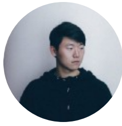


Jae Duk Seo Follow
https://jaedukseo.me | | | | |Your everyday Seo, who likes kimchi
Jun 1 · 7 min read

Trend, Seasonality, Moving Average, Auto Regressive Model : My Journey to Time Series Data with Interactive Code

GIF from this website

Recently I have been working with Time Series Data. I wanted to review what a Time series is as well as make my understanding more concert on Time Series Data.

> ***Please note this post is for my future self and for me to gain more deeper understanding of Time Series.***

. . .

Definition

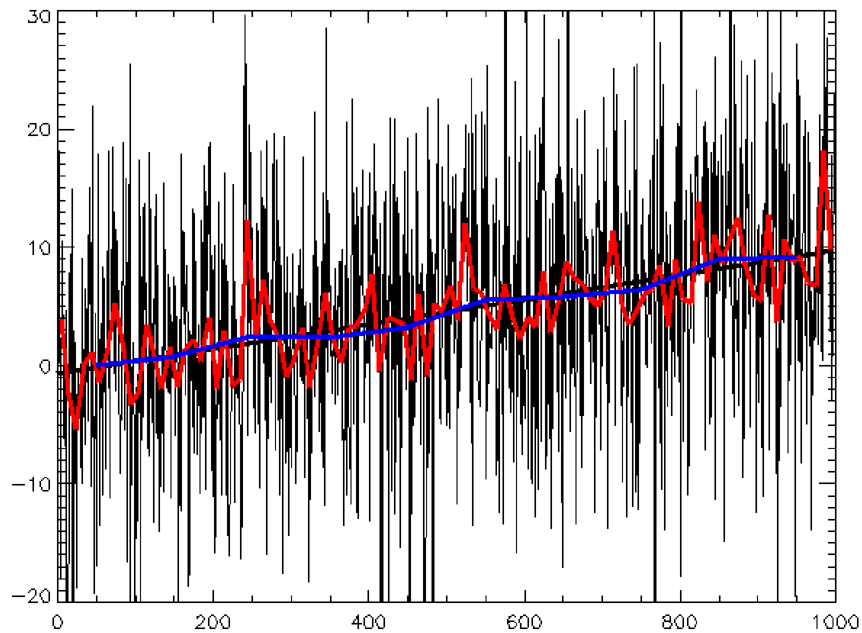
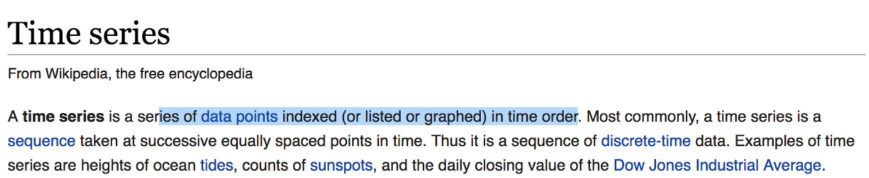


Image from this website

There are multiple of different sources defining the term 'Time Series' , so here I'll try to give a general definition that is easy for my self to understand.



Time series

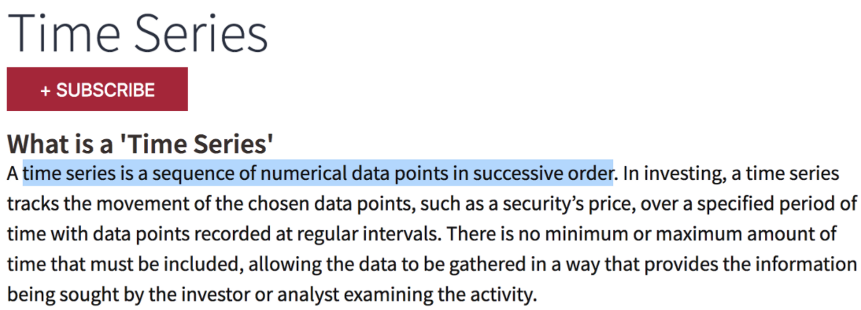
From Wikipedia, the free encyclopedia

A **time series** is a series of data points indexed (or listed or graphed) in time order. Most commonly, a time series is a sequence taken at successive equally spaced points in time. Thus it is a sequence of discrete-time data. Examples of time series are heights of ocean tides, counts of sunspots, and the daily closing value of the Dow Jones Industrial Average.

