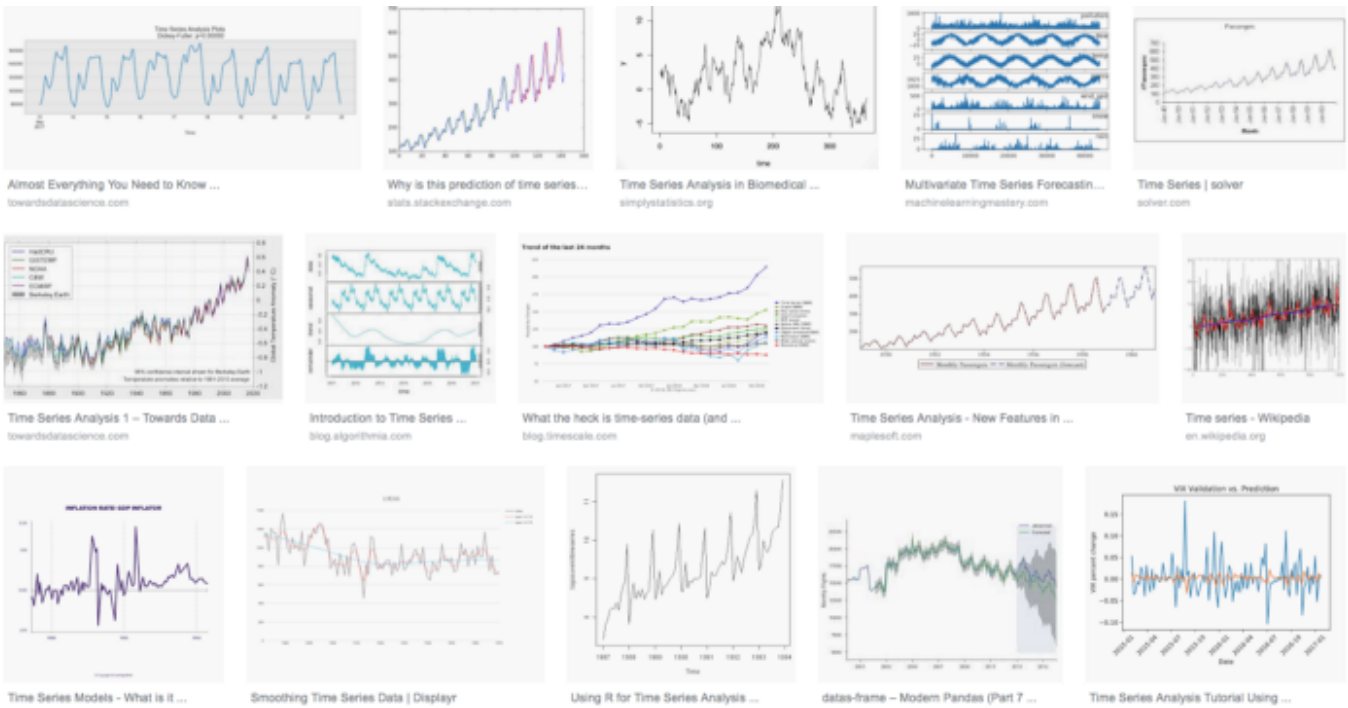
# Preprocessing for Time

Mehul Gupta Follow
Jul 3, 2019 · 5 min read




In the last couple of articles, we have discussed a lot about how to forecast Time Series — multivariate or univariate(links below!!). But Wait! **What about Preprocessing?**

Before reaching forecasting, we must understand how important is preprocessing for time series. It can make or break your forecasting. So let us go through some of the crucial preprocessing steps for time series —

- **First of all, cast your Date column in date datatype and set it as your index**. It might be the case that you might be provided with different date formats, Like 25/03/1997 or 1997/03/25 or 03/25/1997. How to interpret all these formats?

