# **ARIMA Family of Forecasting Techniques**

## **Auto Regression (AR)**

* For stationary time series, one is helpless because of the absence of a pattern in the series.
* Any trend or seasonality that was present has been removed from the series, and it will be added back later, in the final prediction.
* In AR​​ we forecast the variable of interest using a linear combination of past values of the variable.
* For such cases we study a new family of models called **ARIMA**.
* Though such series look like they are completely random, there is still some extent of forecastability here, there is still information left to be extracted from stationary series.

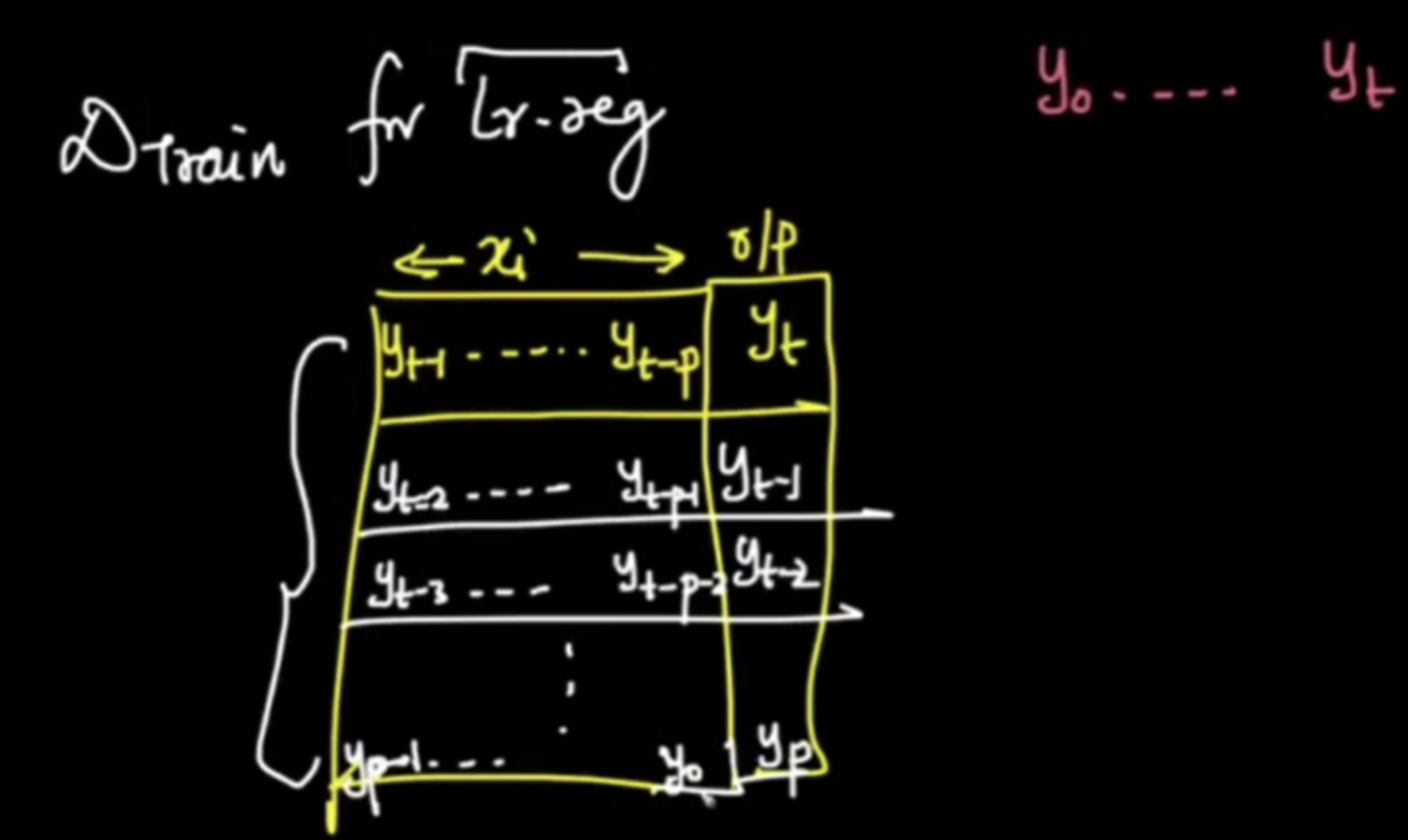
**What if we had a feature in our time series, besides the value to be predicted?**