Image from this website

> *A **time series** is a series of data points indexed (or listed or graphed) in time order.*

As seen above, wiki gives very straight forward definition, any data in a sequential time order. Now lets take a look at the definition from investopedia.



Time Series

+ SUBSCRIBE

What is a 'Time Series'

A time series is a sequence of numerical data points in successive order. In investing, a time series tracks the movement of the chosen data points, such as a security's price, over a specified period of time with data points recorded at regular intervals. There is no minimum or maximum amount of time that must be included, allowing the data to be gathered in a way that provides the information being sought by the investor or analyst examining the activity.

Image from this website

> *A time series is a sequence of numerical data points in successive order.*

As seen above, we can get a general idea of what a time series data can be. It can be any data recored over time in sequential order. From the start we can think of stock prices, however videos, languages, songs, and MRI Scans can be thought of Time Series data as well.

. . .

Types of Time Series Data

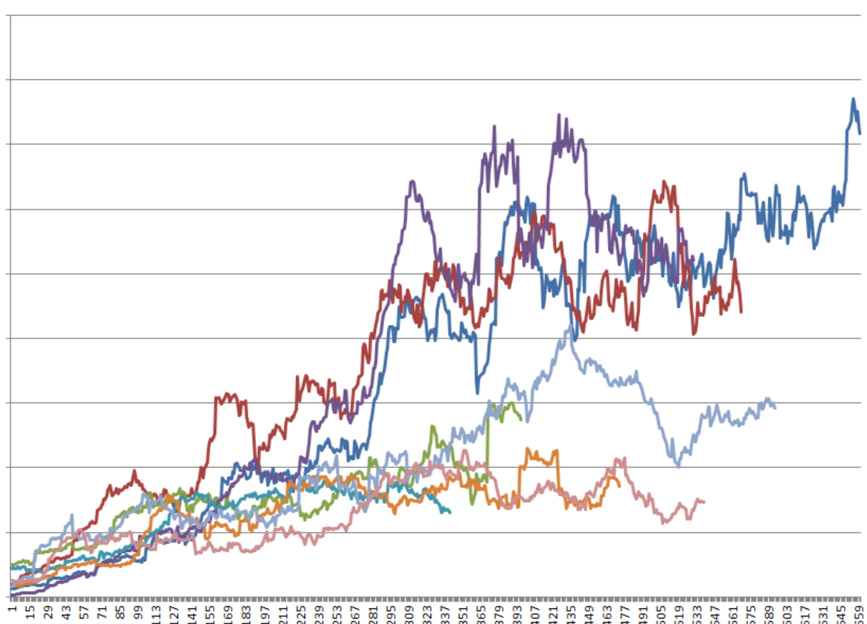


Image from this website

Above is an example image of Stock Prices, and we can observe that in the x-axis we have the time index, and in the y-axis we have the stock prices of different markets.

