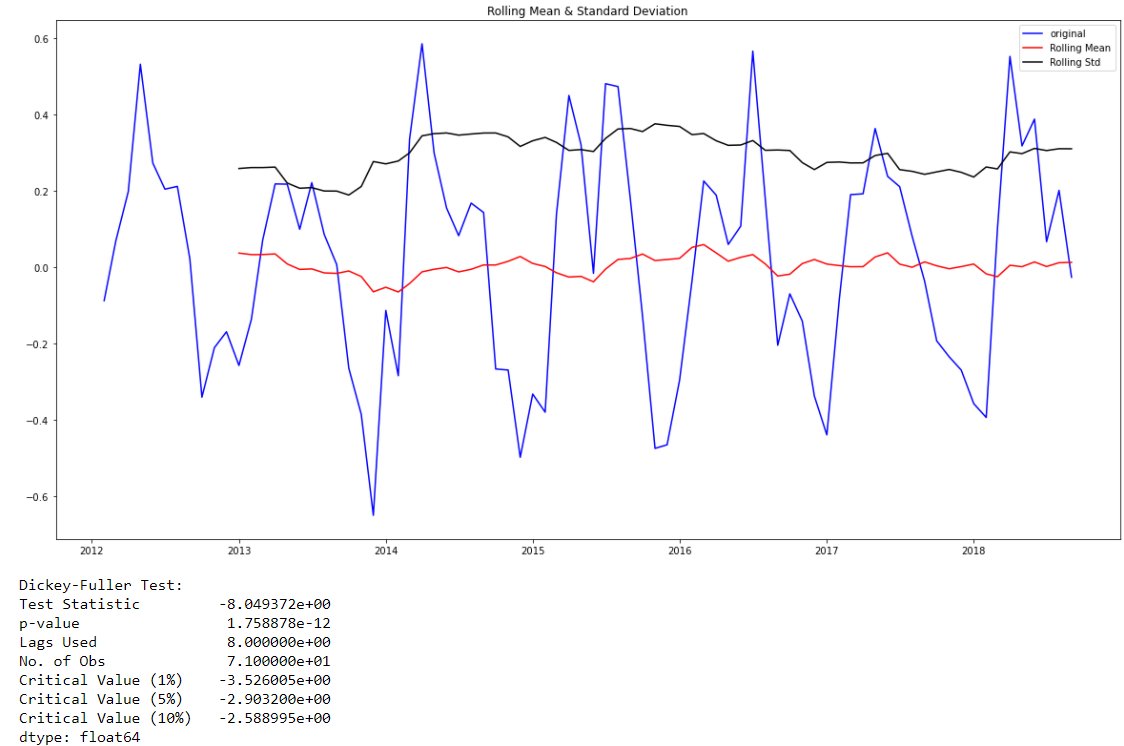
stationarity(data\_shift)











4**.5 DECOMPOSING TIME SERIES INTO ITS COMPONENTS**

We have decomposed time series into its three components i.e. trend, seasonality and residuals.We will again check the stationarity of these components as well.

from statsmodels.tsa.seasonal import seasonal\_decompose

decomp=seasonal\_decompose(data\_log)

trend=decomp.trend

seasonal=decomp.seasonal

residual=decomp.resid

plt.subplot(411)

plt.plot(data\_log, label='Original')

plt.legend(loc='best')

plt.subplot(412)

plt.plot(trend, label='Trend')

plt.legend(loc='best')

plt.subplot(413)

plt.plot(seasonal, label='Seasonality')

plt.legend(loc='best')

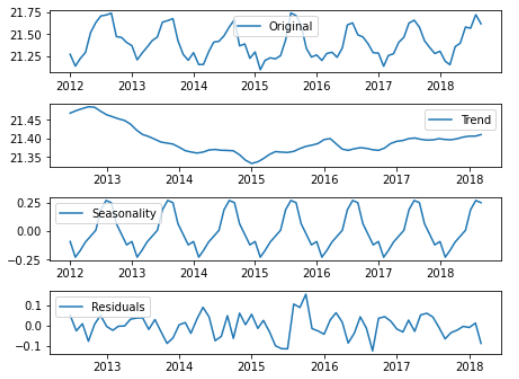
plt.subplot(414)

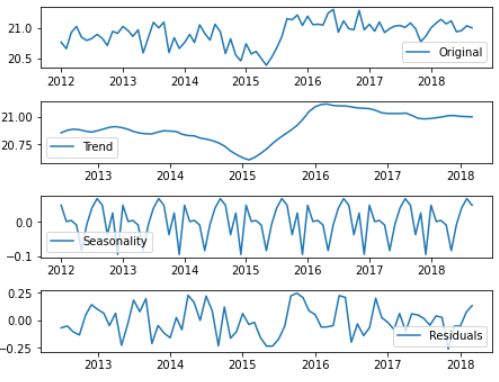
plt.plot(residual, label='Residuals')

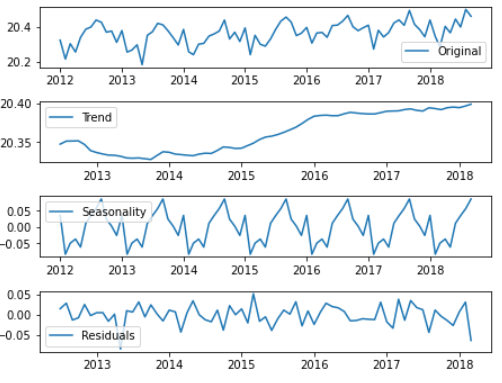
plt.legend(loc='best')