* In that case, one could just utilize Linear Regression, by mapping this feature's values with the value to be forecasted.
* For creating a new feature, we can map the value of the stationary time series at time

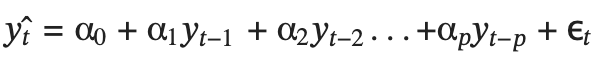
𝑡 with the value of series at time 𝑡−1,𝑡−2,𝑡−3,...,𝑡−𝑝, where p could be a **hyperparameter** we set.

* This way, now our data becomes as shown.
* It contains date as the index,
* Past values 𝑦𝑡−1, 𝑦𝑡−2, ..., 𝑦𝑡−𝑝 as features and
* value at time 𝑡 as the value to be predicted (𝑦^t)

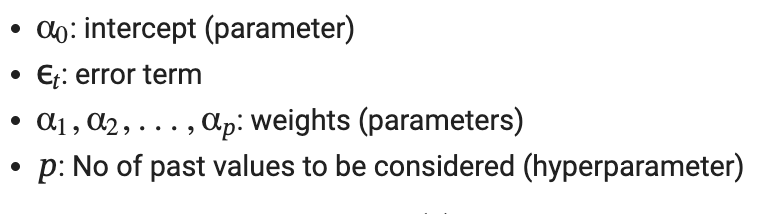
Now we can successfully implement Linear Regression using these features.



* Since the aim is to convert the forecasting problem to Linear Regression, what we’re doing is;
  + Future value 𝑦̂𝑡  = LinearRegression(Past p values)
* Thus, the forecasting problem is converted into the following form:



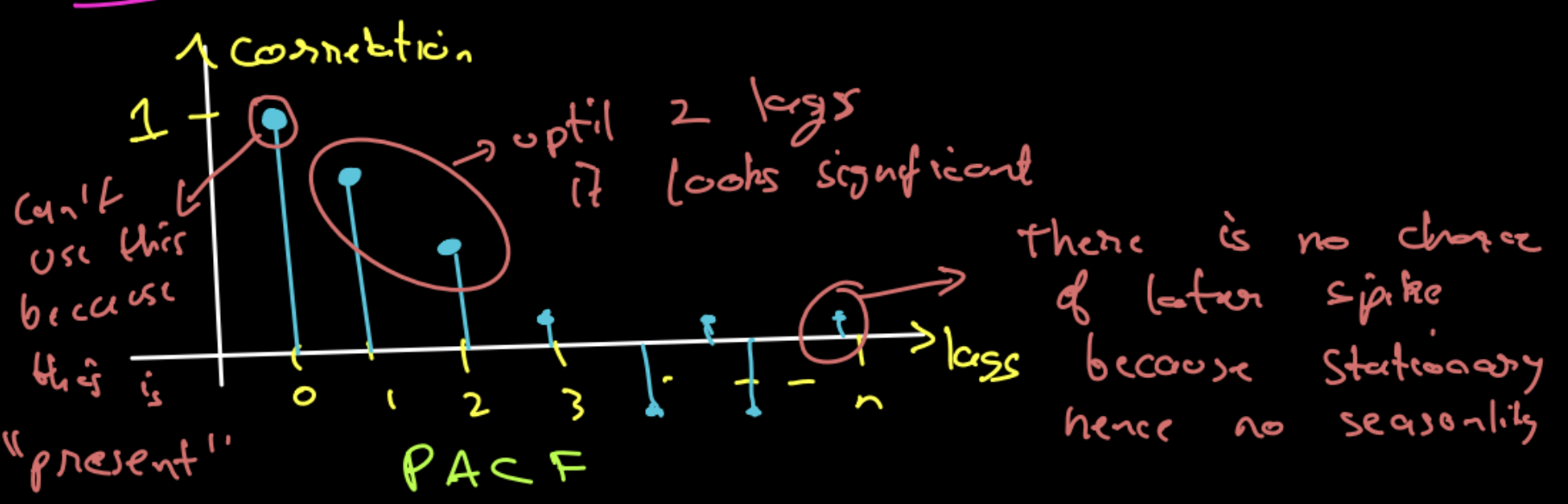
where;