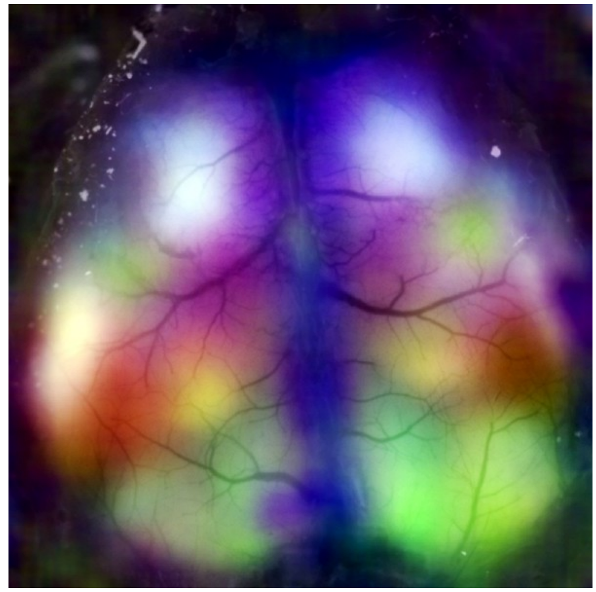


Image from this website

Another great example of Time Series data in computer vision world would be video (above is a GIF), since we capture frame by frame in one time sequence. And we can clearly see fMRI scan can be named as Time Series Data.

· · ·

Goals of Time Series

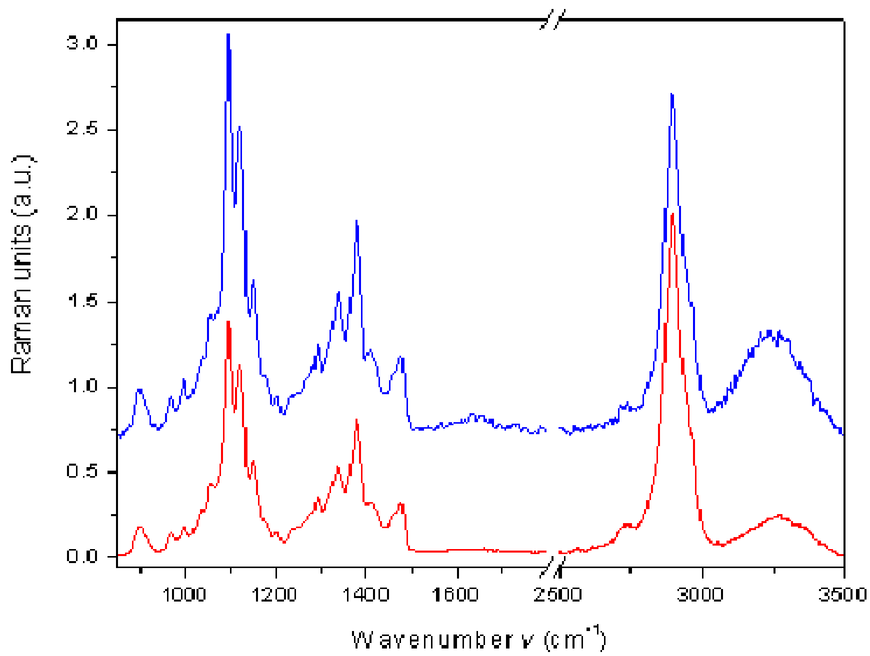


Image from this website

One obvious example use case of Time Series is predicting stock prices. (Well if this was so easy, a lot of Data Scientist would be rich.) But I wanted to add some different uses cases, we can even use GAN's to predict the next frame of a given video, or MRI Scan. Imagine where a GAN takes in few sequences of your MRI Scan and generates a MRI scan if you had cancer inside your body. (That would be pretty cool.). So in general we want to forecast / predict the next value when it comes to Time Series.

. . .

