**Time Series Data & Python**

--------------------------------------

Time series is a sequence of observations recorded at regular time intervals.Depending on the frequency of observations, a time series may typically be hourly, daily, weekly, monthly, quarterly and annual. Sometimes, you might have seconds and minute-wise time series as well, like, number of clicks and user visits every minute etc.

Analyze a time series is imp as it is the preparatory step before you develop a forecast of the series.

Time series analysis involves understanding various aspects about the inherent nature of the series so that you are better informed to create meaningful and accurate forecasts.

A time series typically stores in .csv files or other spreadsheet formats and contains two columns:

the date and the measured value.

**Option 1:**

Note\* When creating a dataframe from your dataset

Adding the parse\_dates=['date'] argument will make the date column to be parsed as a date field.

df = pd.read\_csv('https://raw.githubusercontent.com/ajaykuma/PythonCodes/master/SampleFiles/a10.csv', parse\_dates=['date'])

df.head()

**Option 2:**

Import it as a pandas Series with the date as index. You just need to specify the index\_col argument in the pd.read\_csv()

ser = pd.read\_csv('https://raw.githubusercontent.com/ajaykuma/PythonCodes/master/SampleFiles/a10.csv', parse\_dates=['date'], index\_col='date')

ser.head()

**Panel Data:**

Panel data is also a time based dataset.

The difference is that, in addition to time series, it also contains one or more related variables that are measured for the same time periods.

Typically, the columns present in panel data contain explanatory variables that can be helpful in predicting the Y,

provided those columns will be available at the future forecasting period.

dfP = pd.read\_csv('https://raw.githubusercontent.com/ajaykuma/PythonCodes/master/SampleFiles/MarketArrivals.csv')

dfP = dfP.loc[dfP.market=='MUMBAI', :]

**Visualizing TS data:**

--Seasonal plot for TS Data

--Boxplot of Month-wise (Seasonal) and Year-wise (trend) Distribution

**Patterns in a time series:**

Any time series may be split into the following components:

Base Level + Trend + Seasonality + Error

Trend is observed when there is an increasing or decreasing slope observed in the time series.

Seasonality is observed when there is a distinct repeated pattern observed between regular intervals due to seasonal factors.

It could be because of the month of the year, the day of the month, weekdays or even time of the day.

It is not mandatory that all time series must have a trend and/or seasonality. A time series may not have a distinct trend but have a seasonality. The opposite can also be true.

Another aspect to consider is the **Cyclic behaviour**. It happens when the rise and fall pattern in the series does not happen in fixed calendar-based intervals.

Care should be taken to not confuse ‘cyclic’ effect with ‘seasonal’ effect.

**Difference between a ‘cyclic’ vs ‘seasonal’ pattern?**

If the patterns are not of fixed calendar based frequencies, then it is cyclic. Because, unlike the seasonality, cyclic effects are typically influenced by the business and other socio-economic factors.

**Additive and multiplicative time series**

Depending on the nature of the trend and seasonality, a time series can be modeled as an additive or multiplicative,

wherein, each observation in the series can be expressed as either a sum or a product of the components:

Additive time series:

Value = Base Level + Trend + Seasonality + Error

Multiplicative Time Series:

Value = Base Level x Trend x Seasonality x Error

**Decompose a time series into its components:**

Do a classical decomposition of a time series by considering the series as an additive or multiplicative combination of the base level, trend, seasonal index and the residual.

**seasonal\_decompose in statsmodels implements this**

https://www.statsmodels.org/dev/tsa.html

Setting extrapolate\_trend='freq' takes care of any missing values in the trend and residuals at the beginning of the series.

If you look at the residuals of the additive decomposition closely & it has some pattern left over in comparision to the multiplicative decomposition which looks quite random then latter should be preferred for particular series.

#refer code

**Stationary and Non-Stationary Time Series**

Stationarity is a property of a time series. A stationary series is one where the values of the series is not a function of time.

That is, the statistical properties of the series like mean, variance and autocorrelation are constant over time.

**Autocorrelation of the series is nothing but the correlation of the series with its previous values.**

**Why does a stationary series matter?**

It is possible to make nearly any time series stationary by applying a suitable transformation.

Most statistical forecasting methods are designed to work on a stationary time series. The first step

in the forecasting process is typically to do some transformation to convert a non-stationary series to stationary.

You can make series stationary by:

* Differencing the Series (once or more)
* Take the log of the series
* Take the nth root of the series
* Combination of the above

The most common and convenient method to stationarize the series is by differencing the series at least once until it becomes approximately stationary.

If Y\_t is the value at time ‘t’, then the first difference of Y = Yt – Yt-1.

In simpler terms, differencing the series is nothing but subtracting the next value by the current value.

If the first difference doesn’t make a series stationary, you can go for the second differencing. And so on.

For example, consider the following series: [1, 5, 2, 12, 20]

First differencing gives: [5-1, 2-5, 12-2, 20-12] = [4, -3, 10, 8]

Second differencing gives: [-3-4, -10-3, 8-10] = [-7, -13, -2]

Forecasting a stationary series is relatively easy and the forecasts are more reliable.

