



# Time Series Analysis



Athul Anish · Follow

Published in The Startup · 9 min read · Nov 24, 2020



202



3



## Introduction to Time Series

A time series is a sequence or series of numerical data points fixed at certain chronological time order. In most cases, a time series is a sequence taken at fixed interval points in time. This allows us to accurately predict or forecast the necessities.

Time series uses line charts to show us seasonal patterns, trends, and relation to external factors. It uses time series values for forecasting and this is called extrapolation.

Time series are used in most of the real-life cases such as weather reports, earthquake prediction, astronomy, mathematical finance, and largely in any field of applied science and engineering. It gives us deeper insights into our field of work and forecasting helps an individual in increasing efficiency of output.

## Time Series Forecasting

*Time series forecasting* is a method of using a model to predict future values based on previously observed time series values.

Time series is an important part of machine learning. It figures out a seasonal pattern or trend in the observed time-series data and uses it for future predictions or forecasting. Forecasting involves taking models rich in historical data and using them to predict future observations.

One of the most distinctive features of forecasting is that it does not exactly predict the future, it just gives us a calculated estimation of what has already happened to give us an idea of what could happen.



Image Courtesy: [www.wfmanagement.blogspot.com](http://www.wfmanagement.blogspot.com)

Now let's look at the general forecasting methods used in day to day problems,

*Qualitative forecasting* is generally used when historical data is unavailable and is considered to be highly objective and judgmental.

*Quantitative forecasting* is when we have large amounts of data from the past and is considered to be highly efficient as long as there is no strong external factors in play.

The skill of a time series forecasting model is determined by its efficiency at predicting the future. This is often at the cost of being able to explain why a specific prediction was made, confidence intervals, and even better, understanding the underlying factors behind the problem.

Some general examples of forecasting are:

1. Governments forecast unemployment rates, interest rates, and expected revenues from income taxes for policy purposes.
2. Day to day weather prediction.
3. College administrators forecast enrollments to plan for facilities and faculty recruitment.
4. Industries forecast demand to control inventory levels, hire employees, and provide training.

## **Application of Time Series Forecasting**

The usage of time series models is twofold:

- Obtain an understanding of the underlying forces and structure that produced the data
- Fit a model and proceed to forecast.

There is almost an endless application of time series forecasting problems.

Below are a few of the examples from a range of industries to make the notions of time series analysis and forecasting more strong.

- Forecasting the rice yield in tons by the state each year.
- Forecasting whether an EEG trace in seconds indicates a patient is having a heart attack or not.
- Forecasting the closing price of stock each day.
- Forecasting the birth or death rate at all hospitals in a city each year.
- Forecasting product sales in units sold each day.
- Forecasting the number of passengers booking flight tickets each day.
- Forecasting unemployment for a state each quarter
- Forecasting the size of the tiger population in a state each breeding season.

Now let's look at an example,

We are going to use the google new year resolution dataset,

### **Step 1: Import Libraries**

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
sns.set()
```

Picture 1

## Step 2: Load Dataset

```
df = pd.read_csv('D:\Analytics\TimeSeries/multiTimeline.csv', skiprows=1)
df.head()
```

	Month	diet: (Worldwide)	gym: (Worldwide)	finance: (Worldwide)
0	2004-01	100	31	48
1	2004-02	75	26	49
2	2004-03	67	24	47
3	2004-04	70	22	48
4	2004-05	72	22	43

Picture 2

## Step 3: Change month column into the DateTime data type

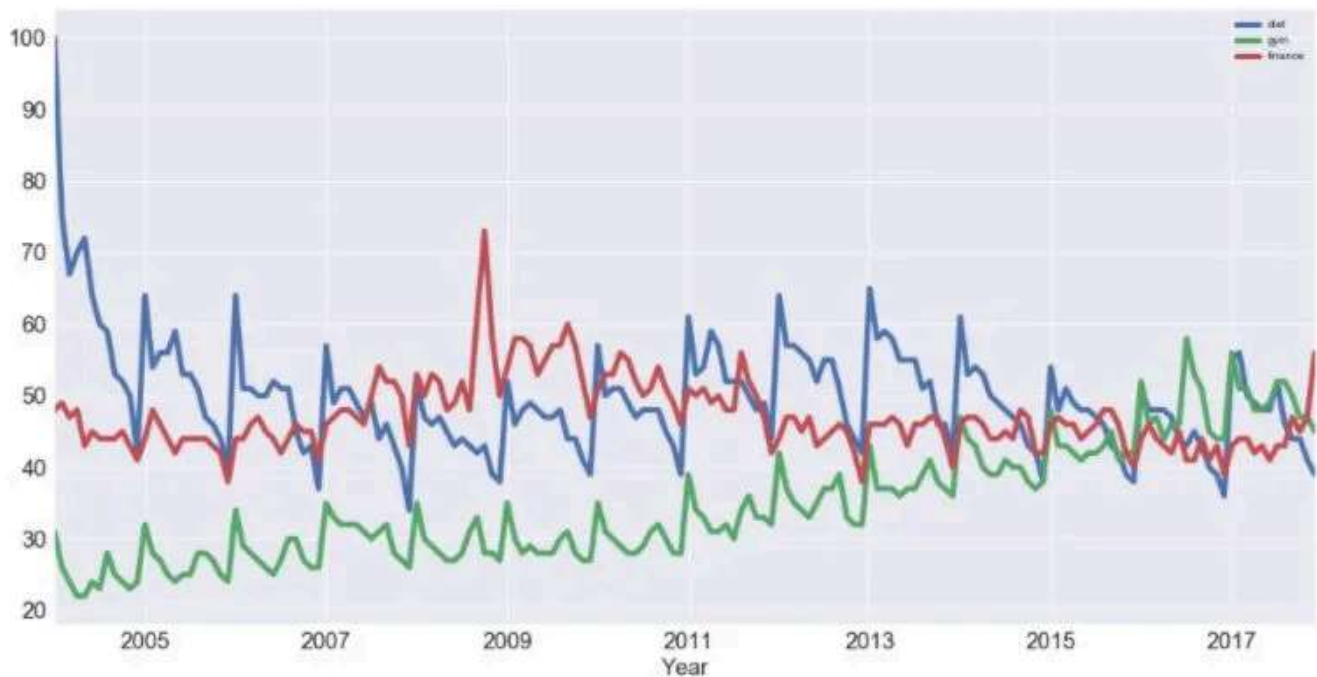
```
df.month = pd.to_datetime(df.month)
df.set_index('month', inplace=True)
df.head ()
```

Picture 3

## Step 4: Plot and visualize