Trend / Seasonality / Noise

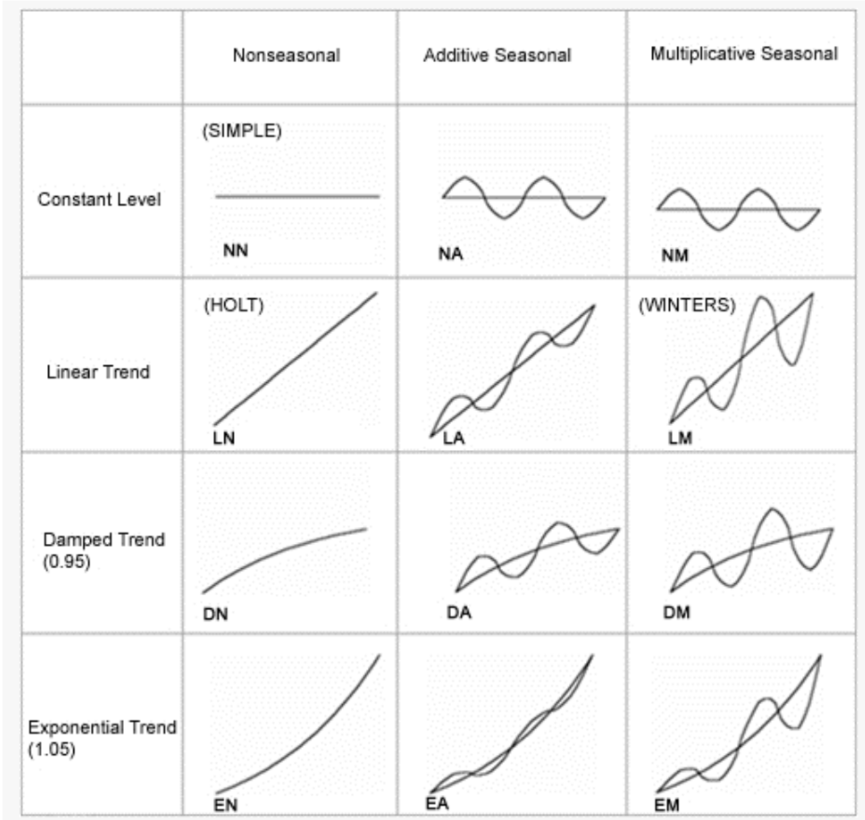
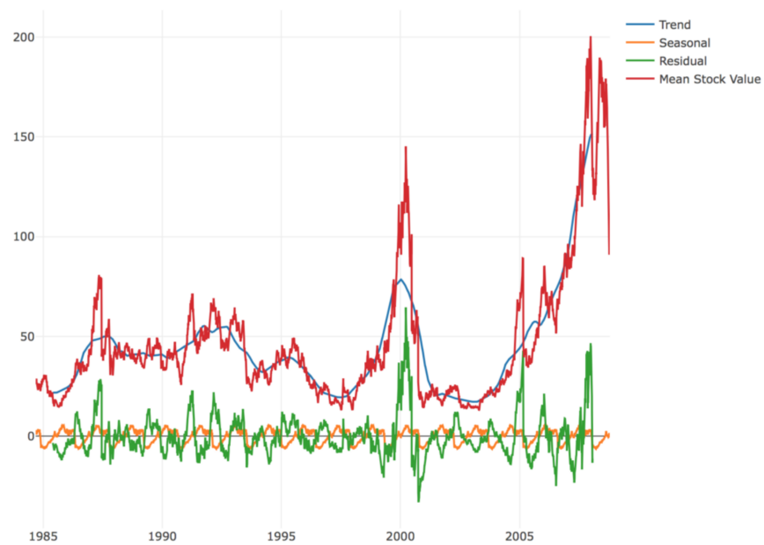


Image from this website

Before we move on we need to discuss something important, most time series data can be described by three components. And those are trend, seasonality and bias.

Trend → a general systematic linear or (most often) nonlinear component that changes over time and does not repeat
Seasonality → a general systematic linear or (most often) nonlinear component that changes over time and does repeat
Noise → a non-systematic component that is nor Trend/Seasonality within the data