* is a residual term considered as a purely random series with mean 0, variance and covariance is 0.
* This seems similar to Simple Exponential Smoothing (SES).
* Though we are essentially doing a **weighted average** of the past time series values in both SES and AR models, there is a fundamental difference.
  + In the case of SES,
    - The weights are **exponentially decaying**
    - The hyperparameter is **α**
  + In the case of **AR**,
    - The weights are **learnt** by multiple iterations.
    - The hyperparameter is 𝑝
* **Pre-requisites of the AR model:**
  + The idea is that this assumption will be true if the Partial Auto Correlation(PACF) Plot has a high value at lag = k.
  + This way, from the plot we know that the future value is highly correlated with one value in the past, which means that it makes sense for us to compute linear regression on the past 1 value.
  + We are using PACF because we don't want features to be correlated with each other in LR. PACF can help to identify that

**Deciding the value of p**

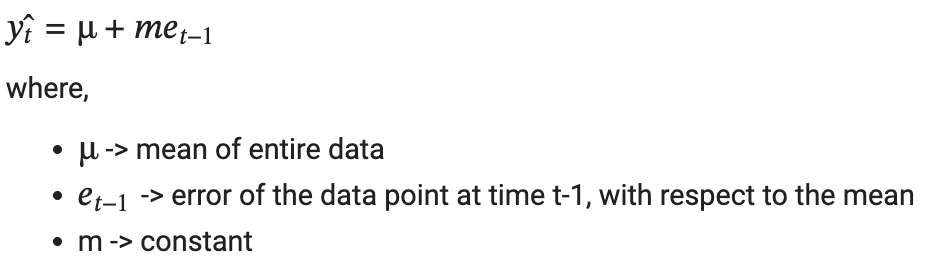
* We look at the PACF plot, and based on the values there, we decide how many lag values we can consider.
* For example, in the given plot,
  + we consider 2 lags only,
  + as for the third lag, the PACF value is 0.1, so considering it will not give as promising results.



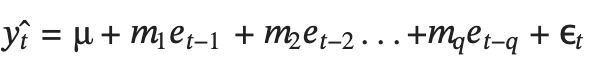
* The simplest AR process is AR(0), which has no dependence between the terms. In fact, AR(0) is essentially white noise.
* We can select the order 𝑝 for AR(𝑝) model based on significant spikes from the PACF plot.
* One more indication of the AR process is that the ACF plot decays more slowly.

## **Moving Averages (MA)**

* The idea here is to use the value of the series at time t-1, we use the error of value at t-1 from the mean of data in the regression setup
* The error of each data point in the series from the mean/average, would be different.
* This should also work, as we are able to successfully create a new feature that is unique for each point.
* In fact, this idea is called the **Moving Averages (MA)** technique.
* Moving Average process considers the past residual values to predict the current time period values.
* One problem of the AR model is the ignorance of correlated noise structures (which are unobservable) in the time series.
* Contrary to the AR model, the finite MA model is always stationary because the observation is just a weighted moving average over past forecast errors.
* Though the name is the same as the smoothing technique, this has nothing to do with that and is a completely different concept.
* The formulation of MA is as follows:



* This idea can be extended to the order of **q.** In that case, the formulation becomes:



Here, **ε** represents the final error remaining that is actually truly random, which we cannot help. This is also added for representation.

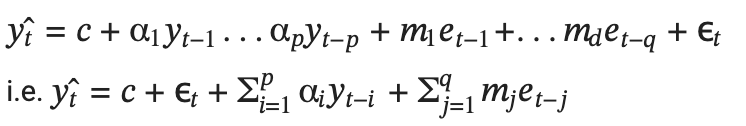
**NOTE:**

* q becomes the **hyperparameter** for the MA(q) model
* In the case of MA, there is a fixed way to determine the value of q, we need to try a bunch of different values to find the best fit.
* ACF provides a considerable amount of information about the order of the dependence q for the MA(q) process.
* Identification of an MA model is often best done with the ACF rather than the PACF.
* In contrast to the AR model, we can select the order q for model MA(q) from ACF if this plot has a sharp cut-off after lag q.
* One more indication of the MA process is that the PACF plot decays more slowly.