```
df.plot(figsize=(20,10), linewidth=3, fontsize=20)
plt.xlabel('Year', fontsize=20);
```

Picture 4.1

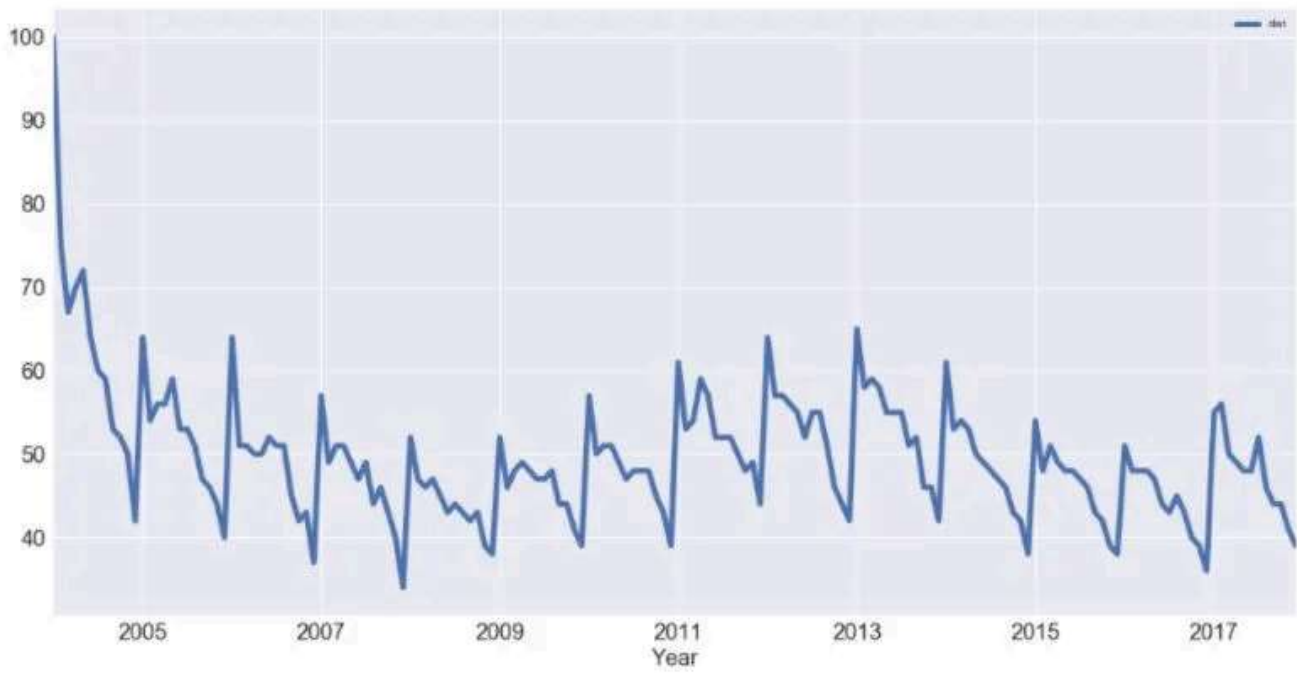


Picture 4.2

## Step 5: Check for trend