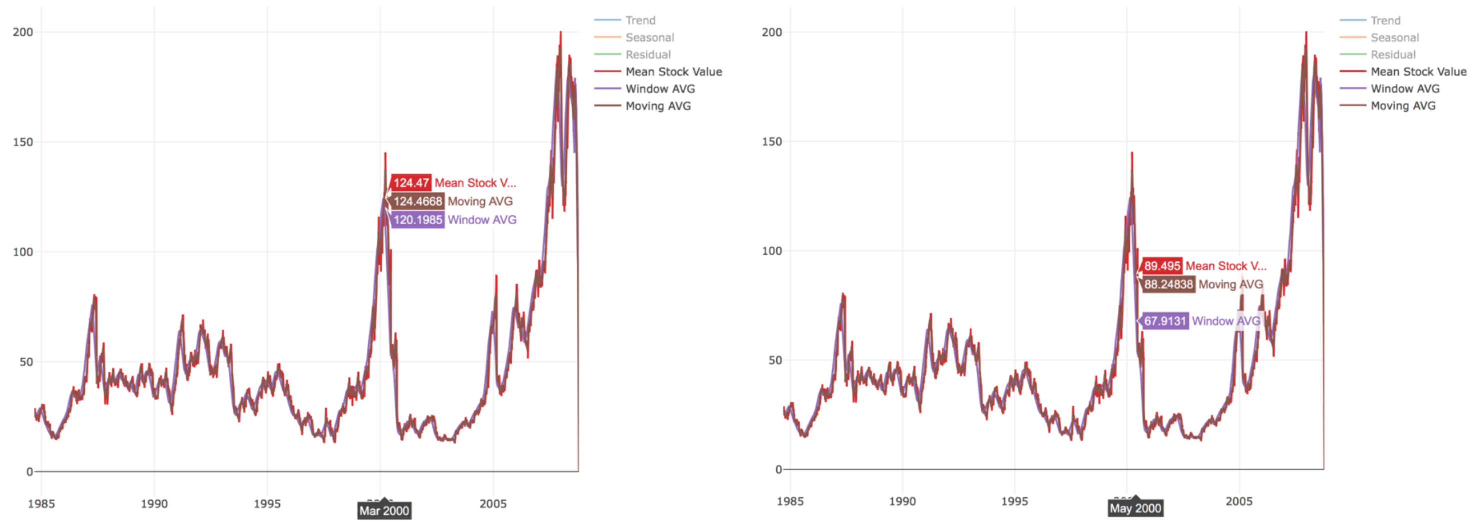

```python
from statsmodels.tsa.seasonal import seasonal_decompose
decomposition = seasonal_decompose(df.Mean, freq=365)
trace1 = go.Scatter(
    x = df.Date,y = decomposition.trend,
    name = 'Trend',mode='line'
)
trace2 = go.Scatter(
    x = df.Date,y = decomposition.seasonal,
    name = 'Seasonal',mode='line'
)
trace3 = go.Scatter(
    x = df.Date,y = decomposition.resid,
    name = 'Residual',mode='line'
)
trace4 = go.Scatter(
    x = df.Date,y = df.Mean,
    name = 'Mean Stock Value',mode='line'
)
data = [trace1,trace2,trace3,trace4]
plot(data)
```

Right Image → Python Code to Create the Visualization
Red Line → Apple Stock Prices from 1985
Blue Line → Trend for Apple Stock Price
Green Line → Residual (Noise) for Apple Stock Price
Orange Line → Seasonal (Yearly) trend for Apple Stock Price

With simple code and statsmodel library we can easily see how each components related to one another. We can observe that there is a seasonal increase every year, as well as general trend for apple stock price is increasing.

. . .

Approach to Predict / Forecast



Left/Right Image → Standard Window Avg / Exponentially Moving Avg

There are lot of different methods that we can use when we want perform forecasting. I will mention the methods that I have found during my research.

Standard / Exponentially Moving Average → calculation to analyze data points by creating series of averages of different subsets of the full data set

Auto Regression → is a representation of a type of random process; as such, it is used to describe certain time-varying processes in nature, economics, etc

Linear/Polynomial Regression → regression analysis in which the relationship between the independent variable x and the dependent variable $y$is modelled as an nth degree polynomial (or 1 degree for linear)

ARMA → model that provide a parsimonious description of a (weakly) stationary stochastic process in terms of two polynomials, one for the autoregression and the second for the moving average.