## 

## **Auto Regression - Moving Averages (ARMA)**

* The combined technique of Auto Regression (AR) and Moving Averages (MA) is called the ARMA model.
* While combining the two ideas, **p**: order of AR and **q**: order of MA, **p** may or may not be equal to **q**
* α1, α2,..., α𝑝: coefficients of AR
* 𝑚1, 𝑚2,..., 𝑚𝑞: coefficients of MA
* Hence the formulation becomes:



* Here, **p** and **q** are hyperparameters. Thus it is also called **ARMA(p,q)** model.
* The major limitation of this technique is that it cannot handle non-stationary time series because, if we're training a Linear Regression, the variables can not be dependent on each other.
* This method can not handle if there is a trend or seasonality present in the data.
* For the AR term refer to the PACF plot which tells about the lag terms and for the MA term refer to the ACF plot which tells about the error terms.

|  | **AR(p)** | **MA(q)** | **ARMA(p,q)** |
| --- | --- | --- | --- |
| **ACF Plot** | Slow Decay | Sharp Cutoff | Sharp Cutoff after lag 1 |
| **PACF Plot** | Sharp Cutoff | Slow Decay | Sharp Cutoff after lag 1 |

**Limitations of ARMA:**

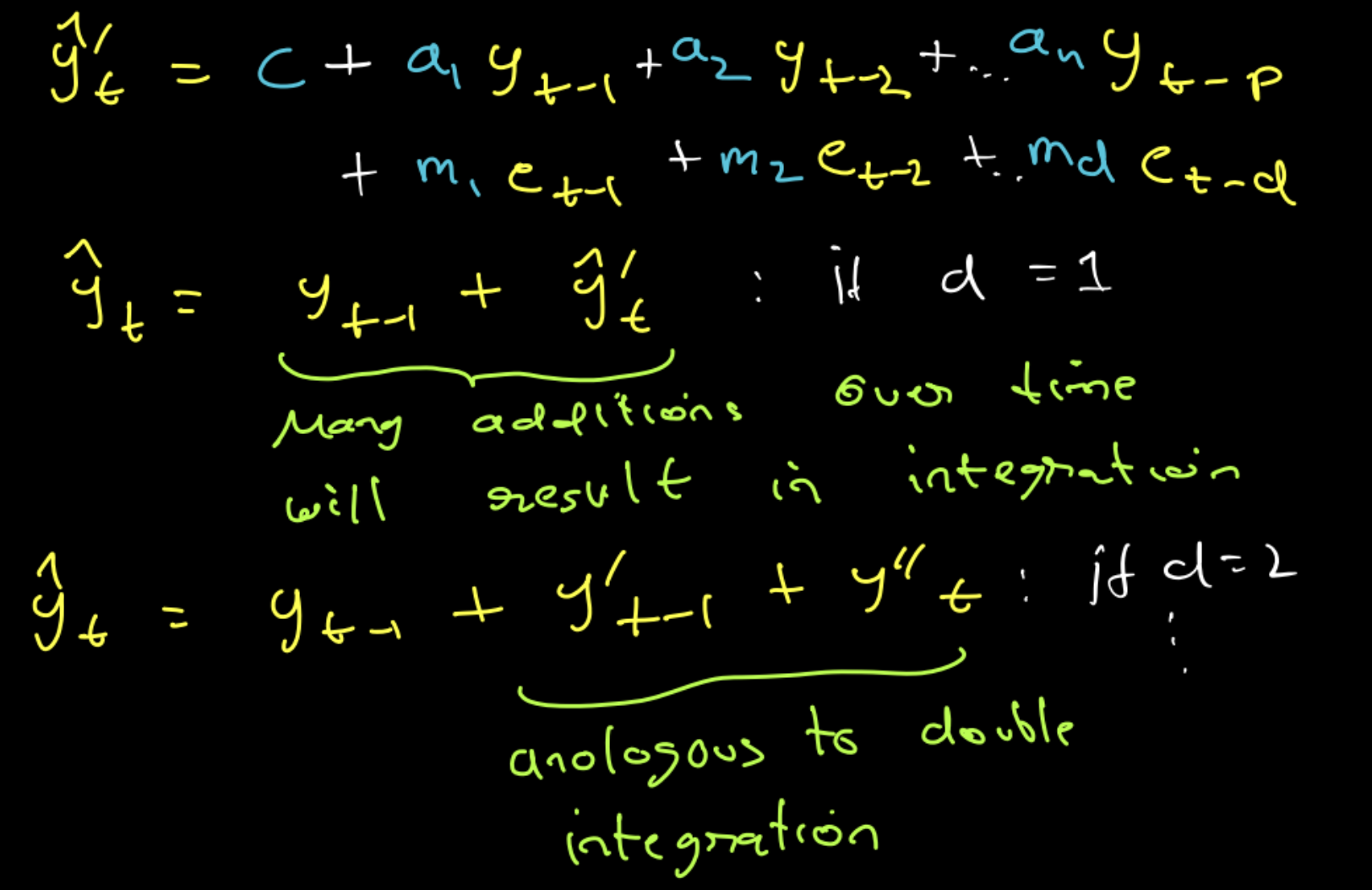
* Cannot handle non-stationary series and we cannot handle seasonal lags.
* Differencing causes the loss of one data point. If the order is 1 and in the case of the differencing the series the value will be NA.
* After the differencing the scale of the series will be changed. So, the forecast will also be on a different scale and we need to manually adjust the by retransforming by doing integration.