```
diet = df[['diet']]
diet.rolling(12).mean().plot(figsize=(20,10), linewidth=5, fontsize=20)
plt.xlabel('Year', fontsize=20);
```

Picture 5.1

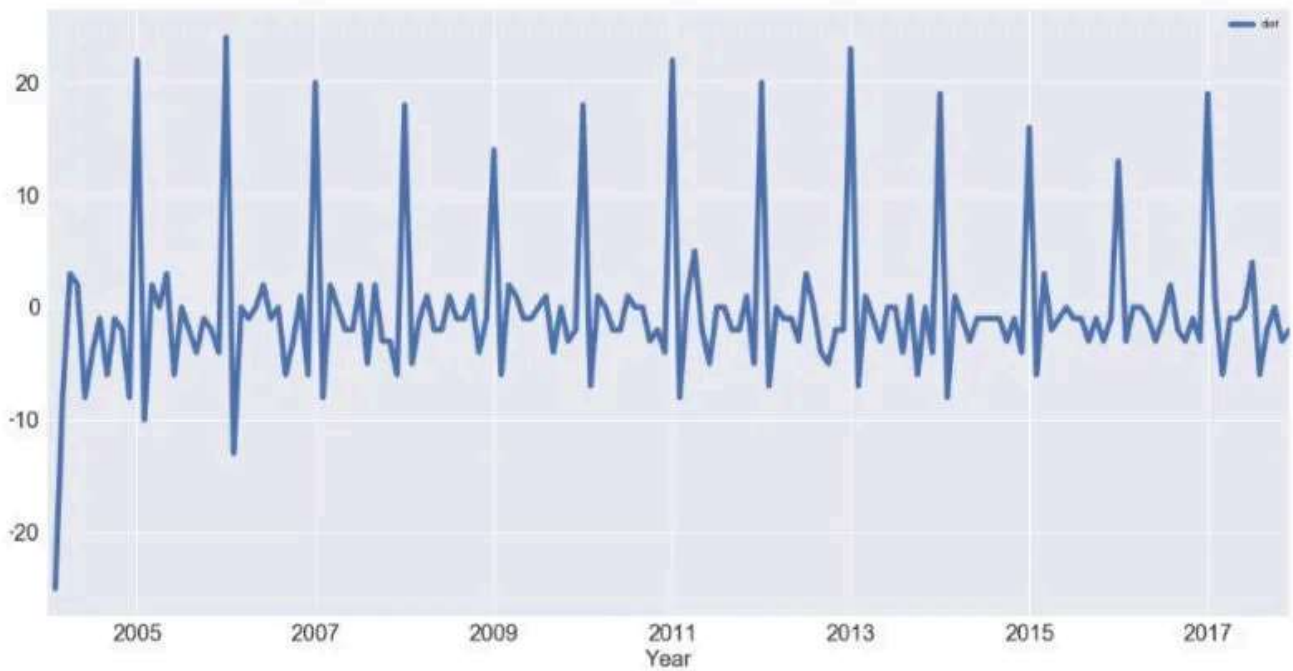


Picture 5.2

## Step 6: Check for seasonality

