Contents

Preprocessing

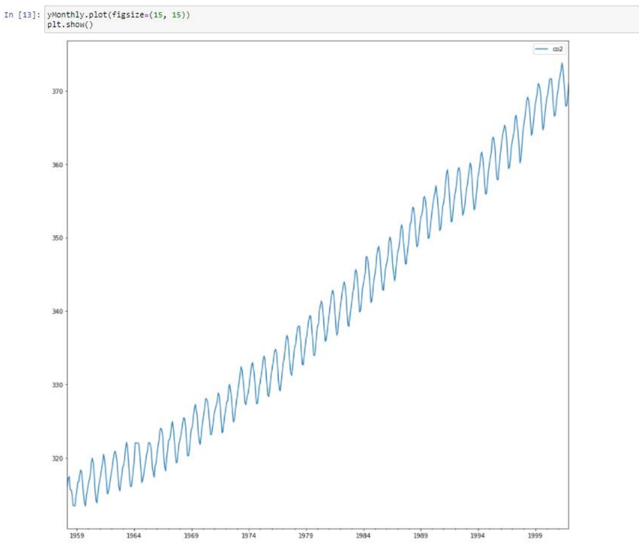
Timestamps

Missing values

Visualizing Data

The first step in Time Series Analysis is visualizing the data. A simple plot can give us the basic information.

```
dataframe.plot(figsize = (15, 15))
plt.show()
```



Decomposing a Timeseries

A simple plot like above may not always be easy to understand. A time series can be thought of as a combination of **Level, Trend, Seasonality and Noise**

Level – The average value of the series

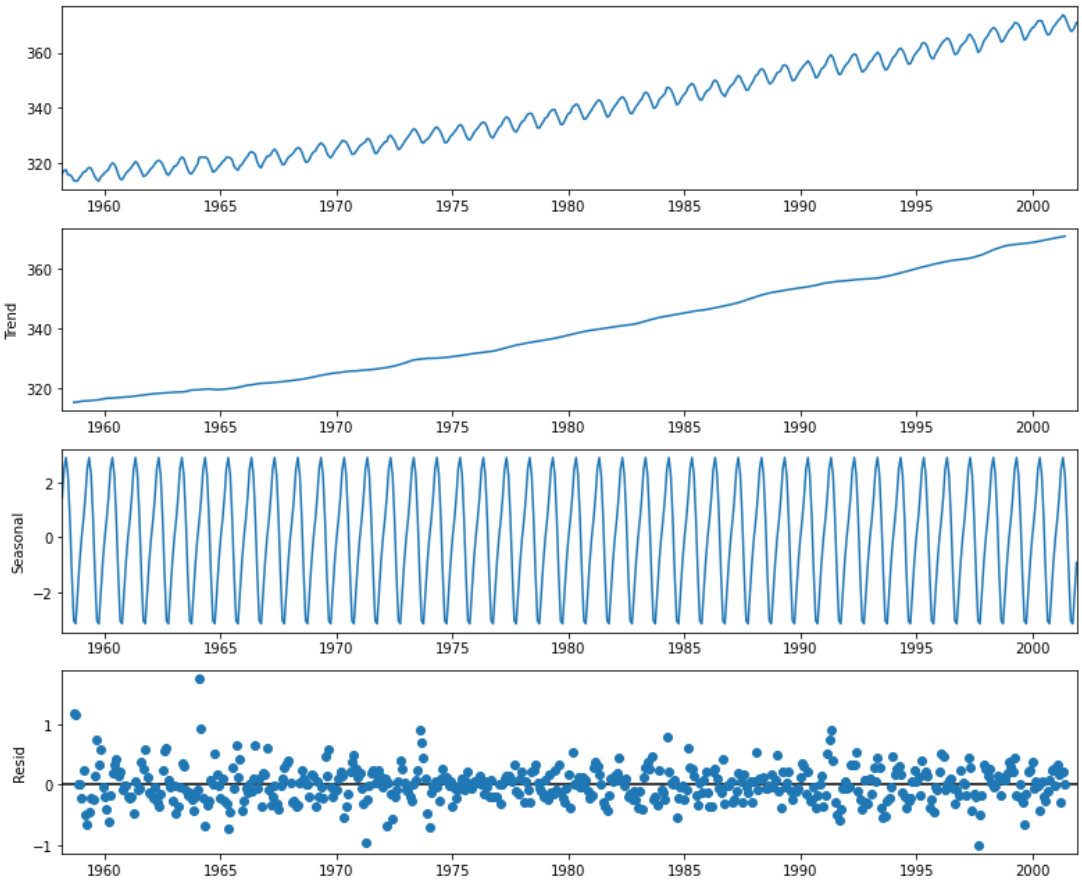
Trend - The increasing or decreasing value in the series

Seasonality – Repeating cycles in the series

Noise – Random variation in the series

We can decompose the given time series in as shown below

```
decomposition = sm.tsa.seasonal_decompose(yMonthly,
model='additive')
fig = decomposition.plot()
plt.show()
```



Stationarity Tests

A stationary time series is one whose statistical properties like mean and variance are constant over time. Most statistical forecasting methods assume that the time series data is stationary. Therefore, it is important to check if the data is non-stationary.

