Contents

[**Fundamental terms of Time series Data** 3](#_Toc108693580)

[**A) Time Series Data:** 3](#_Toc108693581)

[**B) Time Series Analysis:** 3](#_Toc108693582)

[**C) Time Series Forecasting:** 4](#_Toc108693583)

[**GOAL-PLAN-FORECAST** 4](#_Toc108693584)

[**Basic steps involved in any forecasting problem** 4](#_Toc108693585)

[**Three important characteristics of time series data** 5](#_Toc108693586)

[**Components associated with time series** 6](#_Toc108693587)

[**1)** **Level:** 6](#_Toc108693588)

[**2)** **Trend:** 6](#_Toc108693589)

[**3)** **Seasonality:** 6](#_Toc108693590)

[**4)** **Cyclicity:** 7](#_Toc108693591)

[**5)** **Noise:** 7](#_Toc108693592)

[**Handling missing values in Time series** 8](#_Toc108693593)

[**1.Mean Imputation** 8](#_Toc108693594)

[**2.Last observation carried forward**: 8](#_Toc108693595)

[**3.Linear interpolation** 8](#_Toc108693596)

[**4.Seasonal + Linear interpolation** 9](#_Toc108693597)

[**Method of detecting outliers:** 9](#_Toc108693598)

[**2.Box Plot** 10](#_Toc108693599)

[**3.Histogram:** 10](#_Toc108693600)

[**Handling Outliers in Time series Data** 10](#_Toc108693601)

[**Time series Decomposition** 10](#_Toc108693602)

[**Additive Seasonal Decomposition** 11](#_Toc108693603)

[**Multiplicative Seasonal Decomposition** 11](#_Toc108693604)

[**Time Series VS Regression** 11](#_Toc108693605)

[**Forecasting methods** 13](#_Toc108693606)

[**1.Naive method** 13](#_Toc108693607)

[**2.Simple average method: -** 13](#_Toc108693608)

[**3. Simple Moving Average Method:-** 14](#_Toc108693609)

[**Popular error measures** 17](#_Toc108693610)

[**Mean Forecast Error (MFE):** 18](#_Toc108693611)

[**Mean Absolute Error (MAE):** 18](#_Toc108693612)

[**Mean Absolute Percentage Error (MAPE):** 18](#_Toc108693613)

[**Mean Squared Error (MSE):** 19](#_Toc108693614)

[**Root Mean Squared Error (RMSE):** 19](#_Toc108693615)

[**Cross-validation** 19](#_Toc108693616)

[**One-Step Validation** 19](#_Toc108693617)

[**Multi-Step Validation** 20](#_Toc108693618)

[**Assumptions to build an autoregressive model** 21](#_Toc108693619)

[**Making nonstationary data to stationary** 23](#_Toc108693620)

[**ACF and PACF** 25](#_Toc108693621)

[**The Simple Auto Regressive Model (AR)** 26](#_Toc108693622)

[**Moving Average Model (MA)** 27](#_Toc108693623)

[**Auto Regressive Moving Average (ARMA)** 30](#_Toc108693624)

[**Auto Regressive Integrated Moving Average (ARIMA)** 31](#_Toc108693625)

[**Seasonal Auto Regressive Integrated Moving Average (SARIMA)** 32](#_Toc108693626)

[**Choosing The Right Time Series Method** 35](#_Toc108693627)

**Types of Forecasting**

Types of Forecasting There are two types of forecasting. They are:

1. Qualitative forecasting 2. Quantitative forecasting

The following are the differences between the quantitative and qualitative forecasting

A picture containing text

Description automatically generated

One of the quantitative forecasting methods is time series forecasting.

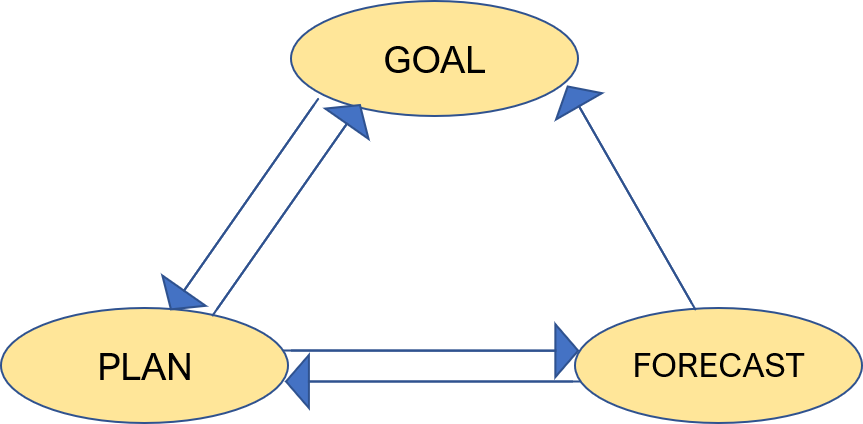
# **Fundamental terms of Time series Data**

**A) Time Series Data:** Any data that has a time component involved in it is termed as a time-series data. For example, the number of orders made on a food ordering app per day is an example of time-series data.

**B) Time Series Analysis:** Performing analysis on a time-series data to find useful insights and patterns in termed as time series analysis. Let's take a food ordering app example again. The app might have the data for every day logged in per hour. They might notice that in this data, the number of orders is significantly higher in, say, the 1-2 PM time slot but is significantly lower in the 3-4 PM time slot. This information might be useful for them as they would then be able to estimate the number of delivery boys required at a particular time of the day. Hence, time series analysis is indispensable while working with any time series data.

**C) Time Series Forecasting:** Time series forecasting is basically looking at the past data to make predictions into the future. Say that the food ordering app wants to predict the number of orders per day for the next month in order to plan the resources better. For this, they will look at tons of past data and use it in order to forecast accurately.

# **GOAL-PLAN-FORECAST**



• **Goal**: A set of business objectives. For example, maximising revenue, maximising capital, etc.

• **Plan**: A set of actions that a business takes to achieve the goal. In order to come up with a good plan, they need a forecast.

• **Forecast**: Is the prediction of the future.

# **Basic steps involved in any forecasting problem**

* Define the problem
* Collect the data
* Analyze the data
* Build and evaluate the forecast model

**Some Caveats in Time series Forecasting**

There are some ***caveats*** associated with a time series forecasting.

**The Granularity Rule**: The more aggregate your forecasts, the more accurate you are in your predictions simply because aggregated data has lesser variance and hence, lesser noise.

As a thought experiment, suppose you work at ABC, an online entertainment streaming service, and you want to predict the number of views for a few newly launched TV show in Mumbai for the next one year. Now, would you be more accurate in your predictions if you predicted at the city-level or if you go at an area-level? Obviously, accurately predicting the views from each area might be difficult but when you sum up the number of views for each area and present your final predictions at a city-level, your predictions might be surprisingly accurate. This is because, for some areas, you might have predicted lower views than the actual whereas, for some, the number of predicted views might be higher. And when you sum all of these up, the noise and variance cancel each other out, leaving you with a good prediction. Hence, you should not make predictions at very granular levels.

**The Frequency Rule**: This rule tells you to keep updating your forecasts regularly to capture any new information that comes in.

Let's continue with the ABC, an online entertainment streaming service, an example where the problem is to predict the number of views for a newly launched TV show in Mumbai for the next year. Now, if you keep the frequency too low, you might not be able to capture accurately the new information coming in. For example, say, your frequency for updating the forecasts is 3 months. Now, due to the COVID-19 pandemic, the residents may be locked in their homes for around 2-3 months during which the number of views will significantly increase. Now, if the frequency of your forecast is only 3 months, you will not be able to capture the increase in views which may incur significant losses and lead to mismanagement.