```
diet.diff().plot(figsize=(20,10), linewidth=5, fontsize=20)  
plt.xlabel('Year', fontsize=20);
```

Picture 6.1



Picture 6.2

We can see that there is roughly a 20% spike each year, this is seasonality.

## Components of Time Series

Time series analysis provides a ton of techniques to better understand a dataset.

Perhaps the most useful of these is the splitting of time series into 4 parts:

1. **Level:** The base value for the series if it were a straight line.
2. **Trend:** The linear increasing or decreasing behavior of the series over time.
3. **Seasonality:** The repeating patterns or cycles of behavior over time.
4. **Noise:** The variability in the observations that cannot be explained by the model.

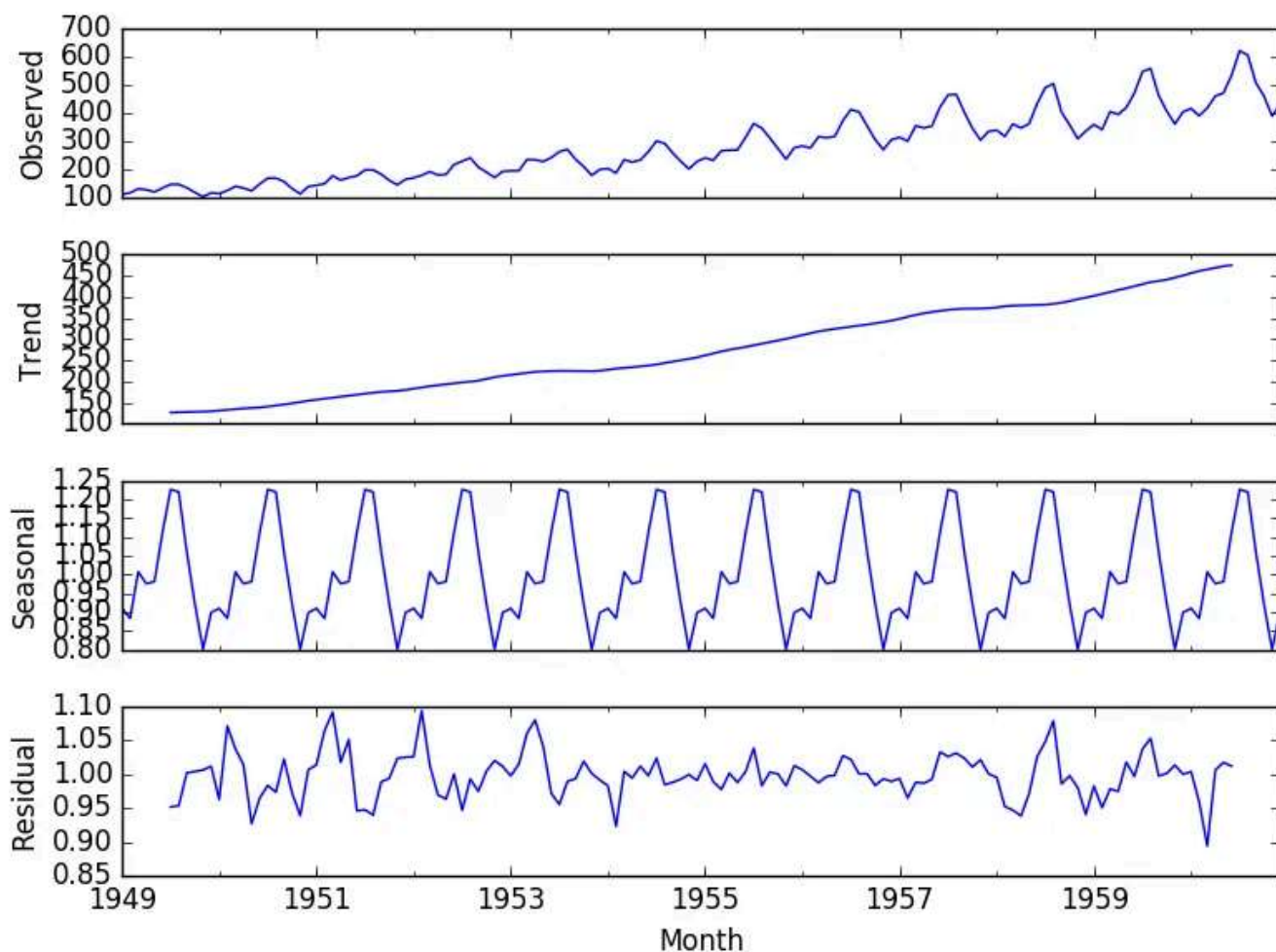


All-time series generally have a level, noise, while trend and seasonality are optional.

The main features of many time series are trends and seasonal variation. Another feature of most time series is that observations close together in time tend to be correlated

These components combine in some way to provide the observed time series. For example, they may be added together to form a model such as:

$$Y = \text{levels} + \text{trends} + \text{seasonality} + \text{noise}$$



These components are the most effective way to make predictions about future values, but may not always work. That depends on the amount of data we have about the past.

## **Analyzing Trend**

Checking out data for repeated behavior in its graphical representation is known as a Trend analysis. As long as the trend is continuously increasing or decreasing that part of data analysis is generally not very difficult. If the time series data contains some kind of considerable error, then the first step in the process of trend identification is smoothing.

*Smoothing.* Smoothing always involves some form of local averaging of data such that the components of individual observations cancel each other out. The most widely used technique is moving average smoothing which replaces each element of the series with a simple or weighted average of surrounding elements. Medians are mostly used instead of means. The main advantage of median as compared to moving average smoothing is that its results are less biased by outliers within the smoothing window. The main disadvantage of median smoothing is that in the absence of clear outliers it may produce more disturbed curves than moving average.

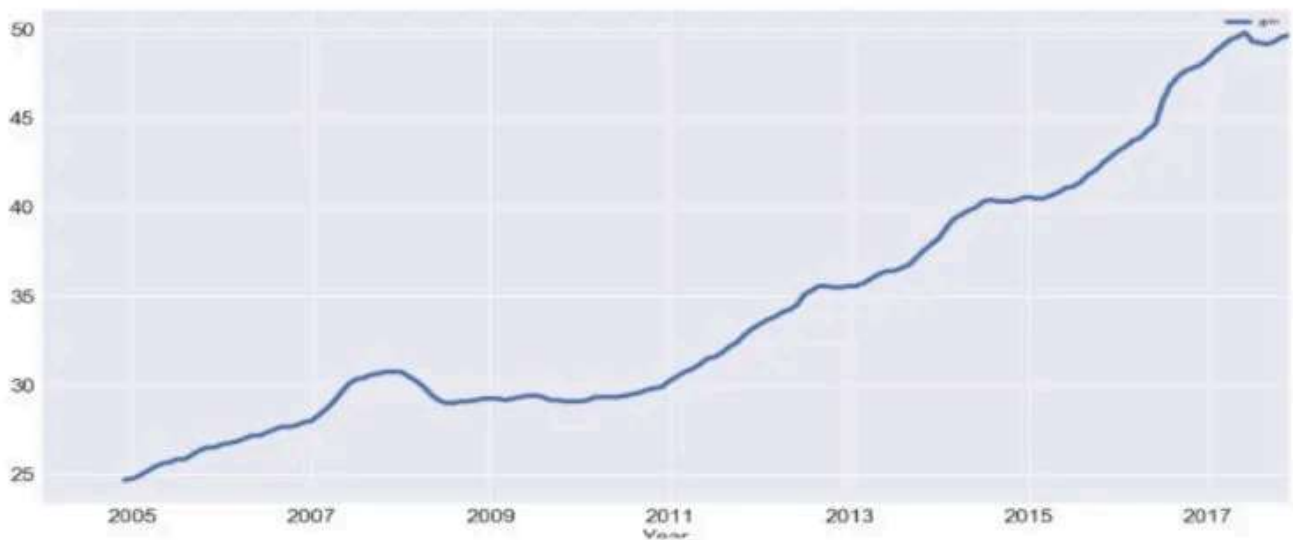
In the other less common cases, when the measurement error is quite large, the distance weighted least squares smoothing or negative exponentially weighted smoothing techniques might be used. These methods generally tend to ignore outliers and give a smooth fitting curve.

*Fitting a function.* If there is a clear monotonous nonlinear component, the data first need to be transformed to remove the nonlinearity. Usually, log, exponential, or polynomial function is used to achieve this.

Now let's take an example to understand this more clearly,