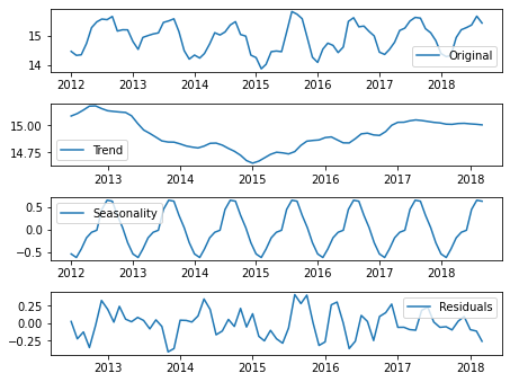
plt.tight\_layout()

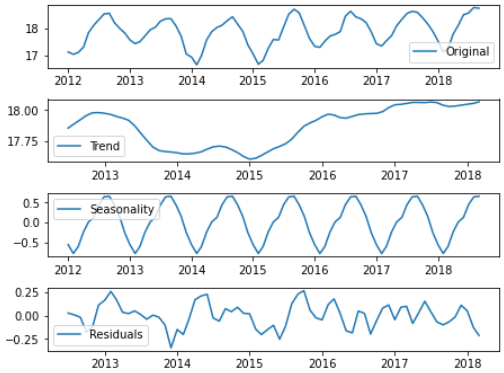
# We can write the same code for all the four attributes as well











4**.6 PLOTTING THE ACF AND PACF GRAPHS**

Plotting the ACF and PACF to find q and p value.We got q from Autocorrelation Function Graph.We got p from Partial Autocorrelation Function Graph.We’ll use these values for our ARIMA Model

# Plotting the ACF and PACF Graphs

from statsmodels.tsa.stattools import acf, pacf

lag\_acf=acf(data\_shift, nlags=20)

lag\_pacf=pacf(data\_shift, nlags=20, method='ols')

plt.figure(figsize=(20,10))

plt.subplot(121)

plt.plot(lag\_acf)

plt.axhline(y=0,linestyle='--',color='green')

plt.axhline(y=-1.96/np.sqrt(len(data\_shift)),linestyle='--',color='green')

plt.axhline(y=1.96/np.sqrt(len(data\_shift)),linestyle='--',color='green')

plt.title('Autocorrelation Function')

plt.subplot(122)

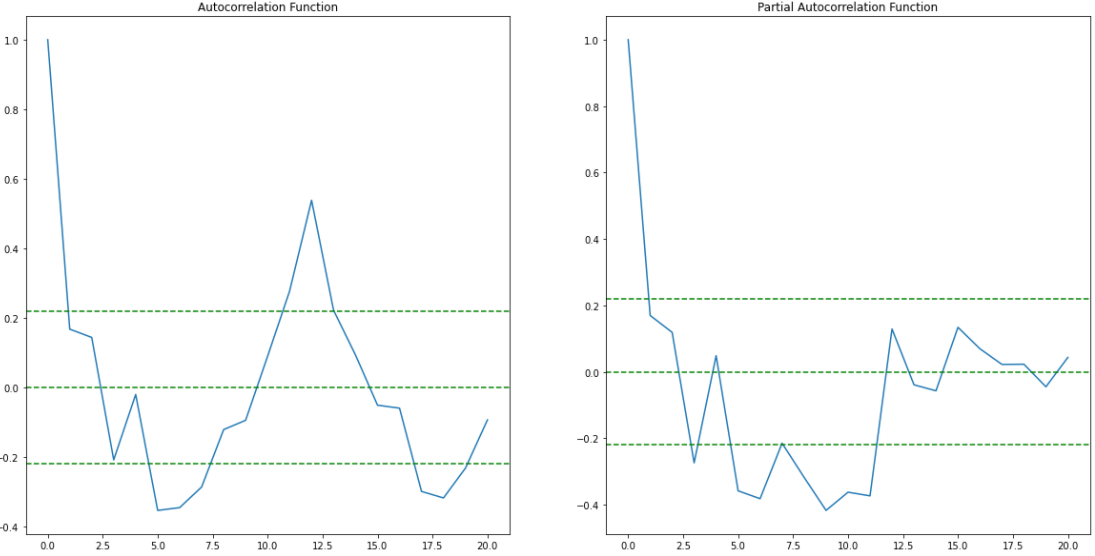
plt.plot(lag\_pacf)

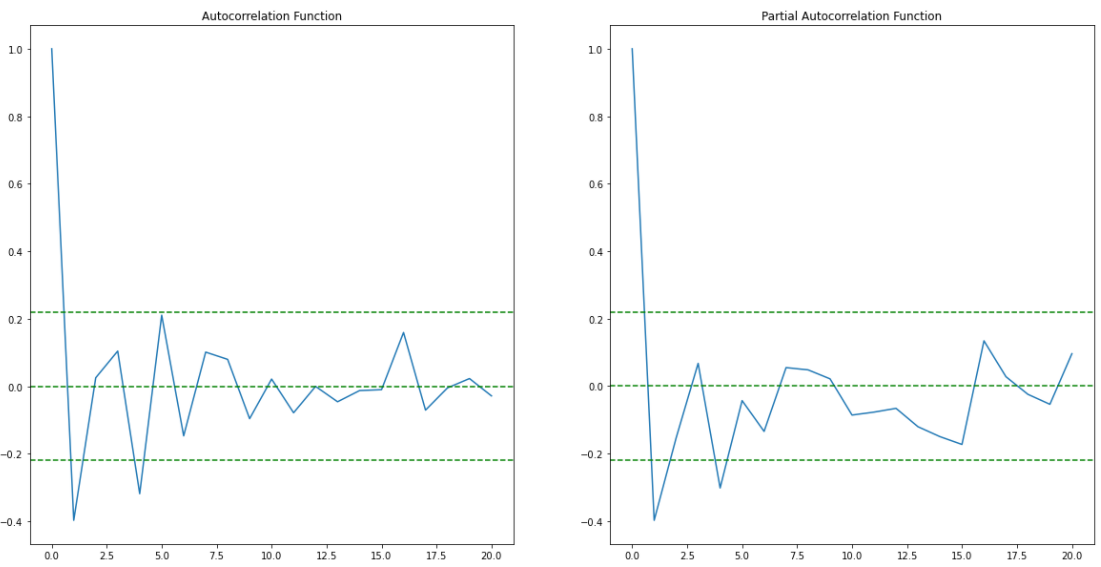
plt.axhline(y=0,linestyle='--',color='green')