**The Horizon Rule:** When you have the horizon planned for a large number of months into the future, you are more likely to be accurate in the earlier months as compared to the later ones. Let's again go back to ABC, an online entertainment streaming service, example. Suppose that the online entertainment streaming service made a prediction for the number of views for the next 6 months in December 2019. Now, it may have been quite accurate for the first two months, but due to the unforeseen COVID-19 situation, the actual number of view in the next couple of months would have been significantly higher than predicted because of everyone staying at home. The farther ahead we go into the future, the more uncertain we are about the forecasts.

Now that you have understood the steps in defining the problem, let’s apply them to the air passenger traffic problem.

* **Quantity**: Number of passengers
* **Granularity**: Flights from city A to city B; i.e., flights for a particular route
* **Frequency**: Monthly
* **Horizon**: 1 year (12 months)

# **Three important characteristics of time series data**

There are three important characteristics that every time series data must exhibit in order for us to make a good forecast.

* **Relevant**: The time-series data should be relevant for the set objective that we want to achieve.
* **Accurate**: The data should be accurate in terms of capturing the timestamps and capturing the observation correctly.
* **Long enough**: The data should be long enough to forecast. This is because it is important to identify all the patterns in the past and forecast which patterns repeat in the future.

# **Components associated with time series**

It seems like there are quite a few components associated with time series. Let's look at them one by one once again.

1. **Level:** This is the baseline of a time series. This gives the baseline to which we add the different other components.
2. **Trend:** Over a longterm, this gives an indication of whether the time series moves lower or higher. For example, in the following Sensex graph you can clearly observe that with time, the overall value is increasing i.e. this particular time series data has an increasing trend.

Every time series has a level and noise, while trend, seasonal and cyclic patterns are optional. Every time series has a level and noise, while trend, seasonal and cyclic patterns are optional.

Chart, line chart

Description automatically generated

1. **Seasonality:** It is a pattern in a time-series data that repeats itself after a given period of time. For example, in the following graph 'Monthly sales data of company X', you can clearly observe that a fixed pattern is repeating every year. The simplest example to explain this could be, say, the sales of winter wear in India. In winter, during months like November-January, you would expect these sales to be very high whereas for the other months, the sales might be low. This shows a seasonality pattern and proves to be very useful when making forecasts.

Chart, line chart

Description automatically generated

1. **Cyclicity:** It is also a repeating pattern in data that repeats itself periodically. We don’t get into the more details of this component as it is out of the scope of this module. Cyclicity: It is also a repeating pattern in data that repeats itself periodically. We don’t get into the more details of this component as it is out of the scope of this module.

Chart

Description automatically generated

1. **Noise:** Noise is the completely random fluctuation present in the data and we cannot use this component to forecast into the future. This is that component of the time series data that no one can explain and is completely random.

Chart, line chart

Description automatically generated

# **Handling missing values in Time series**

Methods of handling missing values.

**1.Mean Imputation**: Imputing the missing values with the overall mean of the data.

Chart, line chart

Description automatically generated

**2.Last observation carried forward**: We impute the missing values with its previous value in the data.

Chart, line chart

Description automatically generated

**3.Linear interpolation:** You draw a straight line joining the next and previous points of the missing values in the data.

Chart, line chart

Description automatically generated

**4.Seasonal + Linear interpolation:** This method is best applicable for the data with trend and seasonality. Here, the missing value is imputed with the average of the corresponding data point in the previous seasonal period and the next seasonal period of the missing value.

Chart, line chart

Description automatically generated

# **Method of detecting outliers:**

It is important before we even start building the time series models, we should check the dataset for outliers as well apart from missing values in the dataset. This will help us analyze the forecast plots more accurately.

1.**Extreme value analysis**: Remove the smallest and largest values in the dataset.

Chart, histogram

Description automatically generated

**2.Box Plot**: The points lying on either side of the whiskers are considered to be outliers as shown in the image. The length of these whiskers is subjective and can be defined by you according to the problem.

Chart, box and whisker chart

Description automatically generated

**3.Histogram:** Simply plotting a histogram can also reveal the outliers - basically the extreme values with low frequencies visible in the plot.

Chart, histogram

Description automatically generated

# **Handling Outliers in Time series Data**

1. Mean,Median or Mode
2. Trimming
3. Lower and upper capping.
4. Zero capping

# **Time series Decomposition**

Time series can be split into its various components that is the

1. trend, b) seasonality, and c) residuals (residual is the part left over after extracting trend and seasonality from the time series).

The data can be decomposed to extract these two components individually.

There are two ways in which the time series data can be decomposed:

**Additive Seasonal Decomposition** - the individual components can be added to get the time-series data

When the magnitude of the seasonal pattern in the data does not directly correlate with the value of the series, the additive seasonal decomposition may be a better choice to split the time series so that the residual does not have any pattern.

**Multiplicative Seasonal Decomposition** - the individual components can be multiplied to get the time-series data

When the magnitude of the seasonal pattern in the data increases with an increase in data values and decreases with a decrease in the data values, the multiplicative seasonal decomposition may be a better choice.

Graphical user interface

Description automatically generated

# **Time Series VS Regression**

Time series is a series of time-stamped values. In other words, it is a sequence of values with time values attached to it. Here, the order attached to the values is very important which is not the case with a normal regression model.

Using time series analysis, you can forecast:

* the value of the stock market index for a future month, or
* the value of the literacy rate for a future census
* the population of a nation in the coming year

 In a time series, the sequence is important. For example, let’s take the data provided below.

Table

Description automatically generated

Using regression or advanced regression, let’s say you predict the value for timestamp 7. Now let’s say you shuffle the data around like this.

Table

Description automatically generated

Even though we shuffled the data in the value's column, linear regression works on the linear relationship between the variables and thus the above data will also give you the same prediction for timestamp 7 if you use regression on the second table. **However, a time series analysis will give you different forecasts for the original data and for the shuffled one**.

Why does this happen? This happens because while forecasting using time series, your model predicts not only on the basis of the values given but also on the basis of the sequence in which the values are given. Hence, the sequence is very important in a time series analysis and should not be played around with.

Two most important differences between time series and regression are:

**Time series have a strong temporal (time-based) dependence** — each of these data sets essentially consists of a series of time-stamped observations i.e. each observation is tied to a specific time instance. Thus, unlike regression, the order of the data is important in a time series.

In a time series, you are not concerned with the causal relationship between the response and explanatory variables. The cause behind the changes in the response variable is very much a black box.

For example, let’s say you want to predict what the value of the stock market index will be next month. You will not look at why the stock market index increases in value or if it’s because of an increase in GDP or there are some changes in any sector or some other factor. You will only look at the sequence of values for the past months and predict for the next month, based on that sequence.

# **Forecasting methods**

### **1.Naive method**

In the Naive forecast method, we set all the forecasts to be the value of the last observation of the train data.

Forecast = Last month’s sales

Chart, line chart

Description automatically generated

### **2.Simple average method: -**

In the simple average method, we take the average of the complete time-series data.

Forecast = Average of all past months’ sales

Chart, line chart

Description automatically generated

### **3. Simple Moving Average Method:-**

Time series has more impact on the future rather than the first observation, in the simple moving average method, we take the average of only the last few observations to forecast the future. In this method, the forecasts are calculated using the average of the time-series data in the moving window considered.