```
gym = df[['gym']]
gym.rolling(12).mean().plot(figsize=(20,10), linewidth=5, fontsize=20)
plt.xlabel('Year', fontsize=20);
```

Picture 7.1



Picture 7.2

From the above diagram, we can easily interpret that there is an *upward* trend for 'Gym' every year!

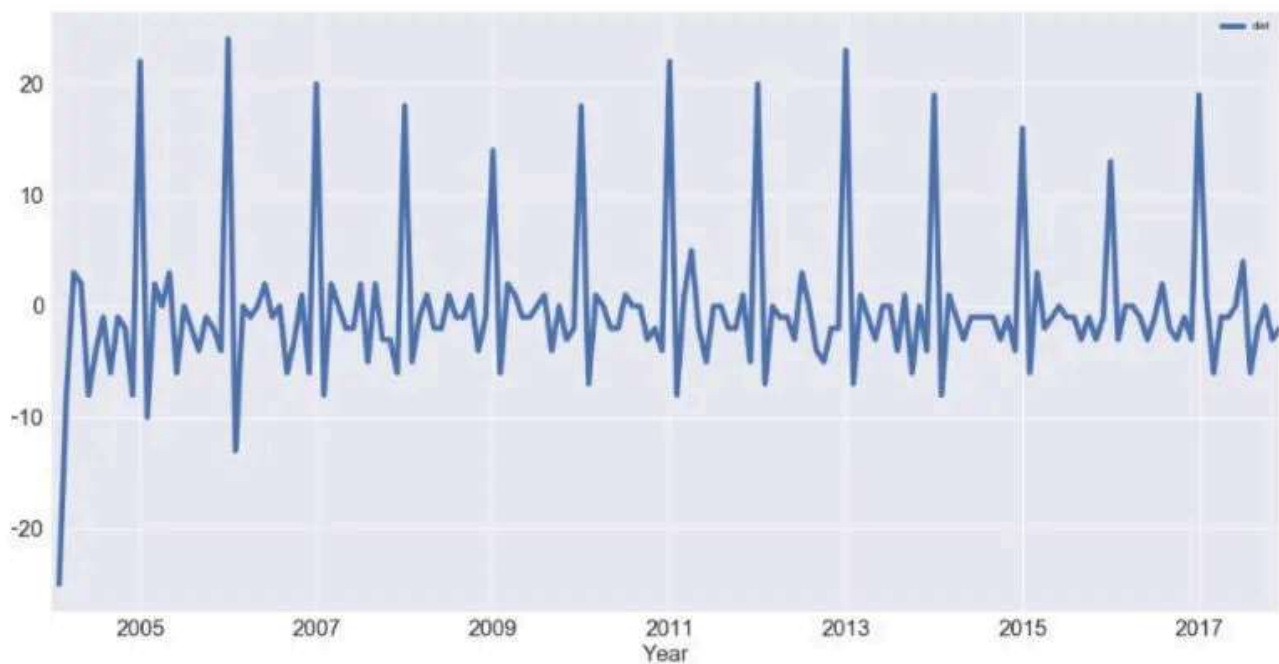
## Analyzing Seasonality

Seasonality is the repetition of data at a certain period of time interval. For example, every year we notice that people tend to go on vacation during the December — January time, this is seasonality. It is one other most important characteristics of time series analysis. It is generally measured by autocorrelation after subtracting the trend from the data.

Lets look at another example from our dataset,