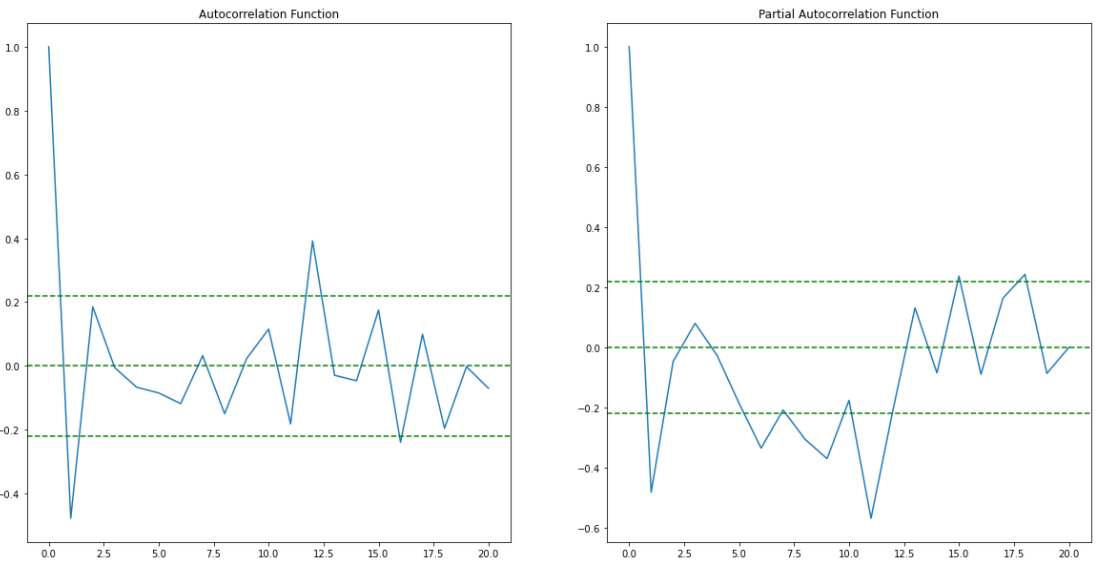
plt.axhline(y=-1.96/np.sqrt(len(data\_shift)),linestyle='--',color='green')

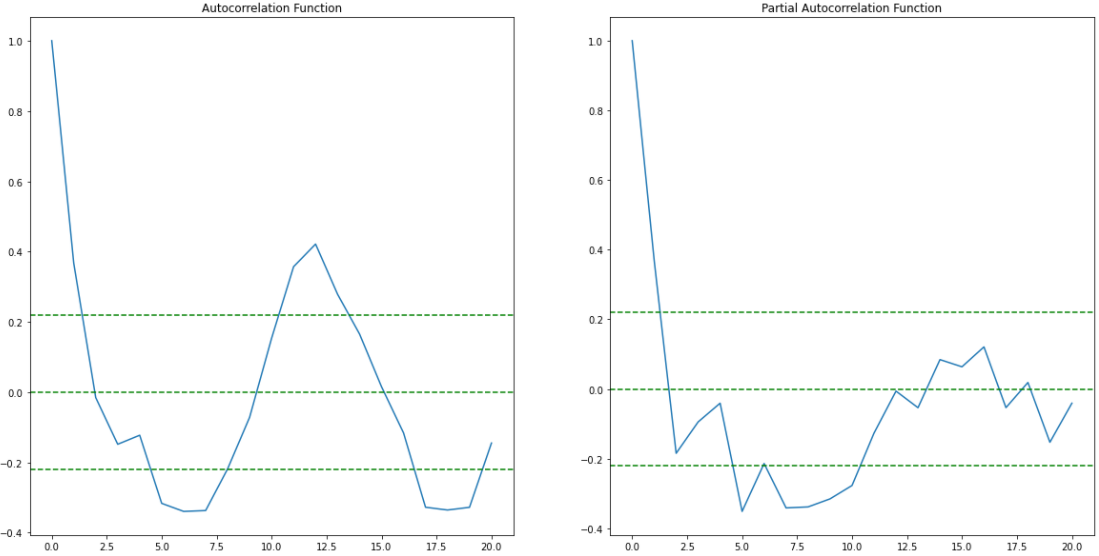
plt.axhline(y=1.96/np.sqrt(len(data\_shift)),linestyle='--',color='green')