## **ARIMA**

* If a time series is not stationary,
  + We perform **differencing** to de-trend
  + Then we apply the ARMA technique, to get an approximation.
  + Now to get a good forecast, we need to **integrate** the trend **back** to get the final result
* This technique is called **ARIMA**
* Instead of us manually doing integrations (and note that it can get very hard when you need to double / tripple differentiate), we can simply use the ARIMA model, which does this job for us.
* ARIMA model is denoted as ARIMA(p,d,q)
  + Where,
    - p - order of the autoregressive part.
    - d - degree of first differencing involved.
    - q - order of the moving average part.

**Formulation of ARIMA**

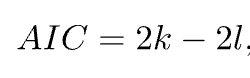
* ARIMA can be formulated as a summation of:
  + Differencing of order d
    - d=1 for linear trends, d=2 for quadratic trends, ...
  + Auto Regression of order p
  + Moving Averages of order q
  + Getting the resulting forecasts
  + Integrating d times to restore the trend back



* We need to try some combinations of p and q parameters and compare results using a validation set.
* In order to find the best combination of p and q, we need to have some objective function that will measure model performance on a validation set.
* The high-level logic behind that is the same as the logic behind hyperparameter tuning of any other machine learning model. We need to try some combinations of p and q parameters and compare results using a validation set.
* Since our search space is not big, usually values p and q are not higher than 10, we can apply a popular technique for hyperparameter optimization called grid search.
* We can also use AIC and BIC for that purpose. The lower the value of these criteria, the better the model is.

**AIC (Akaike Information Criteria )**

* AIC stands for Akaike Information Criteria, and it’s a statistical measure that we can use to compare different models for their relative quality.
* It measures the quality of the model in terms of its goodness-of-fit to the data, its simplicity, and how much it relies on the tuning parameters.
* The formula for AIC is



* where,
  + l is a log-likelihood
  + k is a number of parameters.
* For example, the AR(p) model has p+1 parameters. From this formula, we can conclude that AIC prefers a higher log-likelihood that indicates how strong the model is in fitting the data and a simpler model in terms of parameters.
* Limitation of ARIMA is that it fails to capture the seasonality in the series.

# **SARIMA**

* If we wish to account for seasonality, the process becomes:-

1. differentiating (𝑥[ 𝑖 ] − 𝑥[ 𝑖−𝑇 ]), to remove seasonality
2. perform AR and MA
3. integrating the seasonality back

* This is very tiring.
* Instead of doing so much work to utilize the ARIMA model, we can just apply another model called the **SARIMA model** directly.
* There are 7 parameters for SARIMA: **P, Q, D, p, q, d, s**