```
diet.diff().plot(figsize=(20,10), linewidth=5, fontsize=20)
plt.xlabel('Year', fontsize=20);
```

Picture 8.1



Picture 8.2

From the above graph, it is clear that there is a spike at the starting of every year. Which means every year January people tend to take ‘Diet’ as their resolution rather than any other month. This is a perfect example of seasonality.

## AR, MA, and ARIMA

### Autoregression Model (AR)

AR is a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step. A regression model like linear regression takes the form of:

$$\hat{y} = b_0 + (b_1 * X_1)$$

This technique can be used on time series where input variables are taken as observations at previous time steps, called lag variables. This would look like:

$$X_{t+1} = b_0 + (b_1 * X_t) + (b_2 * X_{t-1})$$

Since the regression model uses data from the same input variable at previous time steps, it is referred to as autoregression.

## **Moving Average Model (MA)**

The residual errors from forecasts in a time series provide another source of information that can be modeled. The Residual errors form a time series. An autoregression model of this structure can be used to foresee the forecast error, which in turn can be used to correct forecasts.

Structure in the residual error may consist of trend, bias & seasonality which can be modeled directly. One can create a model of the residual error time series and predict the expected error of the model. The predicted error can then be subtracted from the model prediction & in turn provide an additional lift in performance.

An autoregression of the residual error is the Moving Average Model.

## Autoregressive Integrated Moving Average (ARIMA)

- Autoregressive integrated moving average or ARIMA is a very important part of statistics, econometrics, and in particular time series analysis.
- ARIMA is a forecasting technique that gives us future values entirely based on its inertia.
- Autoregressive Integrated Moving Average (ARIMA) models include a clear cut statistical model for the asymmetrical component of a time series that allows for non-zero autocorrelations in the irregular component
- ARIMA models are defined for stationary time series. Therefore, if you start with a non-stationary time series, you will first need to 'difference' the time series until you attain stationary time series.

An ARIMA model can be created using the statsmodels library as follows:

1. Define the model by using **ARIMA()** and passing in the p, d, and q parameters.
2. The model is prepared on the training data by calling the **fit()** function.
3. Predictions can be made by using the **predict()** function and specifying the index of the time or times to be predicted.

Now let's look at an example,

We are going to use a dataset called 'Shampoo sales'

```

#import the library
from statsmodels.tsa.arima_model import ARIMA

#load the dataset
data = pd.read_csv('D:/Analytics/TimeSeries/shampoosales.csv')

#ARIMA
model = ARIMA(np.asarray(data), order=(5,1,0))
model_fit = model.fit(dis=0)
print(model_fit.summary())

```

Picture 9.1

ARIMA Model Results						
Dep. Variable:	D.Sales	No. Observations:	35			
Model:	ARIMA(5, 1, 0)	Log Likelihood	-196.170			
Method:	css-mle	S.D. of innovations	64.241			
Date:	Mon, 12 Dec 2016	AIC	406.340			
Time:	11:09:13	BIC	417.227			
Sample:	02-01-1901	HQIC	410.098			
	- 12-01-1903					
	coef	std err	z	P> z	[95.0% Conf. Int.]	
const	12.0649	3.652	3.304	0.003	4.908	19.222
ar.L1.D.Sales	-1.1082	0.183	-6.063	0.000	-1.466	-0.750
ar.L2.D.Sales	-0.6203	0.282	-2.203	0.036	-1.172	-0.068
ar.L3.D.Sales	-0.3606	0.295	-1.222	0.231	-0.939	0.218
ar.L4.D.Sales	-0.1252	0.280	-0.447	0.658	-0.674	0.424
ar.L5.D.Sales	0.1289	0.191	0.673	0.506	-0.246	0.504
Roots						
	Real	Imaginary	Modulus	Frequency		
AR.1	-1.0617	-0.5064j	1.1763	-0.4292		
AR.2	-1.0617	+0.5064j	1.1763	0.4292		
AR.3	0.0816	-1.3804j	1.3828	-0.2406		
AR.4	0.0816	+1.3804j	1.3828	0.2406		
AR.5	2.9315	-0.0000j	2.9315	-0.0000		

Picture 9.2

## ACF and PACF

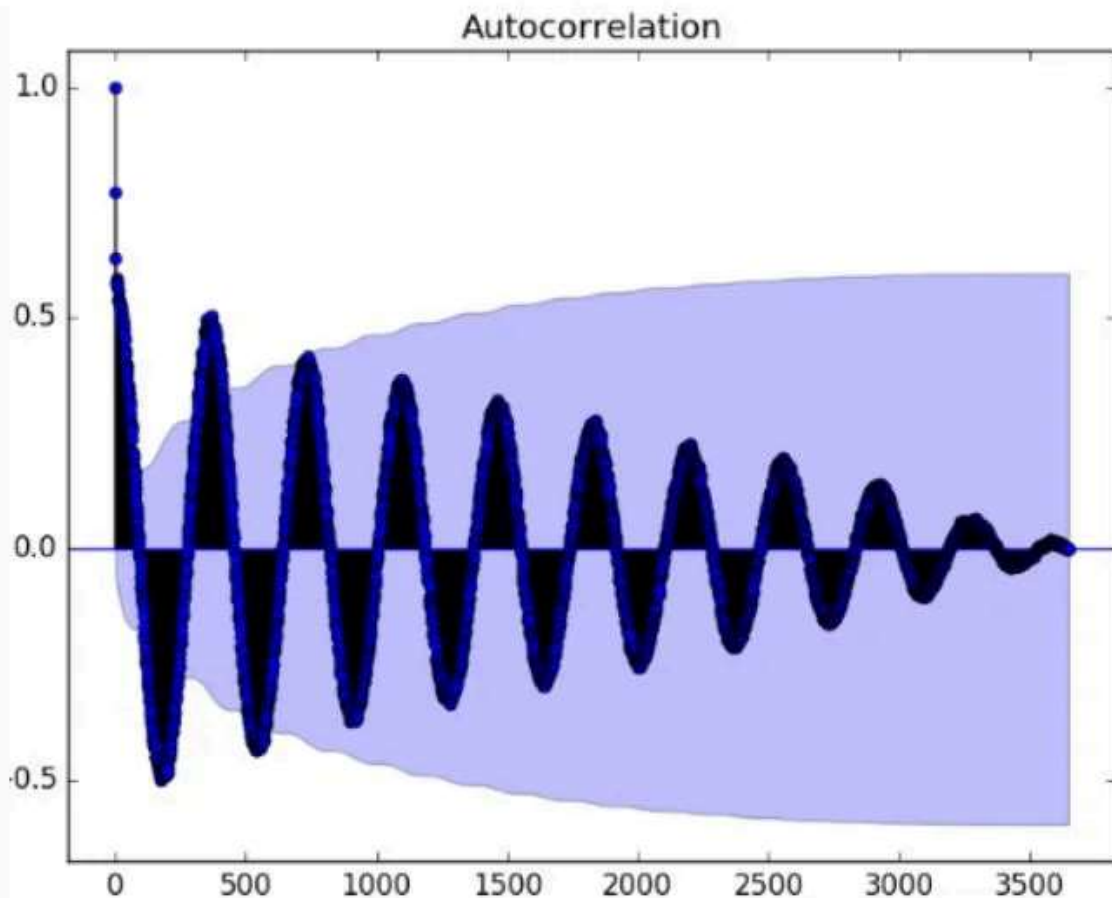


We can calculate the correlation for time-series observations with observations from previous time steps, called *lags*. Since the correlation of the time series observations is calculated with values of the same series at previous times, this is called a serial correlation, or an autocorrelation.

A plot of the autocorrelation of a dataset of a time series by lag is called the AutoCorrelation Function, or the acronym ACF. This plot is sometimes called a *correlogram* or an *autocorrelation plot*.

For example,