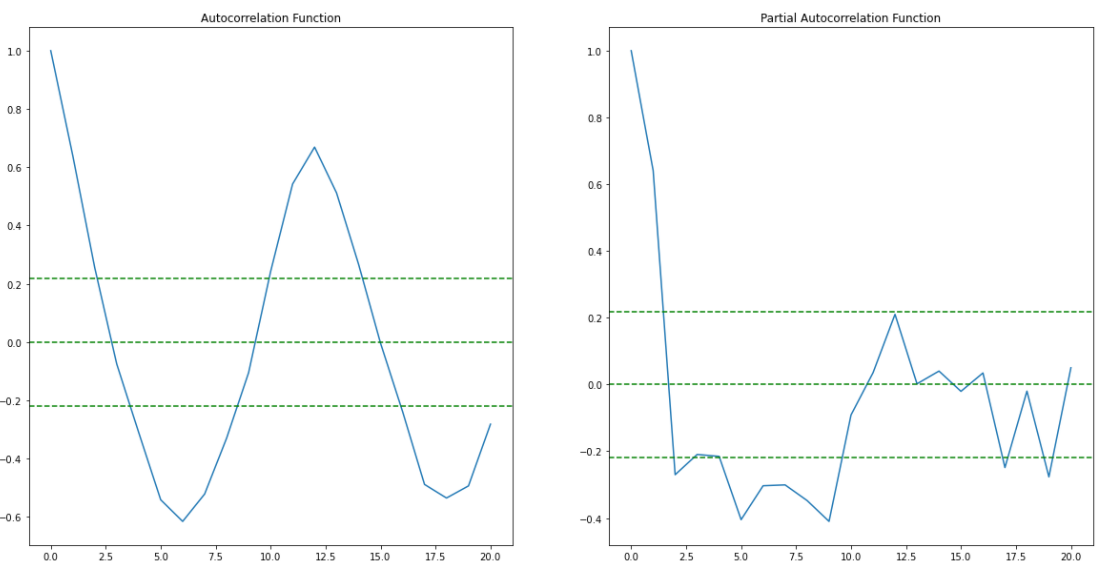
plt.title('Partial Autocorrelation Function')

# We can write the same code for all the four attributes as well









4**.7 BUILDING THE ARIMA MODEL**

Building the ARIMA model with p, q and d.Our RSS value i.e. the Residual Sum of Square comes out to be very low.Lower RSS values indicate a better model.

# Arima Model

from statsmodels.tsa.arima\_model import ARIMA

plt.figure(figsize=(20,10))

model=ARIMA(data\_log, order=(2,1,2))

results=model.fit(disp=-1)

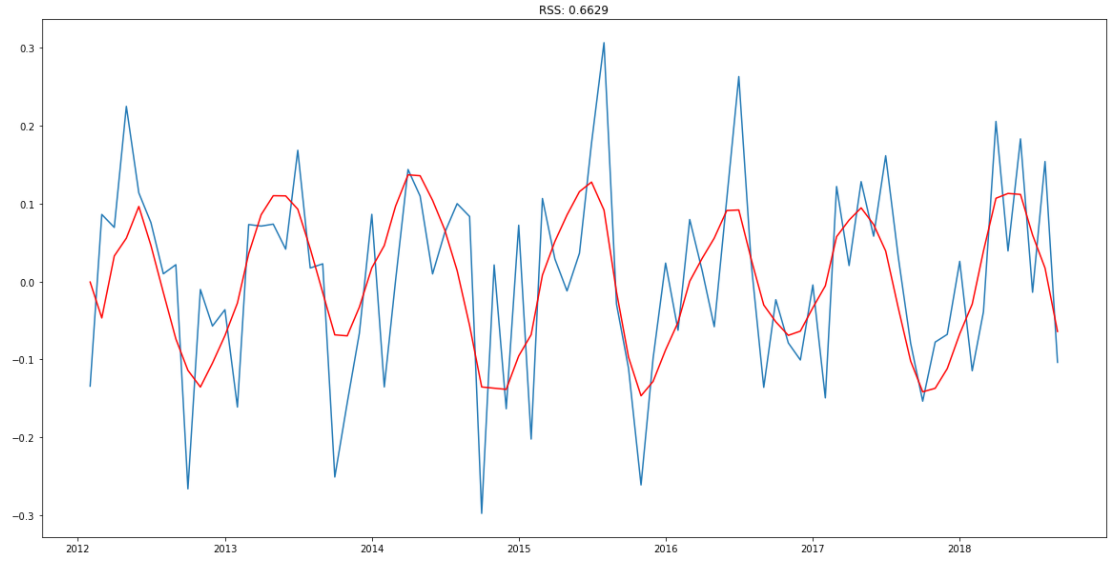
plt.plot(data\_shift)

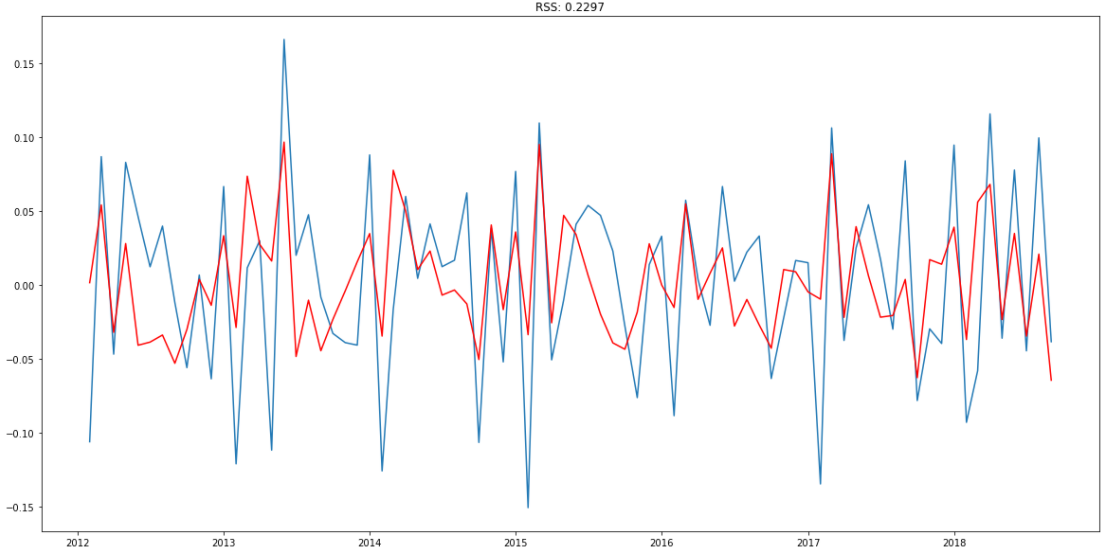
plt.plot(results.fittedvalues, color='red')