An important reason is, autoregressive forecasting models are essentially linear regression models that utilize the lag(s) of the series itself as predictors.

We know that linear regression works best if the predictors (X variables) are not correlated against each other.

So, stationarizing the series solves this problem since it removes any persistent autocorrelation, thereby making

the predictors(lags of the series) in the forecasting models nearly independent.

**To test for stationarity!**

The stationarity of a series can be established by looking at the plot of the series like we did earlier.

Another method is to split the series into 2 or more contiguous parts and computing the summary statistics like the mean, variance and the autocorrelation. If the stats are quite different, then the series is

not likely to be stationary.

You need a method to quantitatively determine if a given series is stationary or not. This can be done using statistical tests called ‘Unit Root Tests’. There are multiple variations of this, where the tests check if a time series is

non-stationary and possess a unit root.

There are multiple implementations of Unit Root tests like:

**Augmented Dickey Fuller test (ADH Test)**

**Kwiatkowski-Phillips-Schmidt-Shin – KPSS test (trend stationary)**

**Philips Perron test (PP Test)**

The most commonly used is the ADF test, where the null hypothesis is the time series possesses a unit root and is non-stationary. So, id the P-Value in ADH test is less than the significance level (0.05),

you reject the null hypothesis.

The KPSS test, on the other hand, is used to test for trend stationarity. The null hypothesis and the P-Value interpretation is just the opposite of ADH test.

#refer code

**Difference between white noise and a stationary series?**

Like a stationary series, the white noise is also not a function of time, that is its mean and variance does not change over time. But the difference is, the white noise is completely random with a mean of 0.

In white noise there is no pattern whatsoever. **If you consider the sound signals in an FM radio as a time series, the blank sound you hear between the channels is white noise**.

#refer code

**Detrend a time series?**

Detrending a time series is to remove the trend component from a time series.

There are multiple approaches.

--Subtract the line of best fit from the time series. The line of best fit may be obtained from a linear regression model with the time steps as the predictor. For more complex trends, you may want to

use quadratic terms (x^2) in the model.

* --Subtract the trend component obtained from time series decomposition we saw earlier.
* --Subtract the mean
* --Apply a filter like Baxter-King filter(statsmodels.tsa.filters.bkfilter) or the Hodrick-Prescott Filter

(statsmodels.tsa.filters.hpfilter) to remove the moving average trend lines or the cyclical components.

#refer code

**Deseasonalize a time series?**

Multiple approaches to deseasonalize a time series

- 1. Take a moving average with length as the seasonal window. This will smoothen in series in the process.

- 2. Seasonal difference the series (subtract the value of previous season from the current value)

- 3. Divide the series by the seasonal index obtained from STL decomposition

If dividing by the seasonal index does not work well, try taking a log of the series and then do the deseasonalizing.

You can later restore to the original scale by taking an exponential.

#refer code

**To test for seasonality of a time series?**

The common way is to plot the series and check for repeatable patterns in fixed time intervals. So, the types of seasonality is determined by the clock or the calendar:

Hour of day

Day of month

Weekly

Monthly

Yearly

However, if you want a more definitive inspection of the seasonality, use the Autocorrelation Function (ACF) plot.

when there is a strong seasonal pattern, the ACF plot usually reveals definitive repeated

spikes at the multiples of the seasonal window.

**To treat missing values in a time series!**

our time series might have missing dates/times. That means, the data was not captured or was not available for those periods. It could so happen the measurement was zero on those days, in which case, case you may fill up those periods with zero.

Secondly, when it comes to time series, **you should typically NOT replace missing values with the mean of the series**, especially if the **series is not stationary**. What you could do instead for a quick and dirty workaround is to forward-fill the previous value.

However, depending on the nature of the series, you want to try out multiple approaches before concluding. Some effective alternatives to imputation are:

* **Backward Fill**
* **Linear Interpolation**
* **Quadratic interpolation**
* **Mean of nearest neighbors**
* **Mean of seasonal couterparts**

To measure the imputation performance, you can manually introduce missing values to the time series, impute it with above approaches and then measure the mean squared error of the imputed against the actual values.

#refer code

Other Approaches

1. If you have explanatory variables use a prediction model like the random forest or k-Nearest Neighbors to predict it.
2. If you have enough past observations, forecast the missing values.
3. If you have enough future observations, backcast the missing values
4. Forecast of counterparts from previous cycles.

**Understanding autocorrelation and partial autocorrelation functions!**

Autocorrelation is simply the correlation of a series with its own lags. If a series is significantly autocorrelated, that means, the previous values of the series (lags) may be helpful in predicting the current value.

Partial Autocorrelation also conveys similar information but it conveys the pure correlation of a series and its lag, excluding the correlation contributions from the intermediate lags.

**#refer code**

**Lag Plots**

**To estimate the forecastability of a time series?**

**Why and How to smoothen a time series?**