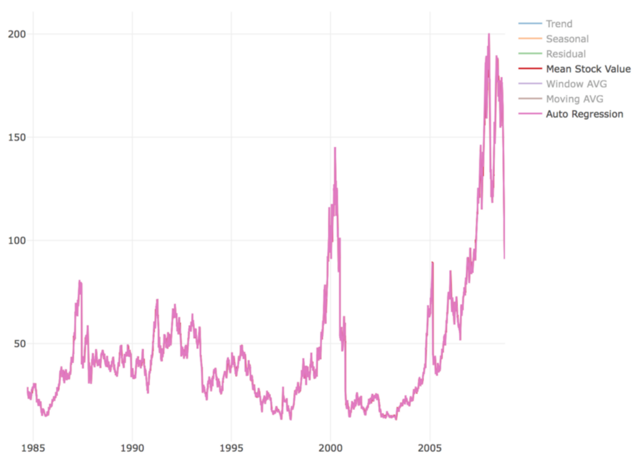
ARIMA (Autoregressive integrated moving average) → is a generalization of an autoregressive moving average (ARMA) model.

Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting)

Seasonal ARIMA → seasonal AR and MA terms predict *xt* using data values and errors at times with lags that are multiples of *S* (the span of the seasonality)

ARIMAX → An ARIMA model with covariate on the right hand side

Recurrent Neural Network (LSTM) → a class of artificial neural networkwhere connections between nodes form a directed graph along a sequence in which allows it to exhibit dynamic temporal behavior for a time sequence.




```python
from statsmodels.tsa.ar_model import AR
window_size = 50
ar_list = list(Mean_list[:window_size])
for pred_idx in range(window_size,N):

    current_window = Mean_list[pred_idx-window_size:pred_idx]
    model = AR(current_window)
    model_fit = model.fit(49)
    current_predict = model_fit.predict(49,49)[0]
    ar_list.append(current_predict)

trace7 = go.Scatter(
    x = df.Date,y = ar_list,
    name = 'Auto Regression',mode='line'
)
```

As seen above even a simple Auto Regressive can fit the stock price quite perfectly. Please click here to read about the difference between ARMA, ARIMA and ARIMAX.

· · ·

Interactive Code

```
!wget https://raw.githubusercontent.com/JaeDukSeo/sample_data/master/aapl.csv

import pandas as pd

import pandas as pd
import numpy as np,sys
import tensorflow as tf
import matplotlib.pyplot as plt

from plotly.offline import download_plotlyjs, init_notebook_mode, plot, iplot
import plotly.graph_objs as go

import matplotlib.pyplot as plt
from numpy import newaxis

# 0. Get the Data and simple sorting and check NaN
df = pd.read_csv('aapl.csv',delimiter=',',usecols=['Date','Open','High','Low','Close'])
df.Date = pd.to_datetime(df.Date)
df['Mean'] = (df.High + df.Low )/2.0


from statsmodels.tsa.seasonal import seasonal_decompose
decomposition = seasonal_decompose(df.Mean.values, freq=365)
trace1 = go.Scatter(
    x = df.Date,y = decomposition.trend,
    name = 'Trend',mode='line'
)
trace2 = go.Scatter(
    x = df.Date,y = decomposition.seasonal,
    name = 'Seasonal',mode='line'
)
trace3 = go.Scatter(
    x = df.Date,y = decomposition.resid,
    name = 'Residual',mode='line'
)
trace4 = go.Scatter(
    x = df.Date,y = df.Mean,
    name = 'Mean Stock Value',mode='line'
)

# a. Standard Average of Window
```

For Google Colab, you would need a google account to view the codes, also you can't run read only scripts in Google Colab so make a copy on your play ground. Finally, I will never ask for permission to access your files on Google Drive, just FYI. Happy Coding!