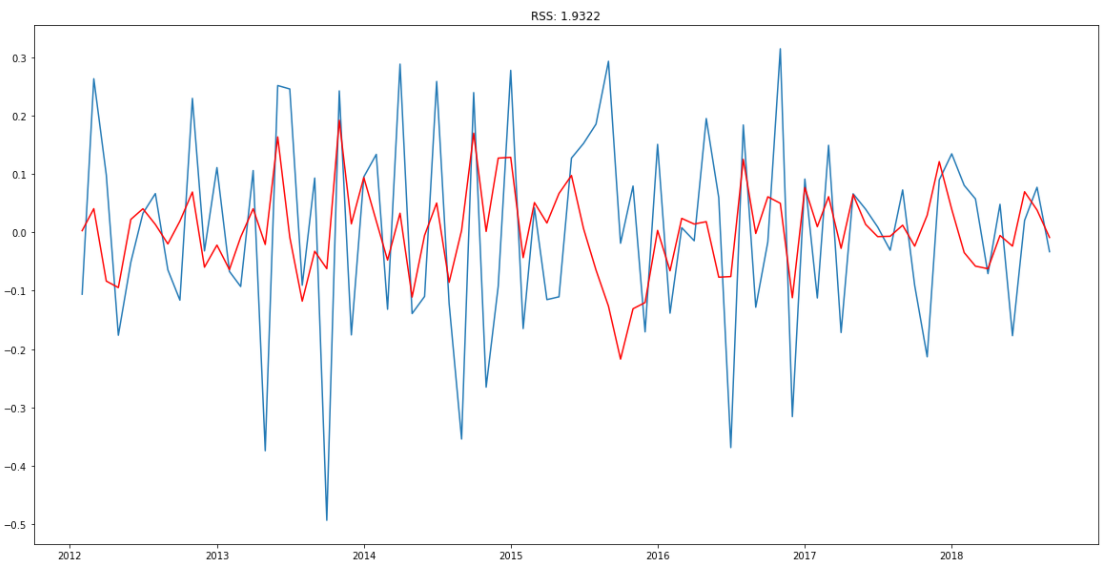
plt.title('RSS: %.4f'% sum((results.fittedvalues-data\_shift['total gallons'])\*\*2))

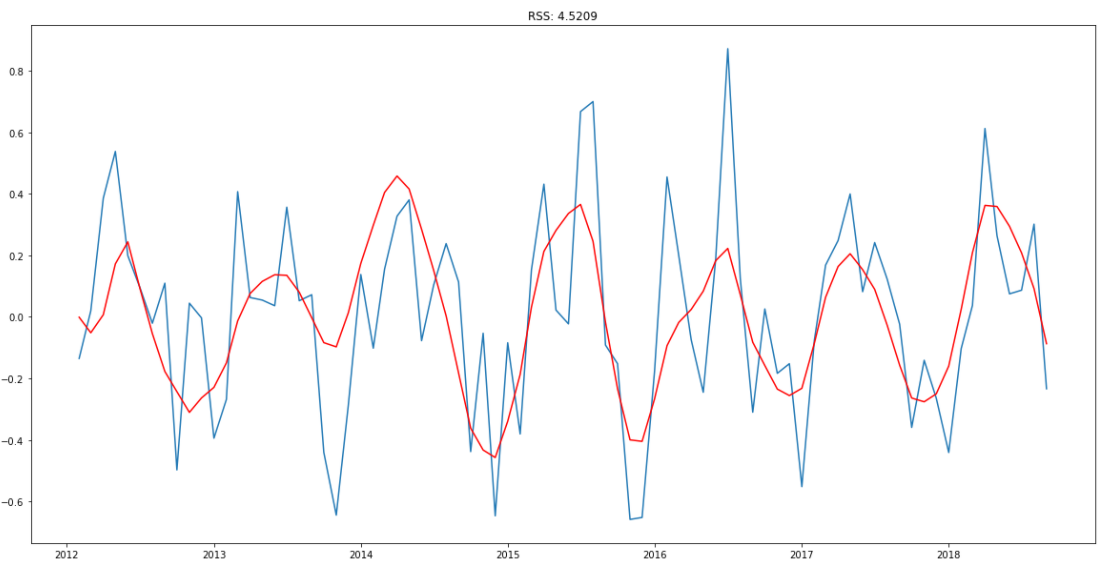
print('plotting ARIMA model')

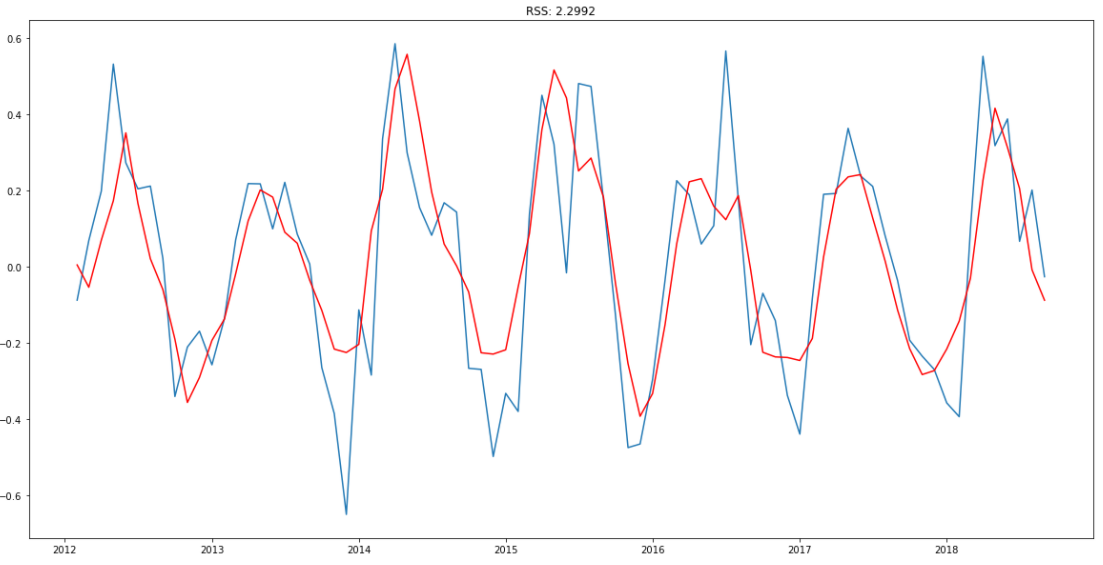
# We can write the same code for all the four attributes as well