The window keeps moving, the average values keep changing. This helps in forecasting values at every step in the dataset. In any time series data, the recent observations have more influence on the forecasts than the previous observations. Thus a smaller window size generally forecasts closer to the actual values as it is able to take the variations in the time series data of the recent values. The higher the window size, these variations get distributed and as a result, the forecasts may not be very close to the actual values.

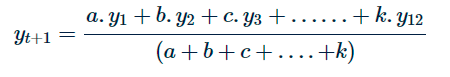
Table

Description automatically generated

**4. weighted moving average technique**

The weighted moving average technique. The underlying idea of this technique is that each observation influencing Yt+1 is assigned a specific weight.

More recent observations get more weight, whereas the previous observations get less weight. Suppose you consider a time series data of 12 months and are forecasting yt+1. Then, the weighted moving average will be calculated as follows:



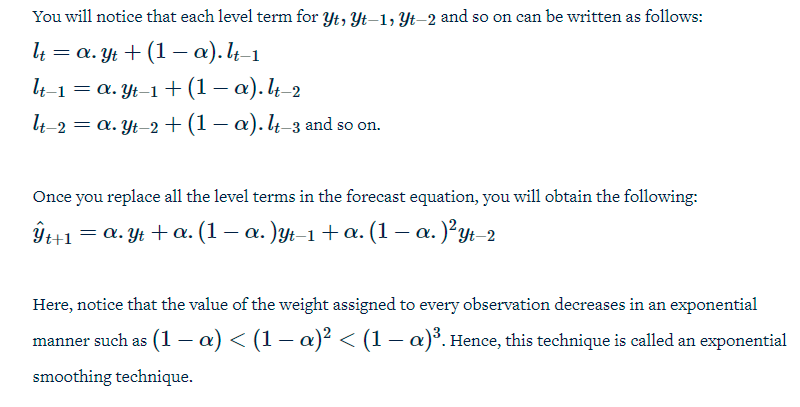
such that a<b<c<.....k, where k is the largest weight assigned to the most recent data point i.e. y12.

**5. exponential smoothing technique**

Exponential smoothing technique. In this smoothing technique, the forecast observation data, Yt+1, is a function of the level component that is denoted by lt. Here, the level component is written as follows:  


 The most recent value yt takes a weight of α, also known as the level smoothing parameter, whereas the previous observation’s level component takes the value of 1−α.

The values of α lie between 0 and 1. You can try to change the values of α such that for an optimum value of this level smoothing parameter the forecast fits well with the actual values and the subsequent values of the error terms are extremely low.

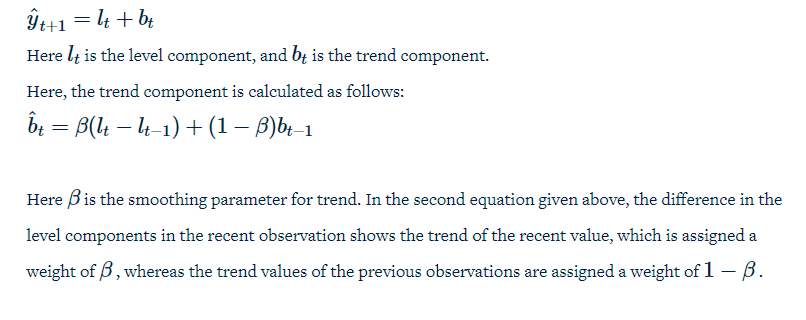


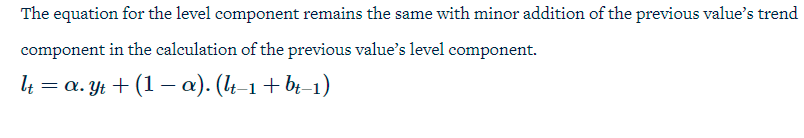
Table

Description automatically generated

**6. Holt exponential smoothing technique**

Holt's exponential smoothing technique which captures both level and trend of a time series in the forecast.  the forecast equation is a function of both level and trend that is





**7. Holt-Winters's Exponential Smoothing**

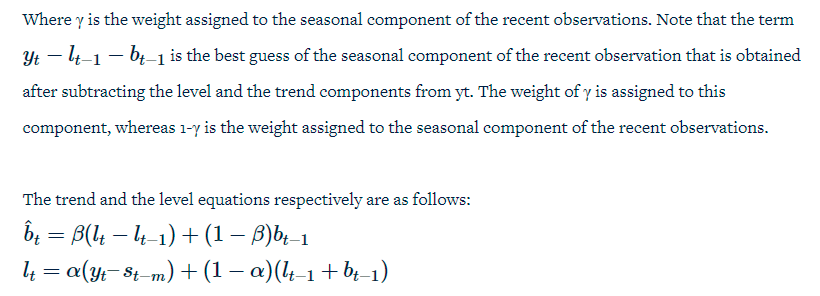
'Holt-Winters' Exponential Smoothing technique which forecasts based on level, trend, and seasonality of a time series.



Here, m is the number of times a season repeats during a period. The seasonal component is calculated using the following equation:

A picture containing text, clock, gauge

Description automatically generated



smoothing techniques: additive and multiplicative methods. In a time-series data, if the seasonality is not a function of the level component or the difference between subsequent troughs of the time series data does not increase as you progress in the graph, then the Holt-Winters’ additive method works best. You q used this method for the quarterly ice cream sales example because as you can observe in the image below, the difference between the troughs does not increase.