```
from statsmodels.graphics.tsaplots import plot_acf
plot_acf(data)
pyplot.show()
```



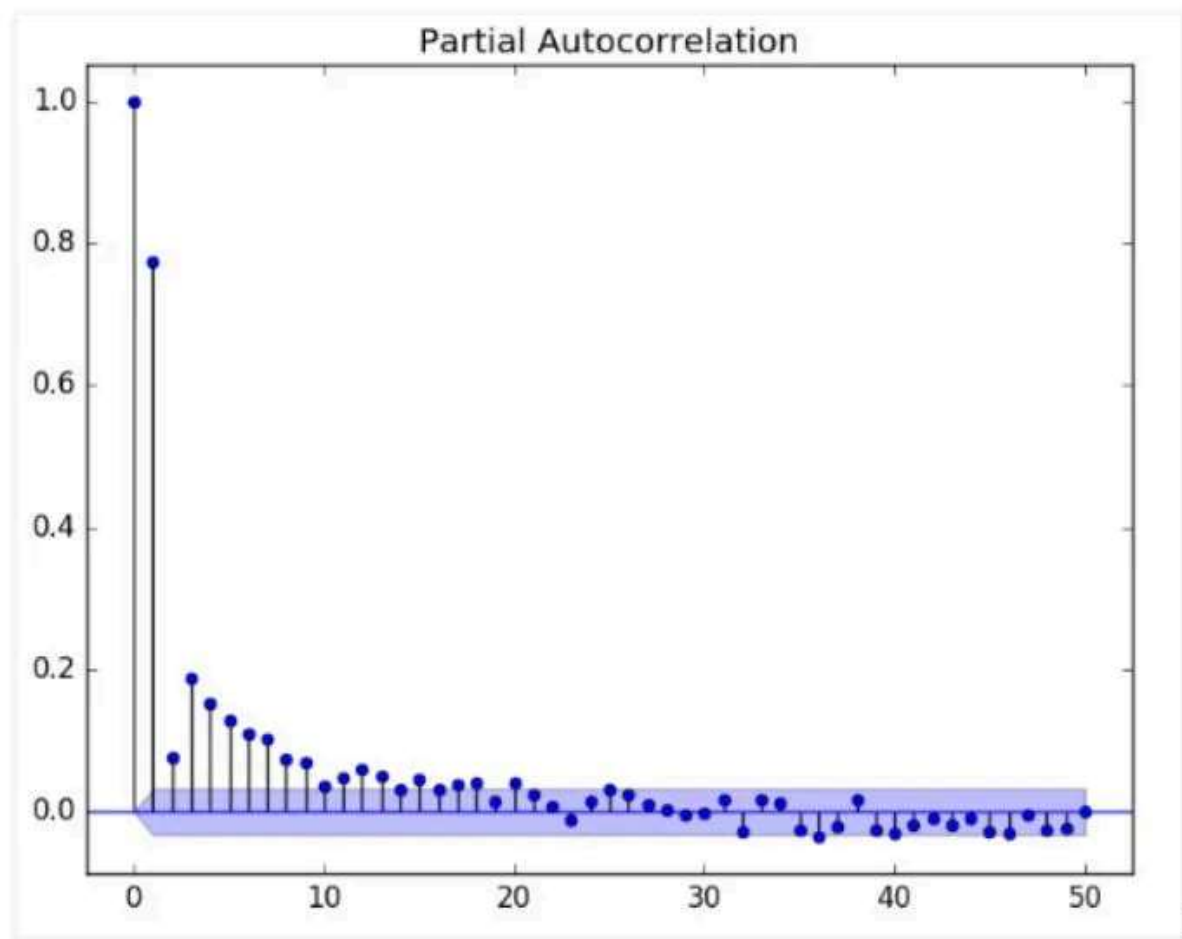


Picture 10

A partial autocorrelation or *PACF* is a summary of the relationship between an observation in a time series with observations at prior time steps with the relationships of in between observations removed.

For example,