To access the code used for this post, please click here.

· · ·

Final Words

I hope to continue my research to gain a deeper understanding about Time Series. Since I think its very common problem in every industry.

If any errors are found, please email me at jae.duk.seo@gmail.com, if you wish to see the list of all of my writing please view my website here.

Meanwhile follow me on my twitter here, and visit my website, or my Youtube channel for more content. I also implemented Wide Residual Networks, please click here to view the blog post.

· · ·

Reference

1. Time series. (2018). En.wikipedia.org. Retrieved 2 June 2018, from https://en.wikipedia.org/wiki/Time_series
2. Staff, I. (2006). Time Series. Investopedia. Retrieved 2 June 2018, from https://www.investopedia.com/terms/t/timeseries.asp
3. How To Identify Patterns in Time Series Data: Time Series Analysis . (2018). Statsoft.com. Retrieved 2 June 2018, from http://www.statsoft.com/Textbook/Time-Series-Analysis
4. statsmodels.tsa.seasonal.seasonal_decompose—statsmodels 0.9.0 documentation. (2018). Statsmodels.org. Retrieved 2 June 2018, from http://www.statsmodels.org/dev/generated/statsmodels.tsa.seasonal.seasonal_decompose.html
5. StatsModels: Statistics in Python—statsmodels 0.9.0 documentation. (2018). Statsmodels.org. Retrieved 2 June 2018, from https://www.statsmodels.org/stable/index.html
6. Deshpande, B. (2018). Time series forecasting: understanding trend and seasonality. Simafore.com. Retrieved 2 June 2018, from http://www.simafore.com/blog/bid/205420/Time-series-forecasting-understanding-trend-and-seasonality
7. Moving average. (2018). En.wikipedia.org. Retrieved 2 June 2018, from https://en.wikipedia.org/wiki/Moving_average
8. Linear regression. (2018). En.wikipedia.org. Retrieved 2 June 2018, from https://en.wikipedia.org/wiki/Linear_regression
9. Polynomial regression. (2018). En.wikipedia.org. Retrieved 2 June 2018, from https://en.wikipedia.org/wiki/Polynomial_regression
10. Autoregressive–moving-average model. (2018). En.wikipedia.org. Retrieved 2 June 2018, from https://en.wikipedia.org/wiki/Autoregressive%E2%80%93moving-average_model
11. Recurrent neural network. (2018). En.wikipedia.org. Retrieved 2 June 2018, from https://en.wikipedia.org/wiki/Recurrent_neural_network

12. Autoregressive model. (2018). En.wikipedia.org. Retrieved 2 June 2018, from https://en.wikipedia.org/wiki/Autoregressive_model

13. The ARIMAX model muddle | Rob J Hyndman. (2010). Robjhyndman.com. Retrieved 2 June 2018, from https://robjhyndman.com/hyndsight/arimax/

14. 4.1 Seasonal ARIMA models | STAT 510. (2018). Newonlinecourses.science.psu.edu. Retrieved 2 June 2018, from https://newonlinecourses.science.psu.edu/stat510/node/67/

15. statsmodels.tsa.ar_model.AR.predict—statsmodels 0.9.0 documentation. (2018). Statsmodels.org. Retrieved 2 June 2018, from http://www.statsmodels.org/dev/generated/statsmodels.tsa.ar_model.AR.predict.html#statsmodels.tsa.ar_model.AR.predict

16. Python, A. (2018). Autoregressive model using statsmodels in Python. Stack Overflow. Retrieved 2 June 2018, from https://stackoverflow.com/questions/28094538/autoregressive-model-using-statsmodels-in-python

17. Anon, (2018). [online] Available at: https://www.researchgate.net/figure/Raman-spectra-of-the-MCC-101-raw-material-with-123-water-content-top-and-of-the-MCC_fig11_8508180 [Accessed 2 Jun. 2018].