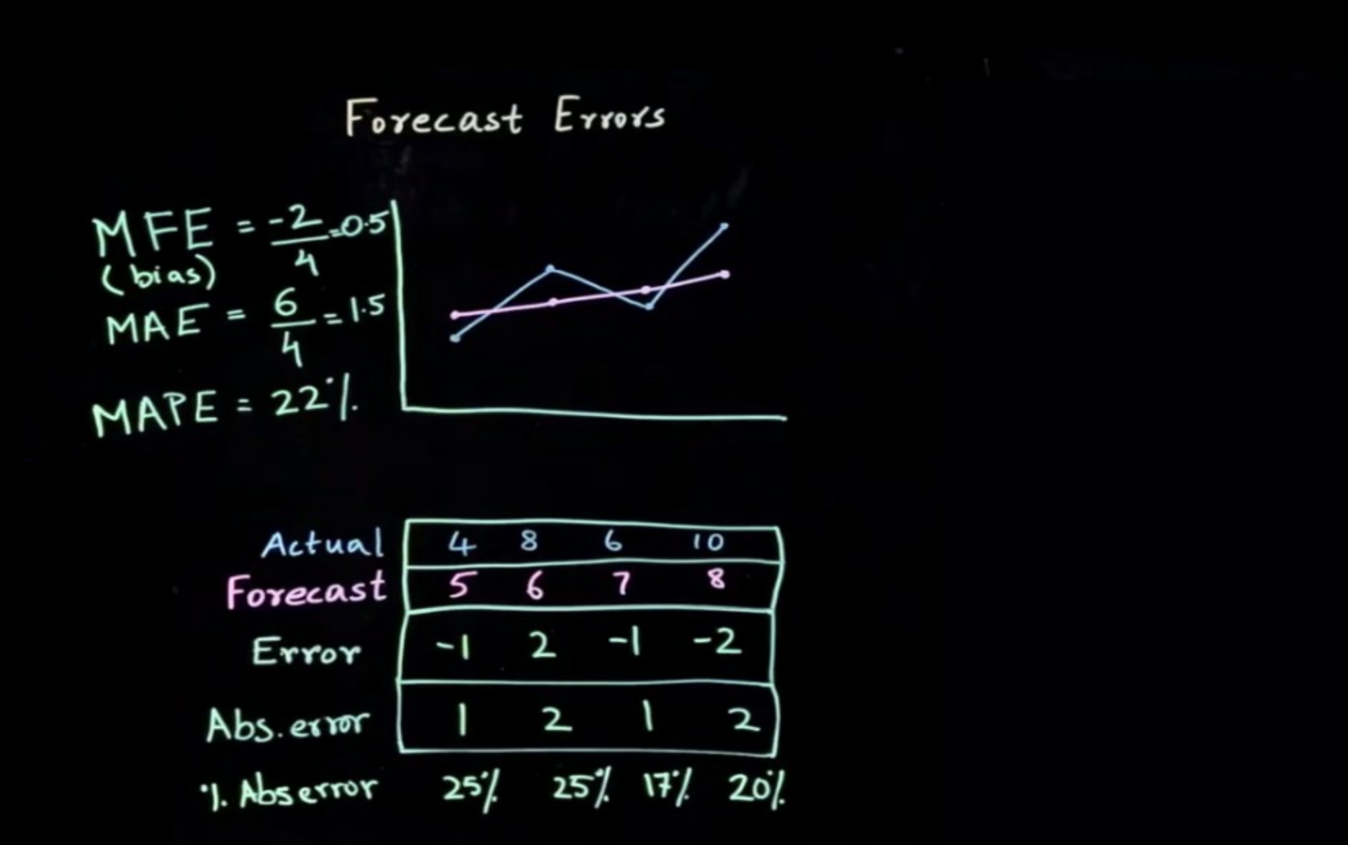
Chart, line chart

Description automatically generated

But suppose seasonality is a function of the level and the difference between the troughs of the time series data increases as you progress in the graph, then you use the multiplicative method.

# **Popular error measures**

Popular error measures are:-



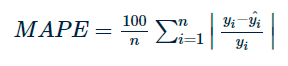
**Mean Forecast Error (MFE):** In this naive method, you simply subtract the actual values of the dependent variable, i.e., 'y' with the forecasted values of 'y'. This can be represented using the equation below.



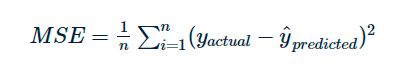
**Mean Absolute Error (MAE):** Since MFE might cancel out a lot of overestimated and underestimated forecasts, hence measuring the mean absolute error or MAE makes more sense as in this method, you take the absolute values of the difference between the actual and forecasted values.

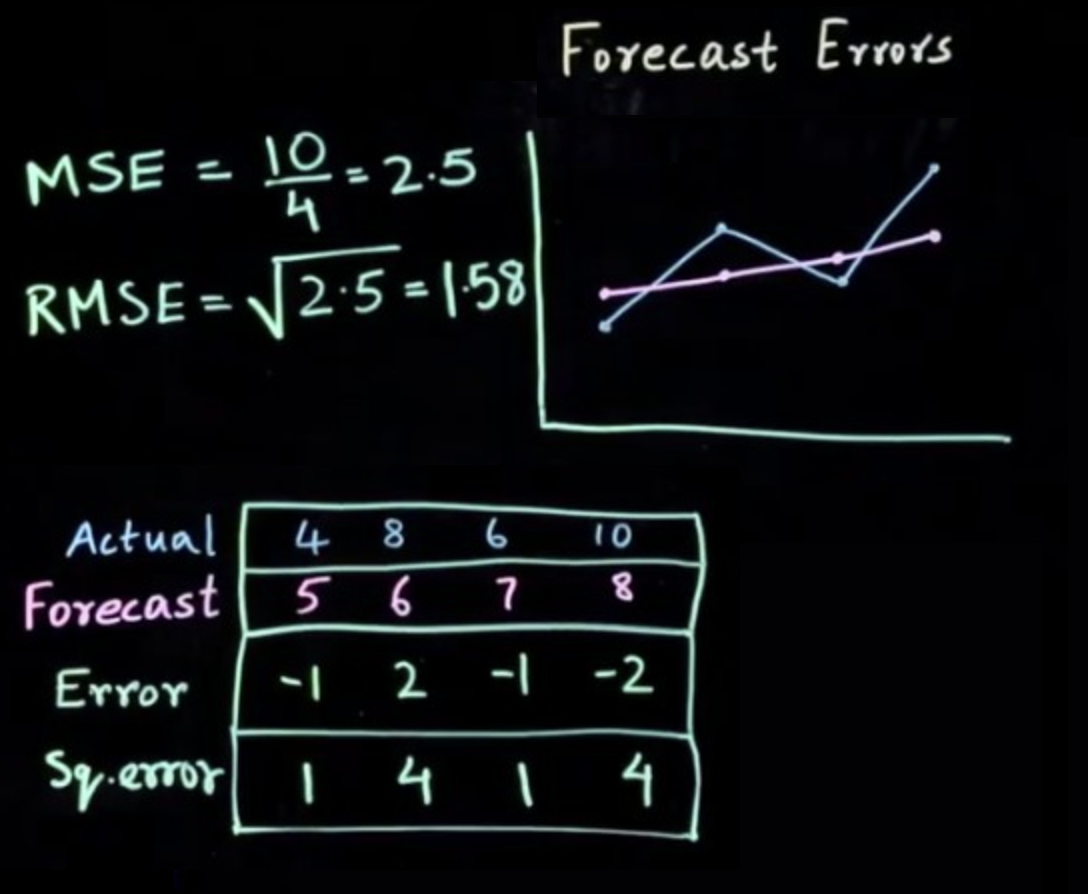


**Mean Absolute Percentage Error (MAPE):** The problem with MAE is that even if you get an error value, you have nothing to compare it against. For example, if the MAE that you get is 1.5, you cannot tell just on the basis of this number whether you have made a good forecast or not. If the actual values are in single digits, this error of 1.5 is obviously high but if the actual values are, say in the order of thousands, an error of 1.5 indicates a good forecast. So in order to capture how the forecast is doing based upon the actual values, you evaluate mean absolute error where you take the mean absolute error (MAE) as the percentage of the actual values of 'y'.

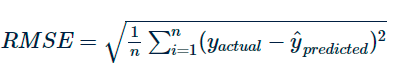


**Mean Squared Error (MSE):** The idea behind mean squared error is the same as mean absolute error, i.e., you want to capture the absolute deviations so that the negative and positive deviations do not cancel each other out. In order to achieve this, you simply square the error values, sum them up and take their average. This is known as mean squared error or MSE which can be represented using the equation below.





**Root Mean Squared Error (RMSE):**Since the error term you get from MSE is not in the same dimension as the target variable 'y' (it is squared), you deploy a metric known as RMSE wherein you take the square root of the MSE value obtained.



# **Cross-validation**

Basically cross-validation in time series is not done in the same way you might have done for any of the classical machine learning algorithms. This caveat stems from the fact that order matters in time series. So while building a forecast model, the test dataset is always on the right-hand side of the train dataset. **Let's understand the two types of validation**:-

### **One-Step Validation**

The testing set is just one step ahead to the training set. So suppose out of 15 data points, you decided to keep the first 10 of them as 'train' and the next 5 of them as 'test'. Now, the data points in the test set will be taken one-by-one starting from the left since you need all the previous values to predict the future values. So firstly, you take the 11th point to be the test set.

Once this value is successfully forecasted, you move on to the 12th point which is now your new test point and so on until you forecast for all the 5 points. This idea is represented in the image below. Here the blue squares represent the train data, the **yellow squares represent the test data**, and the **purple ones are the future values** for which the initial test points need to be predicted first. In the image below, you have 7 test data points, hence it takes 7 iterations (or forecasts) to fully predict the test set.