**4.8 PREDICTION**

predictions=pd.Series(results.fittedvalues, copy=True)

print(predictions.head())

# convert to cumulative sum

predictions\_cum\_sum=predictions.cumsum()

print(predictions\_cum\_sum.head())

#

predictions\_log=pd.Series(data\_log['total gallons'], index=data\_log.index)

predictions\_log=predictions\_log.add(predictions\_cum\_sum,fill\_value=0)

predictions\_log.head()

#

predictions\_ARIMA=np.exp(predictions\_log)

plt.figure(figsize=(20,10))

plt.plot(data)

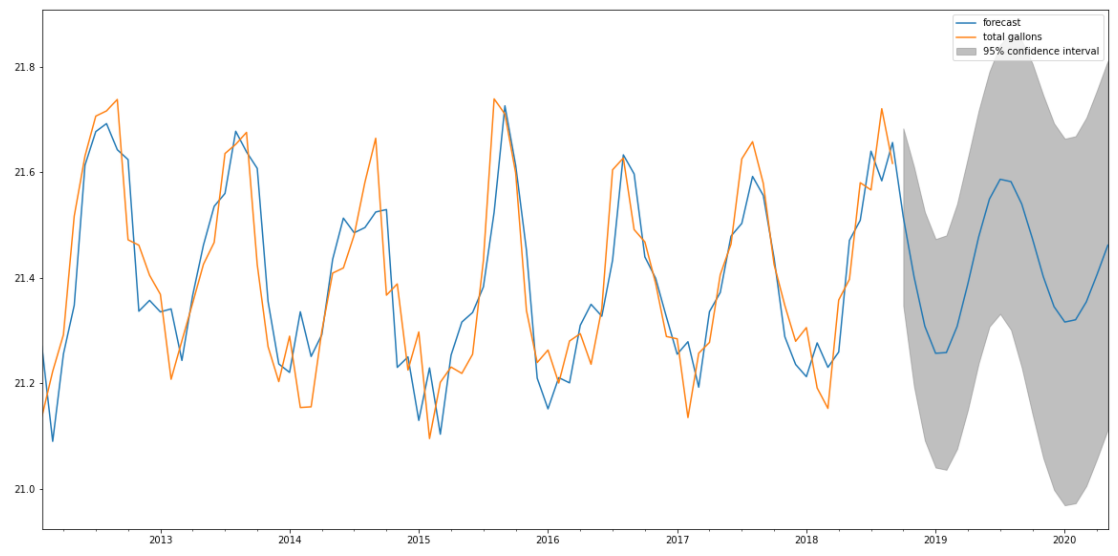
plt.plot(predictions\_ARIMA)