```
from statsmodels.graphics.tsaplots import plot_pacf
plot_pacf(data, lags=50)
pyplot.show()
```



Picture 11

## Conclusion

Time series analysis is one of the most important aspect of data analytics for any large organization as it helps in understanding seasonality, trends, cyclicalities and randomness in the sales and distribution and other attributes. These factors help companies in making a well informed decision which is highly crucial for business.

[Data](#)[Data Science](#)[Startup](#)[Machine Learning](#)[Artificial Intelligence](#)

**Written by Athul Anish**

42 Followers · Writer for The Startup

Multiskilled Engineer | Data Analyst.


Follow



---

**More from Athul Anish and The Startup**





 Athul Anish in The Startup

## Principal Component Analysis

Introduction to PCA

Nov 16, 2020  29

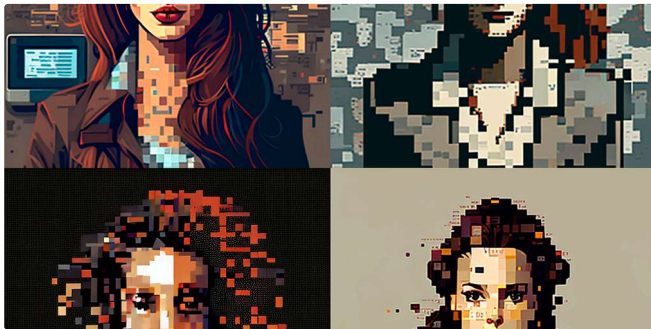



 Anangsha Alammyan  in The Startup

## You're Not Making Money As A Writer Because You're Focusing on...

People often ask me how long it took to make money as a complete beginner.

 Aug 22  10.5K  308



 Zulie Rane in The Startup

## If You Want to Be a Creator, Delete All (But Two) Social Media Platforms

In October 2022, during the whole Elon Musk debacle, I finally deleted Twitter from my...

 Apr 19, 2023  53K  1300



 Athul Anish

## Exploratory Data Analysis.

What is Exploratory Data Analysis?

Oct 28, 2020  50



See all from Athul Anish

See all from The Startup


# Recommended from Medium



 Chanaka

## Signal processing Basics

Signal processing is a field that deals with analyzing, modifying, and manipulating...

Jun 21  20  



 Palash Mishra

## Predicting Time Series Data with Machine Learning, Generative AI,...

Time series data prediction is a critical aspect of various industries, ranging from finance...

 Jun 19  713  9  

## Lists



### Predictive Modeling w/ Python

20 stories · 1602 saves



### Practical Guides to Machine Learning

10 stories · 1959 saves



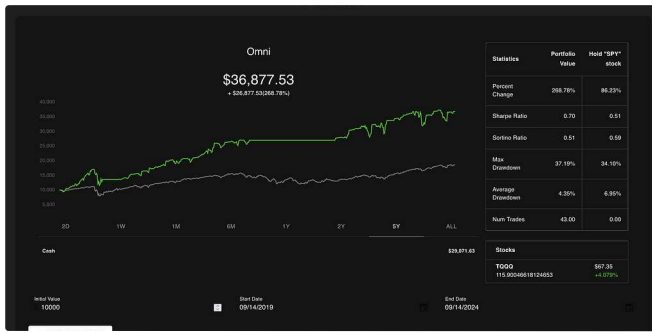
### Natural Language Processing

1765 stories · 1366 saves



### ChatGPT prompts

50 stories · 2119 saves



Austin Starks in DataDrivenInvestor

I used OpenAI's o1 model to develop a trading strategy. It is DESTROYIN...  
It literally took one try. I was shocked.

★ Sep 16 🖱 4.2K 💬 118 📌 ⋮



Vavt Llc

Power of Pandas: Data Science and Analysis  
Hello Folks 😊,

★ Sep 24 🖱 42 📌 ⋮



Ahmad Waleed in PythonForAll

Forecasting Time Series Data with SARIMAX: A Step-by-Step Guide

Time series forecasting plays a pivotal role in fields like finance, economics, and weather...

★ Aug 13 🖱 37 📌 ⋮

**AMAZON.COM** JEROME, WA  
*Software Development Engineer* Mar. 2020 – May 2021

- Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
- Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
- Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by \$25 Million
- Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

#### Projects

##### NinjaPrep.io (React)

- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

##### HeatMap (JavaScript)

- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay

Alexander Nguyen in Level Up Coding

The resume that got a software engineer a \$300,000 job at Google.

1-page. Well-formatted.

★ Jun 1 🖱 24K 💬 486 📌 ⋮

See more recommendations