Diagram

Description automatically generated with medium confidence

### **Multi-Step Validation**

This is the same as one-step validation, the only difference being that you do not consider a few points to the immediate right of the last training datapoint but rather skip a few of these points to make forecasts well into the future. This can be seen in the image below.

Table

Description automatically generated with medium confidence

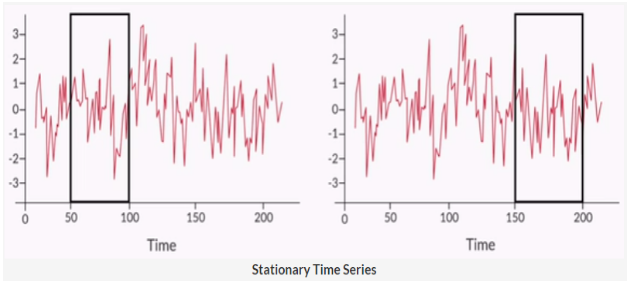
# **Assumptions to build an autoregressive model**

 Two fundamental assumptions to build an autoregressive model. They are -

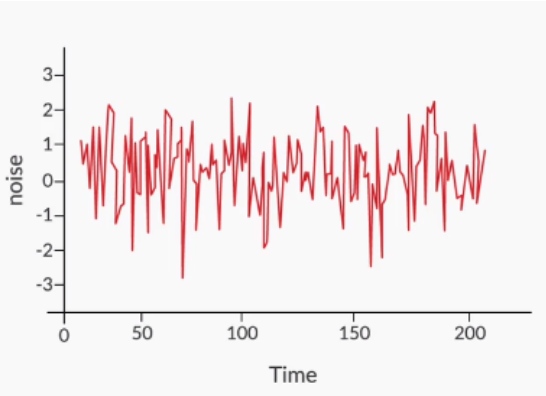
* Stationarity
* Autocorrelation

If a time series is stationary, its statistical properties like mean, variance, and [covariance](https://www.investopedia.com/terms/c/covariance.asp) will be the same throughout the series, irrespective of the time at which you observe them.

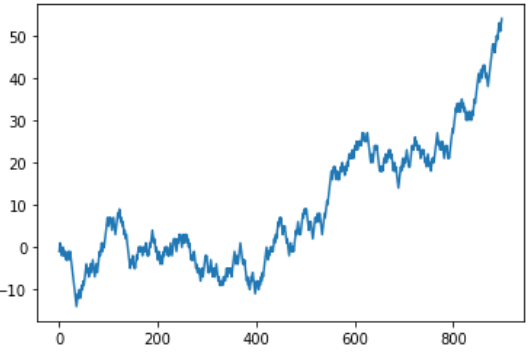
The following image illustrates a stationary time series in which the properties such as mean, variance and covariance are the same for any two-time windows.



White Noise is an example of a stationary time series with purely random, uncorrelated observations with no identifiable trend, seasonal or cyclical components.



Notice that there are no identifiable trends, seasonal or cyclical components. So a white noise series is basically an example of a stationary series.



A random walk is a time series where the current observation is equal to the previous observation plus a random change. Here, variance increases over time resulting in a non-stationary series.

In general, a stationary time series will have no long-term predictable patterns such as trends or seasonality. Time plots will show the series to roughly have a horizontal trend with constant variance.

Stationary processes are easier to analyze and model because their statistical properties remain constant over time. There will be no trend, seasonality and cyclicity in the series. In other words, if the past observations and future observations follow the same statistical properties i.e. there are no change in mean, variance and covariance then the future observation can be easily predicted.

In stationary series, the statistical properties like mean, variance and covariance remain the same, irrespective of the time window. A stationary series will be free of the trend and will have constant variance.A stationary time series has a higher accuracy of prediction as the future statistical properties will not be different from those currently observed.

You have studied two formal tests for stationarity based on hypothesis testing. The unit root test is a common procedure to determine whether a time series is stationary or not.

**Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test**

Null Hypothesis (H0): The series is stationary

p−value>0.05

Alternate Hypothesis (H1): The series is not stationary

p−value≤0.05

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

Null Hypothesis (H0): The series is not stationary

p−value>0.05

Alternate Hypothesis (H1): The series is stationary

p−value≤0.05

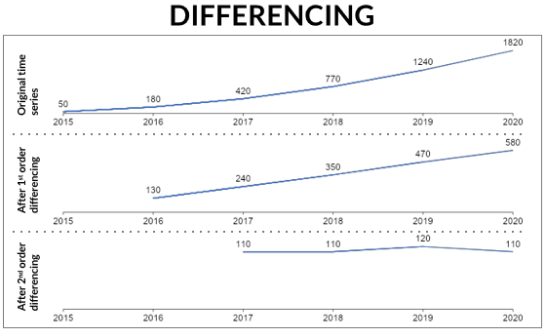
# **Making nonstationary data to stationary**

To remove the trend (to make the mean constant) in a time series you use the technique called **differencing**.

As the name suggests, in differencing you compute the differences between consecutive observations. Differencing stabilizes the mean of a time series by removing changes in the level of a time series and therefore eliminating (or reducing) trend and seasonality.

|  |  |  |
| --- | --- | --- |
| **Original time series** | **After 1st order differencing** | **After 2nd order differencing** |
| 50 |  |  |
| 180 | 130 |  |
| 420 | 240 | 110 |
| 770 | 350 | 110 |
| 1240 | 470 | 120 |
| 1820 | 580 | 110 |

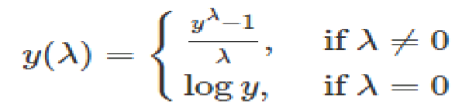
Here 1st order differencing is calculated by the difference of two consecutive observations of the original time series. 2nd order differencing is calculated by the difference of two consecutive observations of the 1st order differenced series.



At the first difference level, it still has a linear trend, but we have successfully removed the higher-order trend observed in the original time series. After the 2nd difference level, the series obtained is mostly a horizontal line. There is no overall visible trend in this series and thus you can say the series obtained is a stationary series.

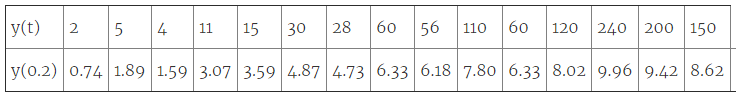
The other method to introduce the stationarity is making the variance constant. There can be many transformation methods used to make a non-stationary series stationary but here, we are discussing the Box-Cox transformation.

The mathematical formulae that Box-cox transformation is:

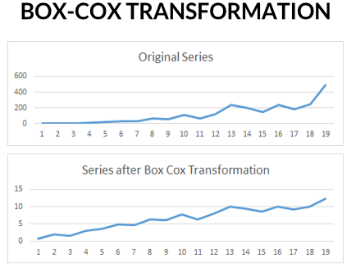


where y is the original time series and y(λ) is the transformed series. The procedure for the Box-Cox transformation is to find the optimal value of λ between -5 and 5 that minimizes the variance of the transformed data

Below is the time series data on which Box-Cox transformation is implemented with λ = 0.2.



Here we clearly see after the Box-Cox transformation the variance has become more or less constant or stabilized but still has a trend. Again, this trend can be removed by differencing.



two fundamental requirements to build an autoregressive model i.e. stationarity and autocorrelation.