One of the most popular methods for checking stationarity is **Augmented Dickey-Fuller test.**

Null Hypothesis (H0): If failed to be rejected, it suggests the time series is non-stationary. It has some time dependent structure.

Alternate Hypothesis (H1): The null hypothesis is rejected; it suggests the time series is stationary. It does not have time-dependent structure.

We interpret this result using the **p-value** from the test. A p-value below a threshold (such as 5% or 1%) suggests we reject the null hypothesis (stationary), otherwise a p-value above the threshold suggests we fail to reject the null hypothesis (non-stationary).

Below code shows a function that accepts a time series and does a stationarity test

```
def adf_test(ts, signif=0.05):
    dftest = adfuller(ts, autolag='AIC')
    adf = pd.Series(dftest[0:4], index=['Test Statistic','p-value','# Lags','# Observations'])

    for key,value in dftest[4].items():
       adf['Critical Value (%s)'%key] = value
    print (adf)

    p = adf['p-value']
    if p <= signif:
        print(f" Series is Stationary")
    else:
        print(f" Series is Non-Stationary")
```

Output of the test

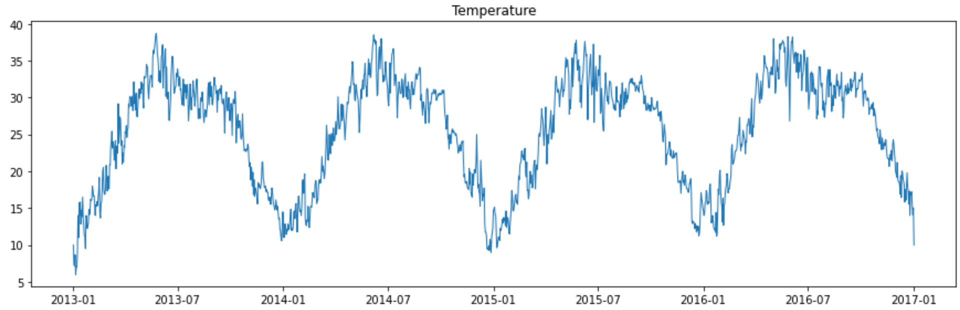
```
Test Statistic           2.359810
p-value                  0.998990
# Lags                  14.000000
# Observations         511.000000
Critical Value (1%)     -3.443212
Critical Value (5%)     -2.867213
Critical Value (10%)    -2.569791
dtype: float64
 Series is Non-Stationary
```

How to make a time series stationary?

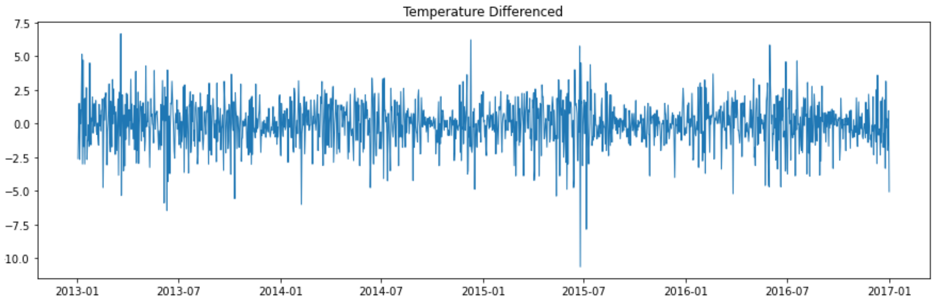
Differencing

The first difference of a time series is the series of changes from one period to the next.

The image below shows the original values of a time series.



The image below shows the first difference of the above time series.



Univariate Analysis

ARIMA Models

ARIMA stands for Auto Regressive Integrated Moving Average. It has 3 main parameters. **p, q, d**

p = Periods to lag (AR Terms)
if P = 3 then we will use the three previous periods of our time series in the autoregressive portion of the calculation

q = This is the number of MA terms. These are lagged forecast errors. If q is 5, the predictors for x(t) will be e(t-1)….e(t-5) where e(i) is the difference between the moving average at ith instant and actual value. This variable denotes the lag of the error component, where error component is a part of the time series not explained by trend or seasonality

d = d refers to the number of differencing transformations required by the time series to get stationarity.

Determining the Parameters

We use ACF and PACF plots to determine the values for p, q and d

Autocorrelation and Partial Autocorrelation Plots

ACF and PACF plots explained. How to read them and determine p q and d

Auto ARIMA and Seasonal ARIMA

Alternative to manually determining the parameters for ARIMA

Analyzing Residuals

Analyzing the results of ARIMA

Holt-Winters Model

Another popular model for univariate time series analysis

Multivariate Analysis

Granger Causality

VAR

VARMAX

Prophet (By Facebook)