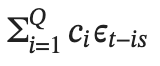
## **Hyperparameter s**

* The parameter s represents **seasonality**
  + This value we can find using **ACF and PACF plots**.
  + Alternately, we can treat s as a **hyperparameter** and tune it to get the best value.
* In case of lag of order 1, at time t, translates to 𝑦𝑡−1
* For example,
  + For a normal time series, a lag of order of 2 means 𝑦𝑡−1 and 𝑦𝑡−2
    - And this makes sense as if we're in March 2022, then order of 2 means, lag of 1, i.e. February 2022 and lag of 2, i.e. January 2022
  + In case of yearly seasonality, if we're in March 2022, a lag of order 2, means March 2021, and March 2020.
  + Therefore, here, an order of m gets translated as 𝑦𝑡−𝑠, 𝑦𝑡−2𝑠,..., 𝑦𝑡−𝑚𝑠.
* There is one **problem** with the SARIMA Model in terms of capturing seasonality.
* **We can only use one value of seasonality.**

## **Hyperparameter P**

* When we set the order of AR as p, the formulation becomes:



* **The effect of setting the `P` hyperparameter for SARIMA is: **
  + Since our seasonality is 12
  + Essentially, P enables in creating of an AR Model on data that is 12 months old, 24 months old, ..., 12P months old: 𝑦𝑡−12, 𝑦𝑡−24,...,𝑦𝑡−12𝑃.
  + So this is exactly like an AutoRegression, but with seasonality.
* Suppose we wish to forecast y^100 (i.e. t=100) with the following hyperparameter values:-
* 𝑝=4
  + Contribution to final forecast:

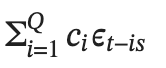


* s=12
* P=3
  + Contribution to final forecast:



## **Hyperparameter Q**

* Similarly, the effect of setting the Q hyperparameter for SARIMA is:



* Suppose if seasonality in a certain time series data is 12.
* Essentially, Q enables in creating of an MA Model on data that is 12 months old, 24 months old, ..., 12Q months old: 𝑦𝑡−12, 𝑦𝑡−24,..., 𝑦𝑡−12𝑄.
* So this is exactly like a Moving Average, but with seasonality.
* Suppose we wish to forecast 𝑦̂100 with the following hyperparameter values:-
  + q = 4
    - Contribution to final forecast:

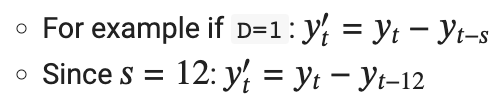


* + s = 12
  + Q = 3
    - Contribution to final forecast:



## **Hyperparameter D**

* Recall that hyperparameter d performs differencing on the time series d times, before applying the model.
  + For example if d=1: 𝑦′𝑡 = 𝑦𝑡 − 𝑦𝑡−1
* Similarly, D helps in doing **seasonal differencing** on the time series.



**So, to sum it up SARIMA can be formulated as:**

𝑦̂t+1 = p(AR term) + q(MA term) + d(differencing) + D(differencing) + P(AR-Seasonality) + Q(MA - Seasonality)

* The parameters p, q, d represent the non-seasonal part of the model
* P, Q, D, s represent the seasonal part of the model.
* The seasonal part of an AR or MA model will be seen in the seasonal lags of the PACF and ACF.
* For example, an ARIMA(0,0,0)(0,0,1)12 the model will show:
* a spike at lag 12 in the ACF but no other significant spikes;
* exponential decay in the seasonal lags of the PACF (i.e., at lags 12, 24, 36, …).
* Similarly, an ARIMA(0,0,0)(1,0,0)12 the model will show:
* exponential decay in the seasonal lags of the ACF.
* a single significant spike at lag 12 in the PACF.

# 

# 

# **SARIMAX Model**

* We can utilize the exogenous variable and incorporate it into our SARIMA Model.
* This idea of incorporating exogenous variables into SARIMA gave rise to a new model: **SARIMAX Model**
* Here, the X represents an exogenous variable
* The only difference between SARIMA and SARIMAX is that here, we can incorporate exogenous variables into the calculations of our forecasts.
* Exogenous variables are assigned a weight, say 𝑤𝑖.
* We don't need to initialize this, it is learned and trained by SARIMAX, and taken care of under the hood.
* This is done in addition to the SARIMAX operations.
* The SARIMAX model takes exogenous variables into account
  + i.e. variables measured at time 𝑡 that influence the value of our time series at time 𝑡, but that are not autoregressive.
* To do this, we simply add the terms on the right-hand side of our ARIMA and SARIMA equations.