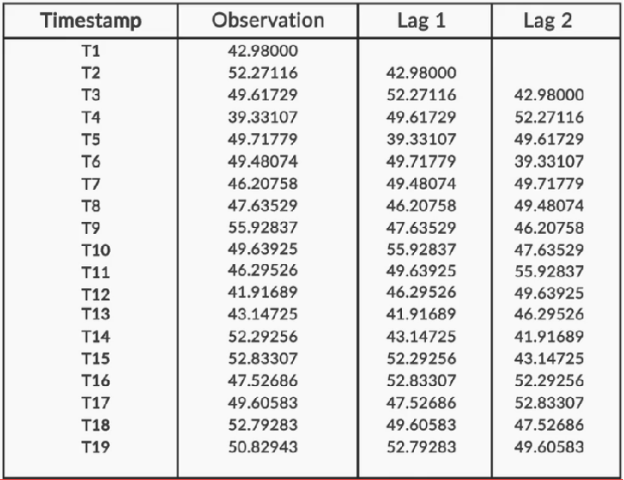
# **ACF and PACF**

**Autocorrelation**

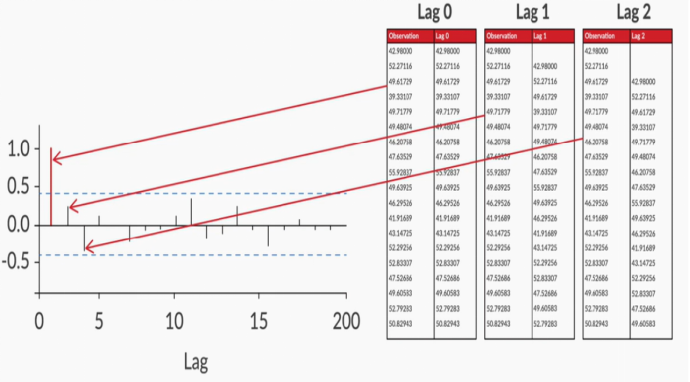
Autocorrelation is capturing the relationship between observations yt at time t and  yt−k at time k time period before t. In simpler words, autocorrelation helps us to know how a variable is influenced by its own lagged values. We will look at two Autocorrelation measures here:

* Autocorrelation function (ACF)
* Partial autocorrelation function (PACF)

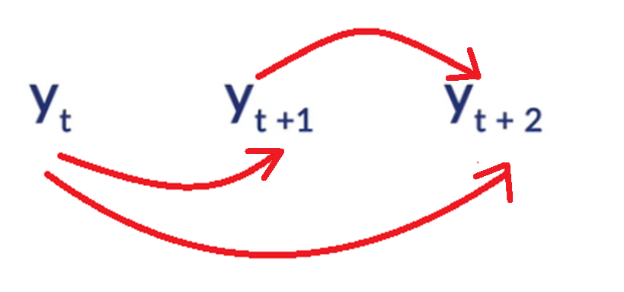
 The autocorrelation function tells about the correlation between an observation with its lagged values. It helps you to determine which lag of the observation is influencing it the most.



In the above example, we see the time series with lag 1 and lag 2 of its original time series. Now let’s see the autocorrelation plot of the same data.



we clearly see that the current observation is significantly correlated with lag 1 and lag 2. The other interesting thing to notice is that the autocorrelation function captures both direct and indirect relationships between the variables. For example:



**For** yt**,** yt+1**and**yt+2**:** Autocorrelation function captures both direct and indirect relationship with its lagged values. Here, the big arrow on the bottom indicates the direct relationship that is captured between yt and yt+2.

Autocorrelation function also captures the indirect relationship between yt and  yt+2 through yt+1. In simpler words, yt will have some correlation with yt+1, and yt+1 will also have a correlation with yt+2. This transitive correlation that passes through yt+1 is the indirect relationship which is also captured by the Autocorrelation function.

Thus, you can’t differentiate out only the direct relationship using ACF. To capture only direct relationships, you have another measure called Partial Autocorrelation Function or PACF.



A partial autocorrelation function captures only the direct relationship between an observation and its lagged value with the relationships of intervening observations removed.

**For**yt**,**yt+1**and**yt+2**:** Partial autocorrelation function captures the direct relationship between yt and yt+2 and does not capture indirect relationship between yt and yt+2 passing through yt+1.

# **The Simple Auto Regressive Model (AR)**

Two basic requirements to build an Auto Regressive model, Stationarity and Autocorrelation. In this segment and the next session, few Auto Regressive models that you will be studying are as follows:

* Auto Regressive (AR)
* Moving Average (MA)
* Auto Regressive Moving Average (ARMA)
* Auto Regressive Integrated Moving Average (ARIMA)
* Seasonal Auto Regressive Integrated Moving Average(SARIMA).
* Seasonal Auto Regressive Integrated Moving Average with Exogenous variable (SARIMAX).

In this session we will cover the Simple Auto Regressive model and the Moving Average model.

The Simple Auto Regressive model predicts the future observation as linear regression of one or more past observations. In simpler terms, the simple autoregressive model forecasts the dependent variable (future observation) when one or more independent variables are known (past observations). This model has a parameter **‘p’** called **lag order**. Lag order is the maximum number of lags used to build ‘p’ number of past data points to predict future data points.

**Example:**  Consider an example of forecasting monthly sales of ice cream for the year 2021 on the basis of the previous 3 years' monthly sales data of the ice cream. This can be one of the simple autoregressive model.  
**To determine the value of parameter ‘p’.**

Plot partial autocorrelation function

Select p as the highest lag where partial autocorrelation is significantly high

Here, the lag value of 1, 2, 4 and 12 has a significant level of confidence. i.e., significant level of influence on future observation (refer the red line). Hence, the value of 'p' will be set to 12 since that is the highest lag where partial autocorrelation is significantly high.

Build autoregression model equation:

3.3

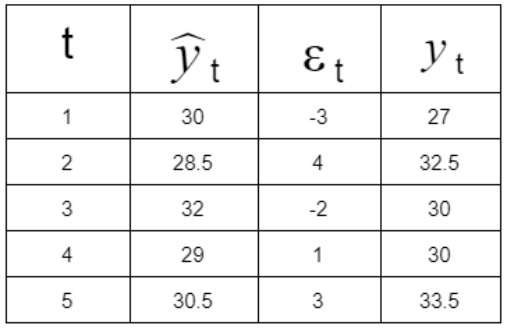
The past values which have a significant value are 1, 2, 4 and 12. Therefore, in the regression model the independent variables yt−1, yt−2, yt−4 and yt−12  which are the observations from the past has been taken to predict the dependent variable ^yt.

 Now let us get back again to the airline passenger dataset and build an AR model on it.  In order to build the AR model, the stationary series is divided into train and test data.