**Try strptime(x, format)** from DateTime library. Here 'x' is your sample to be cast & 'format' is the desired format.



**Example —**

if 25/03/1997, format='%d/%m/%Y'

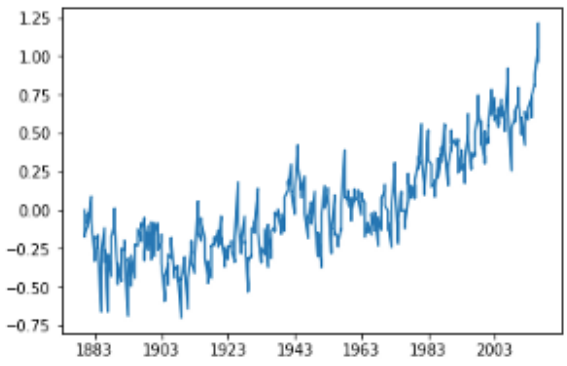
if 1997/03/25, format='%Y/%m/%d',e

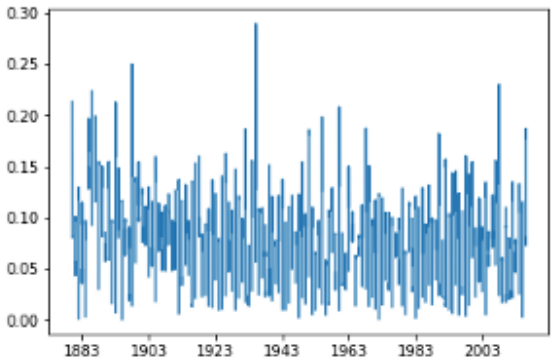
- Don't forget to sort your data according to the Date column/index.
- The most important step is to check whether the Time Series is stationary or not.

**Why? check here**

> *We need to have a constant mean and standard deviation, which can be checked using plotting mean & standard deviation for a rolling window as you can see below.*

```
In [7]: plt.plot(data1.rolling(window='30D').mean())
        plt.show()
        plt.plot(data1.rolling(window='30D').std())
        plt.show()
```



It shows that though the standard deviation is quite constant, rolling mean is not. If that too would have been constant, then we c
mean, we need to stop and apply some t
would be going through three methods f



- **Self Lag Differencing —** It can be ta
***series and a lagged version of the se***
For items where we don't have any lagged version item, take them as NULL.

**Example —** Let your data frame be 'Time' and column with values be 'Temperature' indexed on a date. So self differencing can be done like this-

Time['Temperature_Diff']=Time['Temperature']-Time['Temperature'].shift(1) if lagged version used is 1

Time['Temperature_Diff']=Time['Temperature']-Time['Temperature'].shift(2) if lagged version used is 2

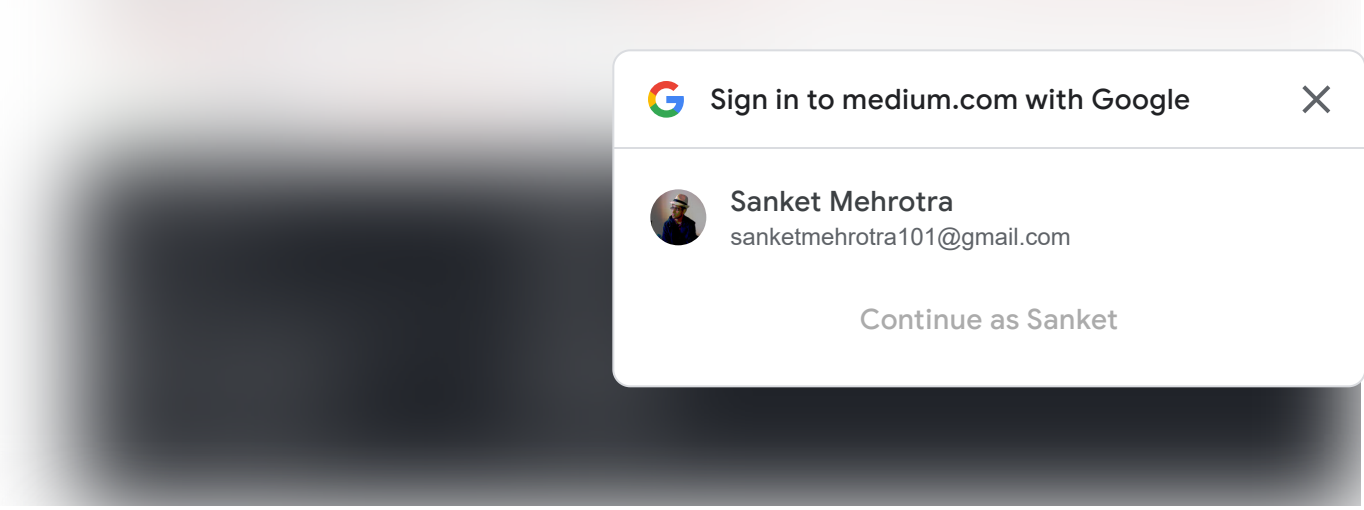
- **Log Self Differencing —** It can be taken as the difference between present series and a lagged version of the series. But you can just **apply log transformation** over the actual series.

