Time series Analysis

Stationarity of Time series --

A TS is said to be stationary if its **statistical properties** such as mean, variance remain **constant over time**.

But why is it important? Most of the TS models work on the assumption that the TS is stationary. Intuitively, we can say that if a TS has a particular behaviour over time, there is a very high probability that it will follow the same in the future. Also, the theories related to stationary series are more mature and easier to implement as compared to non-stationary series.

Stationarity is defined using very strict criterion. However, for practical purposes we can assume the series to be stationary if it has constant statistical properties over time, ie. the following:

1. constant mean
2. constant variance
3. an autocovariance that does not depend on time.

We can check stationarity using the following:

1. **Plotting Rolling Statistics:** We can plot the moving average or moving variance and see if it varies with time. By moving average/variance I mean that at any instant ‘t’, we’ll take the average/variance of the last year, i.e. last 12 months. But again this is more of a visual technique.
2. **Dickey-Fuller Test:** This is one of the statistical tests for checking stationarity. Here the null hypothesis is that the TS is non-stationary. The test results comprise of a **Test Statistic** and some **Critical Values** for difference confidence levels. If the ‘Test Statistic’ is less than the ‘Critical Value’, we can reject the null hypothesis and say that the series is stationary.

**How to make a Time Series Stationary?**

Though stationarity assumption is taken in many TS models, almost none of practical time series are stationary. So statisticians have figured out ways to make series stationary, which we’ll discuss now. Actually, it’s almost impossible to make a series perfectly stationary, but we try to take it as close as possible.

Let’s understand what is making a TS non-stationary. There are 2 major reasons behind non-stationarity of a TS:  
1. **Trend** – varying mean over time. For e.g., in this case we saw that on average, the number of passengers was growing over time.  
2. **Seasonality** – variations at specific time-frames. E.g. people might have a tendency to buy cars in a particular month because of pay increment or festivals.

SMAPE is used to evaluate Forecast Accuracy

Dickey Fuller Test used to test stationarity

## ARMA Time Series Modeling

ARMA model, AR stands for auto-regression and MA stands for moving average.

**AR or MA are not applicable on non-stationary series**.

### Auto-Regressive Time Series Model

**x(t) = alpha \*  x(t – 1) + error (t)**

### Moving Average Time Series Model

**x(t) = beta \*  error(t-1) + error (t)**

### Difference between AR and MA models

The primary difference between an AR and MA model is based on the correlation between time series objects at different time points. The correlation between x(t) and x(t-n) for n > order of MA is always zero. This directly flows from the fact that covariance between x(t) and x(t-n) is zero for MA models. However, the correlation of x(t) and x(t-n) gradually declines with n becoming larger in the AR model. This difference gets exploited irrespective of having the AR model or MA model. The correlation plot can give us the order of MA model.

ARIMA

ARIMA stands for **Auto-Regressive Integrated Moving Averages**. The ARIMA forecasting for a stationary time series is nothing but a linear (like a linear regression) equation.  It is a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration.

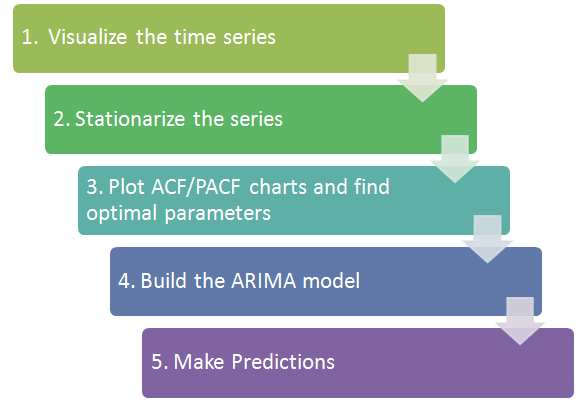
This acronym is descriptive, capturing the key aspects of the model itself. Briefly, they are:

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

Each of these components are explicitly specified in the model as a parameter. A standard notation is used of ARIMA (p,d,q) where the parameters are substituted with integer values to quickly indicate the specific ARIMA model being used.

The parameters of the ARIMA model are defined as follows:

* **p**: The number of lag observations included in the model, also called the lag order.
* **d**: The number of times that the raw observations are differenced, also called the degree of differencing.
* **q**: The size of the moving average window, also called the order of moving average.



## Configuring an ARIMA Model

The classical approach for fitting an ARIMA model is to follow the [Box-Jenkins Methodology](https://en.wikipedia.org/wiki/Box%E2%80%93Jenkins_method).

This is a process that uses time series analysis and diagnostics to discover good parameters for the ARIMA model.

In summary, the steps of this process are as follows:

1. **Model Identification**. Use plots and summary statistics to identify trends, seasonality, and autoregression elements to get an idea of the amount of differencing and the size of the lag that will be required.
2. **Parameter Estimation**. Use a fitting procedure to find the coefficients of the regression model.
3. **Model Checking**. Use plots and statistical tests of the residual errors to determine the amount and type of temporal structure not captured by the model.

The process is repeated until either a desirable level of fit is achieved on the in-sample or out-of-sample observations (e.g. training or test datasets).

The **SARIMA** model (Seasonal ARIMA) extends the ARIMA by adding a linear combination of seasonal past values and/or forecast errors.