# **Moving Average Model (MA)**

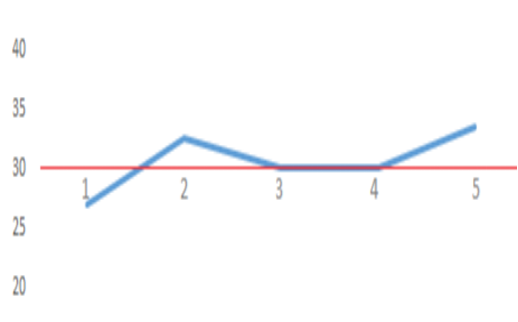
The Moving Average Model models the future forecasts using past forecast errors in a regression-like model. This model has a **parameter ‘q’** called **window size** over which linear combination of errors are calculated

**Example:**Forecast daily ice cream sales for the next 5 days when on an average daily ice cream sale is 30.



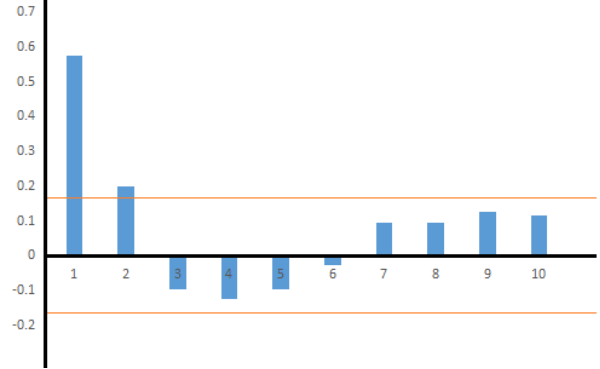
3.6

**Explanation:** On the first day you predicted the sale to be the average sale that is 30 but actual demand came to be around 27. There is an error of -3. For the next day, you predicted the sale of ice cream to be mean along with a percentage of error of previous day prediction. i.e., 30+0.5(-3) = 28.5. Similarly, you calculate the forecast for the rest of the days. The prediction of sales of ice cream is moving around the overall mean 30. For this reason, the model is called the moving average (MA) model. We can look at the following plot to confirm this.



Now, in order to build the moving average model, you  need to determine the value of parameter ‘q’. Let's follow the steps below to do that.

Plot the Autocorrelation function (ACF)



Select q as the highest lag beyond which autocorrelation dies down: ​​​​​​

Here, for lag 1 and lag 2 the autocorrelation is above significance level. Select q=2 as it is the highest lag beyond which autocorrelation dies down.

Build Moving Average model equations as:

3.8

Here, lag 1 and lag 2 have autocorrelation above the significance level. Therefore, in the moving average model, the errors with lag 1 and lag 2 are taken to predict the dependent variable ^yt.

Two assumptions of Auto Regressive models are:-

* Stationarity
* Auto Regression

ADF and KPSS tests to detect stationarity. The two methods to convert a non-stationary series into a stationary series.

1. Box-Cox Transformation makes the ***variance constant***
2. Differencing helps in ***removing the trend*** in the series by making the mean constant.

The second assumption of Auto Regressive models that is Auto Regression. There are two Auto Regression measures

1. Autocorrelation Function (ACF) and 2) Partial Autocorrelation Function (PACF).
2. ACF captures both the direct and indirect relationship of a time series variable with its lagged values
3. PACF captures only the direct relationship between a time series variable and its lagged values.

The Simple Auto Regressive model predicts future observation as a linear regression of one or more past observations. You learned the method to find the parameter 'p' of the AR(p) model by looking at the PACF plot.

The next Autoregressive model is Moving Average. The Simple Moving Average model predicts future observation as a linear regression of its past errors. You learned the method to find the parameter 'q' of the MA(q) model by looking at the ACF plot.

# **Auto Regressive Moving Average (ARMA)**

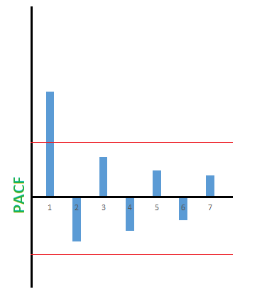
An Auto Regressive Model consisting of both 'AR' and 'MA' components called Auto Regressive Moving Average (ARMA). A time series that exhibits the characteristics of an AR(p) and/or an MA(q) process can be modelled using an ARMA(p,q) model.

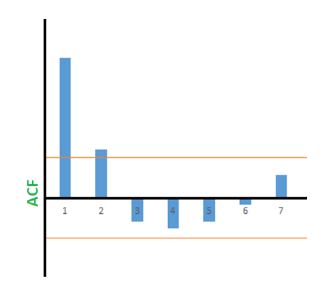
ARMA (1,1) equation: For p = 1 and q=1 —  ^y=β0+β1St−1+ϕ1εt−1

Here, ^y is forecasted value.

 To determine the parameters ‘p’ and ‘q’ —

Plot autocorrelation function (ACF) and partial autocorrelation function (PACF)





If you check the plot for PACF, you will see that you need to select p = 1 as the highest lag where partial autocorrelation is significantly high.

Similarly, from the ACF plot, select q = 2 as the highest lag beyond which autocorrelation dies down.So, we would select an ARMA (1, 2) model in this example.

As per the concepts, to find p, plot the partial autocorrelation functions, and to estimate q, plot the autocorrelation function.

# **Auto Regressive Integrated Moving Average (ARIMA)**

Steps of ARIMA model

● Original time series is differenced to make it stationary

● Differenced series is modeled as a linear regression of

○ One or more past observations

○ Past forecast errors

● ARIMA model has three parameters

○ p: Highest lag included in the regression model

○ d: Degree of differencing to make the series stationary

○ q: Number of past error terms included in the regression model

○ Here the new parameter introduced is the ‘I’ part called integrated.

It removes the trend (non-stationarity) and later integrates the trend to the original series.

So if you think about it, ARIMA is nothing different from what you have done so far. Initially you applied both the boxcox transformation and differencing in order to covert the data into a stationary time-series data. Here, you are just applying boxcox before building the model and letting the model take care of the differencing i.e. the trend component itself.

Let's now quickly revisit the equations. ARIMA(1,1,1) Equations:

Text

Description automatically generated

Here, Zt is the first order differencing for time series.

To determine the parameters ‘p’, ‘d’ and ‘q’

● For ‘d’: Select d as the order of difference required to make the original time series stationary. We can verify if this differenced series is stationary or not by using the stationarity tests: ADF or KPSS test.

● For ‘p’ and ‘q’: Plot ACF and PACF of the 1st order differenced time series. Find the value of ‘p’ and ‘q’ as discussed previously with the earlier Auto Regressive Models.

● The last step in the ARIMA model is to recover the original time series forecast.

# **Seasonal Auto Regressive Integrated Moving Average (SARIMA)**

SARIMA brings all the features of an ARIMA model with an extra feature - seasonality.

The non-seasonal elements of SARIMA

● Time series is differenced to make it stationary.

● Models future observation as linear regression of past observations and past forecast errors. The seasonal elements of SARIMA

● Perform seasonal differencing on time series.

● Model future seasonality as linear regression of past observations of seasonality and past forecast errors of seasonality. Example: To forecast quarterly ice cream sales of 2020 using the Quarterly ice cream sales data for the last 4 years.

Chart, line chart

Description automatically generated

**Explanation:**

● Historical Sales follow a quarterly seasonality and in four years we get 16 data points.

● Future sales is related to past sales with lag = 4

● Sales over the last 4 years is steadily increasing The above three points make clear that this example has a seasonal component.

The parameters ‘p’, ‘d’, ‘q’ and ‘P’, ‘D’, ‘Q’:

● Non-seasonal elements

○ p: Trend autoregression order

○ d: Trend difference order

○ q: Trend moving average order

● Seasonal elements

○ m: The number of time steps for a single seasonal period

○ P: Seasonal autoregressive order

○ D: Seasonal difference order

○ Q: Seasonal moving average order Equation for SARIMA(1,0,0)(0,1,1)4:

Text

Description automatically generated

**SARIMAX**

has three components:

● Non-seasonal elements

○ Models future observation as a linear regression of past observations and past forecast errors.

○ Performs differencing to make time-series stationary.

● Seasonal elements ○ Models seasonality as the linear regression of past observations and past forecast errors from previous seasons.

○ Perform seasonal differencing to make time-series stationary over seasons.