# Enter the steps till which you want to take predictions

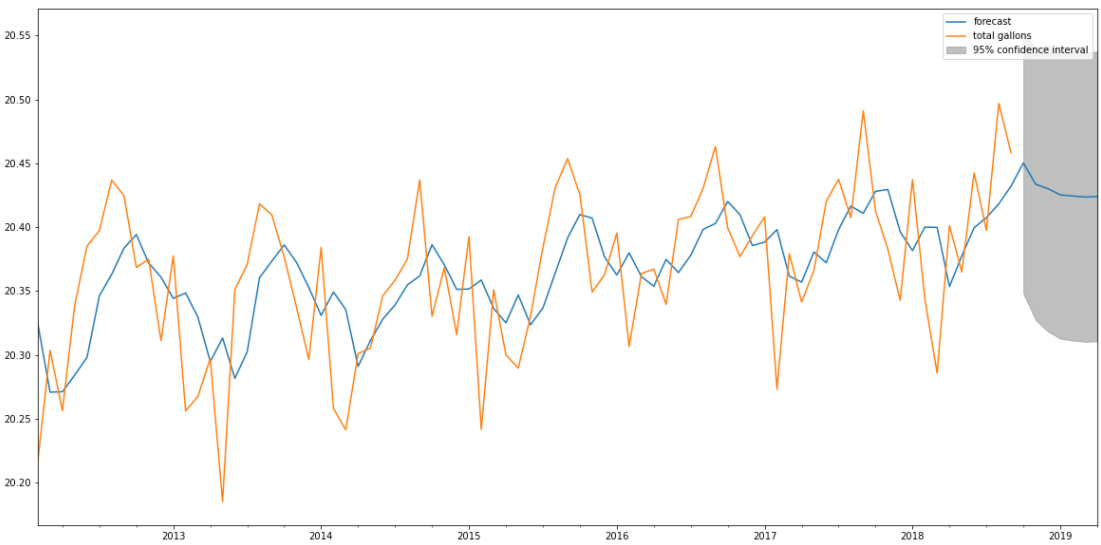
rcParams['figure.figsize']=20,10

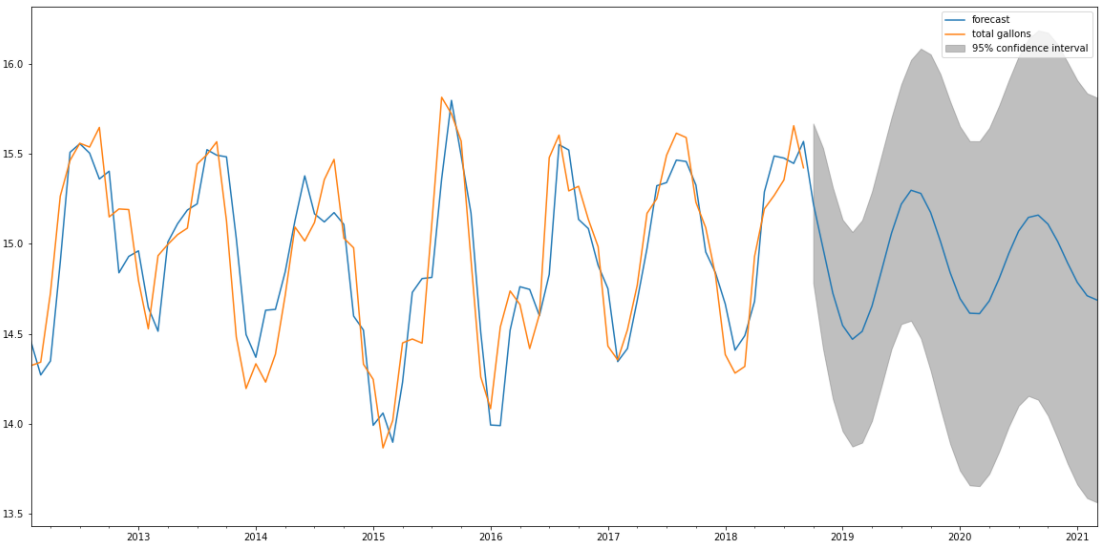
results.plot\_predict(1,105)

# We can write the same code for all the four attributes as well





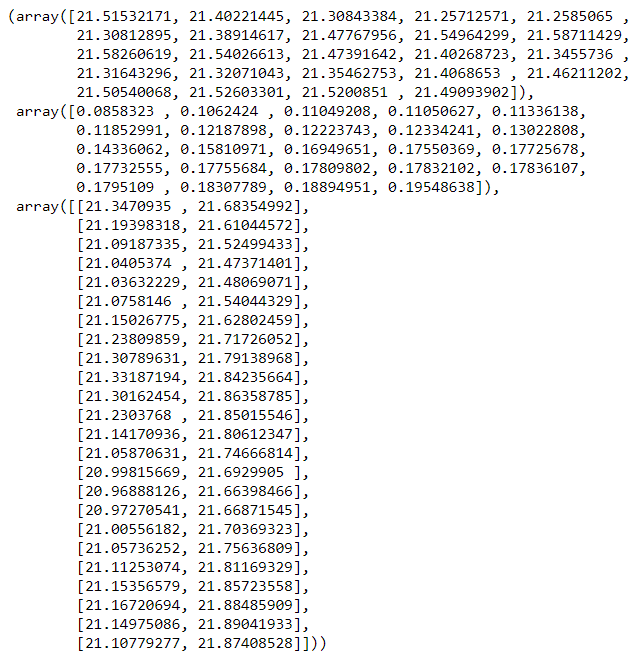


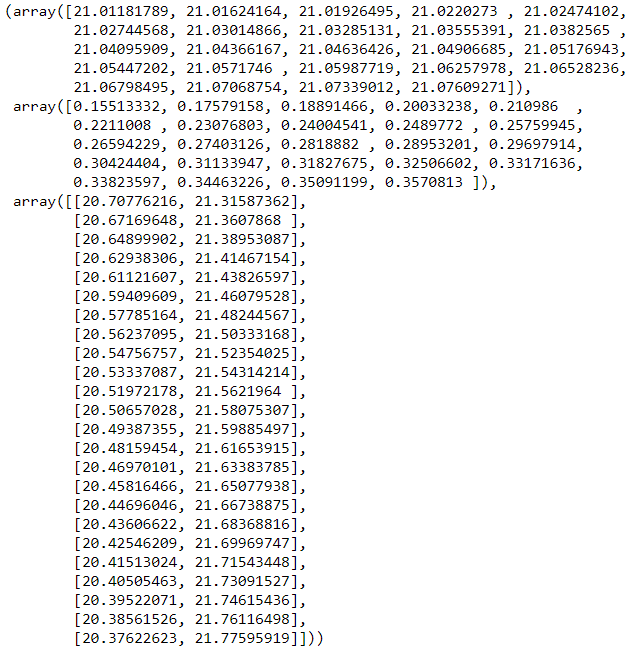
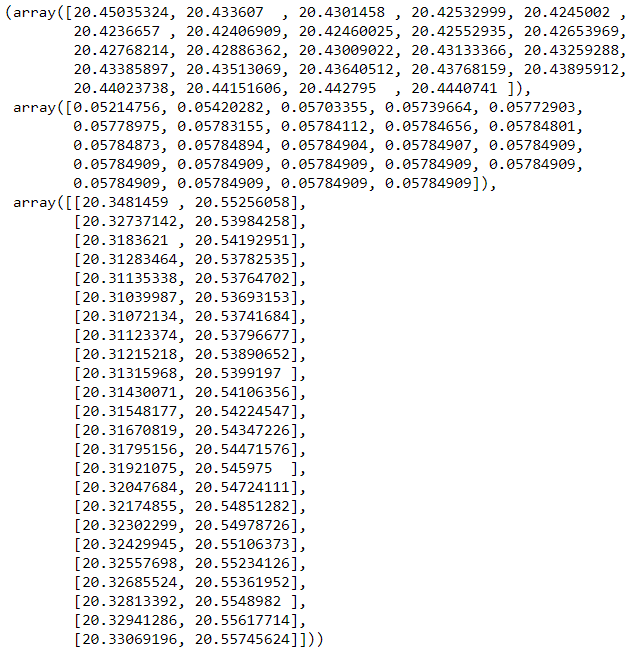