And the best method to follow up is using:

```
from statsmodels.tsa.seasonal import seasonal_decompose
```

**Use seasonal_decompose and it will give you three components-Trend, Seasonality and Residuals.** Take these **residuals** and it will be our stationary time series for forecasting.

## If you don't find any abnormality in your rolling plots, give a final check using the ADFuller Test.




If you are not familiar with Hypothesis testing, **check here.**

Now moving with the mathematics of AD Fuller Test, let us understand the output for now and how it will help us in our task.

For an intuitive understanding of the AD Fuller Test, click here.

**Using and ADfuller(), we get 5 outputs. We will concentrate on Test statistics & the dictionary we get as the 5th output. Depending upon the significance level of the test, we will compare it with the statistic provided(against the critical value) and if the test statistic is below that, Series is stationary else not.**

*Like in the above illustration, the Test statistic e is greater for even significance level 10%, hence not stationary.* Some major characteristics of the AD Fuller test are —

- The time series has a unit root is the Null Hypothesis
- auto lag = 'AIC' is a criteria to choose from the max lagged versions of the series to use for the test. Auto log can take 'AIC','BIC' or if not set, takes the maxlags set by the user.

Now at this point in time, we have a stationary time series!!!

## Finally, we start the Forecasting.

**I would be going with ARIMA and how to tune its parameters.** A very important point is that we need to tune 3 parameters as explained in my previous articles (check below).

- P = Using AutoCorrelation plot for AR
- D = For Integrated term in AR-I-MA
- Q = Using PartialAutoCorrelation pl



For any terms you don't get, refer below



And here comes the crux —

Things you must note down from the above picture:

- The dotted lines represent the confidence interval(95%).
- For 95% confidence interval, z-score is +1.96,-1.96.
- Plot these intervals using the codes used above. It has been divided by the root of the total sample numbers. This has been done because when the number of samples is known, it is always a t-score we calculate and not z-score(there is no such rule that if less than 30 samples than only t-score else z-score.).

- Once plotted ACF & pacf, look for the first time the blue line crosses the dashed(confidence interval). For bo
something near 1 or 2. Hence P=1,2



***For D, we need to look at which lagged***
***stationary.if X — X.shift(1) makes you***

And our forecast is done!

Once you are done with forecasting, **don't forget to trace back all transformation you made to your original series**(most common mistake). Additionally, in the same reverse sequence(follow LIFO, Last applied transformation reverted first). Do create a copy of your series. It will help you out in recovering things back. For example, if you applied log() first and then differencing, first add what has been subtracted(using a backup copy I asked you to keep) and then exp() over that.

**Do check out my other Time-series articles-**

- **Basic Time-Series Terminologies**
- **Why Time Series has to be Stationary?**
- **AR & MA models for Time Series Forecasting**
- **Holt-Winters & Exponential Smoothing for Time Series Forecasting**
- **Multivariate Time Series Forecasting**
- **Intuitive explanation for ADFuller Test**
- **Tokenization algorithms in NLP**
- **Reinforcement Learning Basics (5 parts)**
- **How to get your 1st Data Science intern**
- **Tensorflow for beginners (concepts + Examples) (4 parts)**
- **Preprocessing Time Series (with codes)**
- **Data Analytics for beginners**

- **Statistics for beginners (4 parts)**

Timeseries Time Series Analysis Time Seri