● Exogenous variable

○ Models future observations as linear regression of an external variable.

Example: Forecast quarterly ice cream sales for 2020 when you have the data of quarterly sales of the ice cream for the last 4 years.

Chart, line chart

Description automatically generated

In the above time plot, sales data displays trend and seasonality but peaks are not periodic. The peak in 2017 has appeared in the first quarter, while in 2018 it has appeared in the 2nd quarter and again it has appeared in the first quarter of 2018. Thus, it is evident that peaks, in spite of being close to summer, are not periodic.

This irregularity in the peak is due to an extra factor, maybe the promotions. The respective months when a promotion is done, there is an extra sale leading to a peak in the quarter. You can visually observe this here:

Chart, line chart

Description automatically generated

If we know this promotion period for the quarter, it will help us to forecast the sales more accurately. Equations: SARIMAX(1,0,0)(0,1,1)4

Text, letter

Description automatically generated

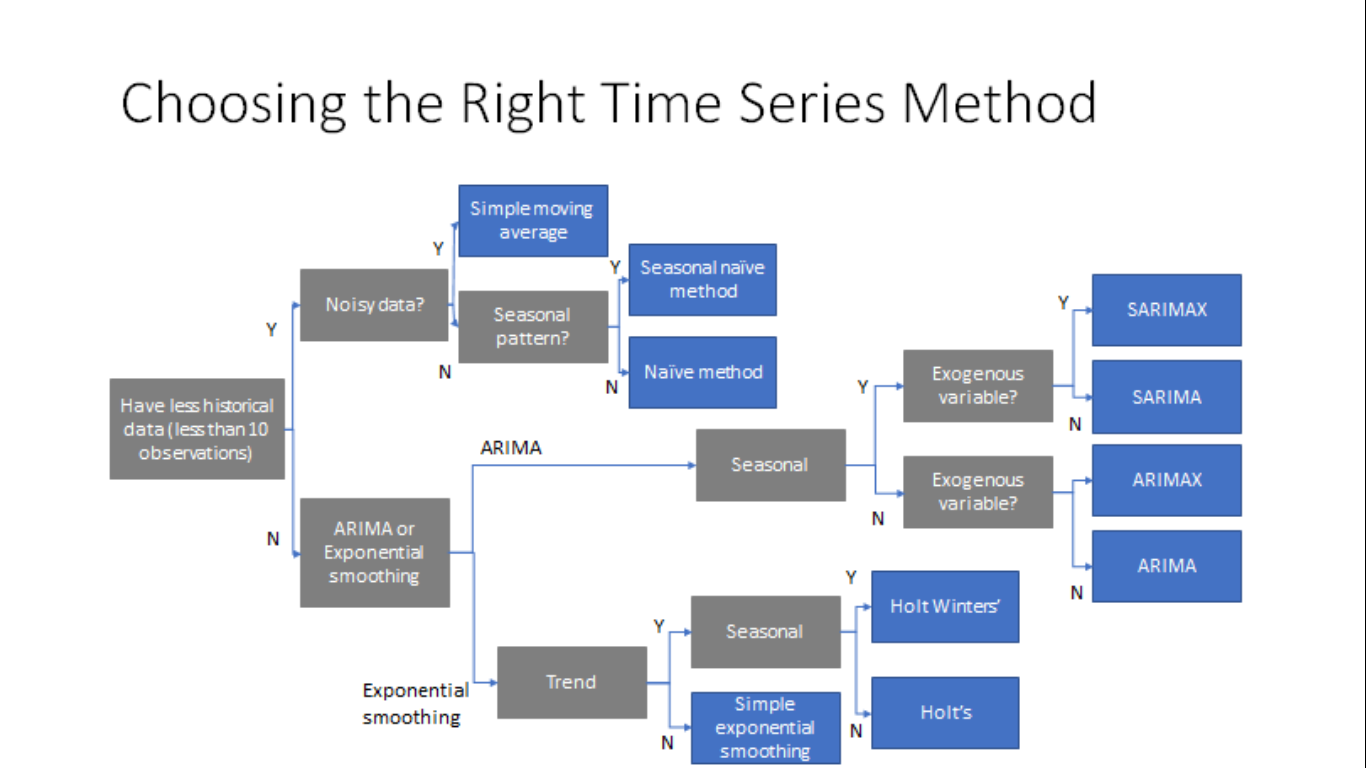
The parameters ‘p’, ‘d’, ‘q’ and ‘P’, ‘D’, ‘Q’ will be the same as SARIMA(p,d,q)(P,D,Q)m.

● Determining parameter values ○ PACF plots to determine non-seasonal ‘p’ value ○ ACF plots to identify non-seasonal ‘q’ value

● Use stationarity tests to determine the value 'd'

● Use grid search to choose optimal seasonal P, D and Q parameter values

# **Choosing The Right Time Series Method**



When you have time-series data of fewer than 10 observations that are noisy, you should use a simple moving average method because it helps cancel out the noise to some extent. This method does not work well if the data has a seasonal component or more than 10 observations.

An example in which a simple moving average method works well is the daily forecasting of stock prices. This is because the number of observations is fewer and the data is noisy. Thus, the simple moving average method is able to predict the forecast better since it takes the variation of very few data points.

If the data points are fewer than 10 but the data is neither noisy nor has any seasonal pattern, then the Naive method works well because it will forecast the next values based on the previous values of the train data. In the case of a higher number of observations(generally more than 10), the forecast tends to underpredict or overpredict the values.

Thus, the naive method works better when there are fewer than 10 observations. An example of the usage of the naive method would be in forecasting the sales of a grocery store that opened recently. In this case, the sales are not much dependent on any seasonal component, and thus, the naive method can be used here

in the case of non-noisy data and seasonality with data points fewer than 10, the seasonal naive method works well. An example in which the seasonal naive method works well is forecasting the sales of a newly opened store that sells umbrellas on a monthly basis.

In the case of more than 10 data points, for using a technique among the smoothing/ARIMA techniques, you need to keep the following points in mind:

1. To capture the level in time series data, the simple exponential smoothing technique is used. An example in which the simple exponential smoothing technique is used is the forecasting of the annual GDP of a developed country. In this case, there may or may not be an obvious trend in the data and the seasonal component is absent. Thus, using a simple exponential smoothing technique makes sense here.

2. Holt's exponential smoothing technique / ARIMA method works best in capturing both the level and the trend in the time series data. An example in which Holt's exponential smoothing technique/ARIMA method can be used is when there is an obvious trend in the data but no specific seasonality exists. For example, the sales data for iPhones for the last 10–12 years show an increasing trend but no specific seasonality; thus, the Holt’s method / ARIMA method can be used to forecast the time series data.

3. Now, to capture the level, the trend and seasonality, the Holt-Winters’ exponential smoothing technique / SARIMA works best. However, this method should be used when the dataset has no exogenous variables. An example in which the techniques mentioned here can be used is in determining the number of monthly visitors to an amusement park. With increased marketing, the number of visitors to the amusement park keeps increasing and thus, the data shows a trend. However, these numbers show a seasonal pattern. For example, during the summer and winter vacation months, visitors tend to flock to the amusement park even more, which implies that the dataset has a seasonality component.

4. The ARIMAX/SARIMAX method works best for capturing the level, the trend and the seasonality in time series data when some exogenous variables are present. The ARIMAX method may not capture the seasonality but the SARIMAX method does. An example of using these methods is in determining the monthly sales of an e-commerce website. Here, the sales are affected by numerous external factors and thus, the ARIMAX/SARIMAX method will work well here.