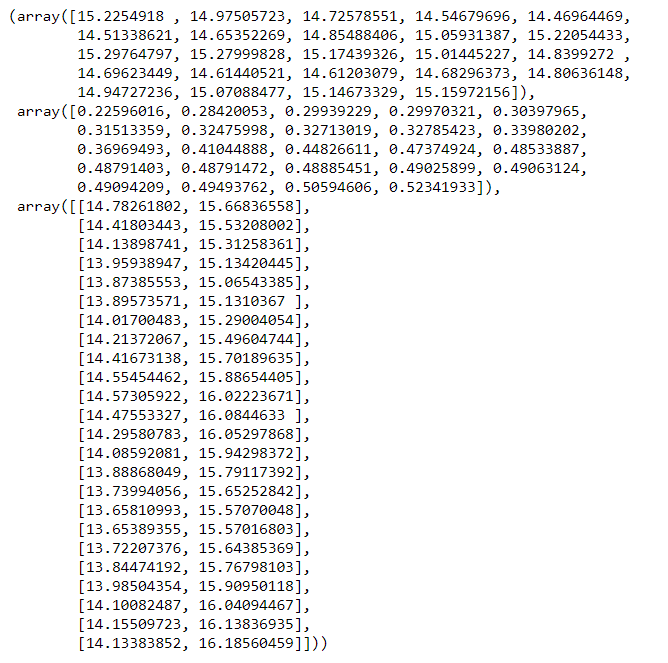
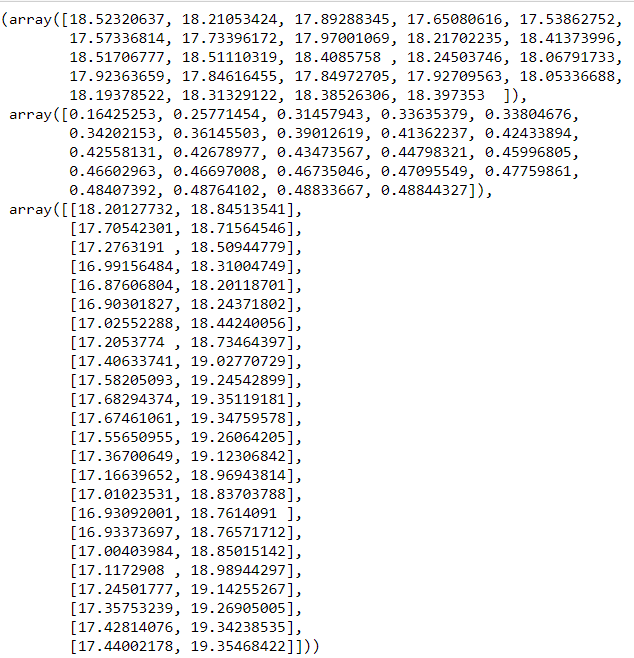


**4.9 PREDICTED VALUES IN A LOGARITHMIC FORM INSIDE AN ARRAY**

results.forecast(steps=24)





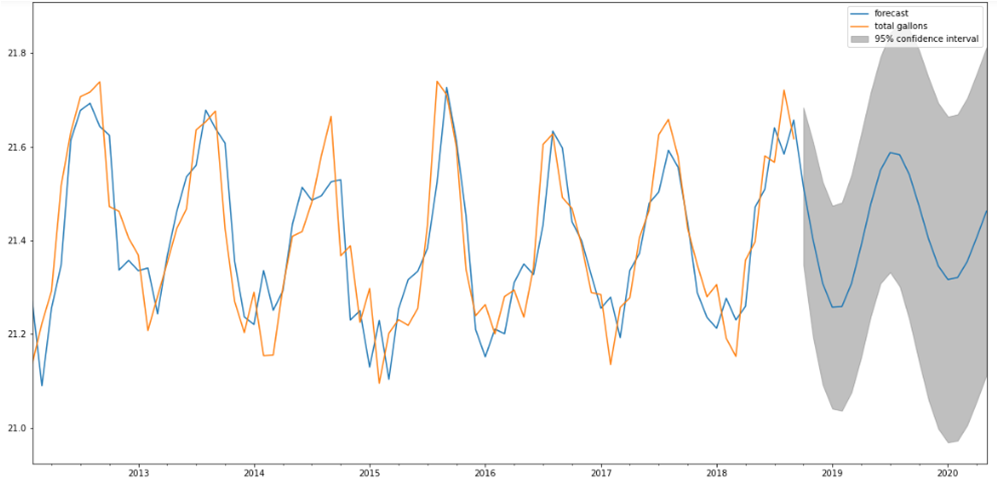


**CHAPTER-5**

**RESULT AND CONCLUSION**

**5.1: RESULT**

In this project we forecast that how much gallons of water are going to consumption at the month level in the future. Here we took the data of year 2012 to 2018 and we are going to predict quantity of water in gallons for next 10 years. So now we have this kind of data and we need to analyse what will be the th quantity of water if we have to do it for next 10 years.

****

**5.2: CONCLUSION**

Here we successfully forecast water consumption for next two years. Here we have taken data of Austin City, USA but here we can take real time data also. By real time data we mean that we can also feed the real time data of a reservoir management system to this model.

Using this model, we estimate the amount of water to be consumed in the future. If we predict the water in the reservoir management system located in each city of the country, then we can solve a big problem. We can arrange water for the city where there is scarcity of water by taking water from the city where there is excess of water. But the main concern to be taken care of, is always keeping a check over the water resources thereby using it appropriately and moving forward in a sustainable way.