PROPHET

[Prophet](https://github.com/facebook/prophet), or “Facebook Prophet,” is an open-source library for univariate (one variable) time series forecasting developed by Facebook.

Prophet implements what they refer to as an [additive time series forecasting model](https://en.wikipedia.org/wiki/Additive_model), and the implementation supports trends, seasonality, and holidays.

Prophet is optimized for the business forecast tasks encountered at Facebook, which typically have any of the following characteristics:

* hourly, daily, or weekly observations with at least a few months (preferably a year) of history
* strong multiple “human-scale” seasonalities: day of week and time of year
* important holidays that occur at irregular intervals that are known in advance (e.g. the Super Bowl)
* a reasonable number of missing observations or large outliers
* historical trend changes, for instance due to product launches or logging changes
* trends that are non-linear growth curves, where a trend hits a natural limit or saturates

We have frequently used Prophet as a replacement for the **forecast** package in many settings because of two main advantages:

1. **Prophet makes it much more straightforward to create a reasonable, accurate forecast.** The forecast package includes many different forecasting techniques (ARIMA, exponential smoothing, etc), each with their own strengths, weaknesses, and tuning parameters.
2. **Prophet forecasts are customizable in ways that are intuitive to non-experts.** There are smoothing parameters for seasonality that allow you to adjust how closely to fit historical cycles, as well as smoothing parameters for trends that allow you to adjust how aggressively to follow historical trend changes. For growth curves, you can manually specify “[capacities](https://en.wikipedia.org/wiki/Logistic_function#Time-varying_carrying_capacity)” or the upper limit of the growth curve, allowing you to inject your own prior information about how your forecast will grow (or decline). Finally, you can specify irregular holidays to model like the dates of the Super Bowl, Thanksgiving and Black Friday.

## Exponential smoothing

In its basic form it is called **simple exponential smoothing** and its forecasts are given by:

Ŷ(t+h|t) = ⍺y(t) + ⍺(1-⍺)y(t-1) + ⍺(1-⍺)²y(t-2) + … with 0<⍺<1.

We can see that forecasts are equal to a weighted average of past observations and**the corresponding weights decrease exponentially** as we go back in time.

## GARCH

The previous models assumed that the error terms in the stochastic processes generating the time series were **homoskedastic**, i.e. with constant variance.

Instead, the **GARCH**model assumes that the variance of the error terms follows an AutoRegressive Moving Average (ARMA) process, therefore allowing it to change in time. It is particularly useful for modelling financial time series whose volatility changes across time. The name is an acronym for Generalised Autoregressive Conditional Heteroskedasticity.

In [econometrics](https://en.wikipedia.org/wiki/Econometrics), the autoregressive conditional heteroscedasticity (ARCH) model is a [statistical model](https://en.wikipedia.org/wiki/Statistical_model) for [time series](https://en.wikipedia.org/wiki/Time_series) data that describes the [variance](https://en.wikipedia.org/wiki/Variance) of the current [error term](https://en.wikipedia.org/wiki/Errors_and_residuals_in_statistics) or [innovation](https://en.wikipedia.org/wiki/Innovation_(signal_processing)) as a function of the actual sizes of the previous time periods' error terms;often the variance is related to the squares of the previous [innovations](https://en.wikipedia.org/wiki/Innovation_(signal_processing)). The ARCH model is appropriate when the error variance in a time series follows an [autoregressive](https://en.wikipedia.org/wiki/Autoregressive) (AR) model; if an [autoregressive moving average](https://en.wikipedia.org/wiki/Autoregressive_moving_average_model) (ARMA) model is assumed for the error variance, the model is a generalized autoregressive conditional heteroskedasticity (GARCH) model.[[2]](https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity#cite_note-GARCH_1986-2)

ARCH models are commonly employed in modeling [financial](https://en.wikipedia.org/wiki/Mathematical_finance) [time series](https://en.wikipedia.org/wiki/Time_series) that exhibit time-varying [volatility](https://en.wikipedia.org/wiki/Volatility_(finance)) and [volatility clustering](https://en.wikipedia.org/wiki/Volatility_clustering), i.e. periods of swings interspersed with periods of relative calm. ARCH-type models are sometimes considered to be in the family of [stochastic volatility](https://en.wikipedia.org/wiki/Stochastic_volatility) models, although this is strictly incorrect since at time *t* the volatility is completely pre-determined (deterministic) given previous values.

## NNETAR

The NNETAR model is a fully connected neural network. The acronym stands for Neural NETwork AutoRegression.

The NNETAR model takes in**input the last elements of the sequence** up to time t and outputs the forecasted value at time t+1. To perform multi-steps forecasts the network is applied iteratively.

In presence of seasonality, the input may include also the seasonally lagged time series.

## TBATS

The TBATS model is a forecasting model based on exponential smoothing. The name is an acronym for Trigonometric, Box-Cox transform, ARMA errors, Trend and Seasonal components.

The main feature of TBATS model is its capability to deal with **multiple seasonalities** by modelling each seasonality with a trigonometric representation based on Fourier series. A classic example of complex seasonality is given by daily observations of sales volumes which often have both weekly and yearly seasonality.

Important links:

https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb