Critical Thinking Group 4 : DATA621 Final Project

Table of Contents

Contents

TEAM MEMBERS

Rajwant Mishra
Priya Shaji
Debabrata Kabiraj
Isabel Ramesar
Sin Ying Wong
Fan Xu

ABSTRACT

- In this project, 5 years of live historical dataset (2015 to 2020) of stocks has been explored and analyzed with time series models.
- Time series models used are AR(Auto Regression) and MA(Moving Average) .
- Time series forecasting process has been executed using ACF() function and ACF plots.
- Lastly, we have evaluated all data models comparing its prediction score to analyze which model has performed better.

KEYWORDS

- Lag : A "lag" is a fixed amount of passing time; One set of observations in a time series is plotted (lagged) against a second, later set of data. The kth lag is the time period that happened "k" time points before time i. The most commonly used lag is 1, called a first-order lag plot.
- Seasonality : In time series data, seasonality is the presence of variations that occur at specific regular intervals less than a year, such as weekly, monthly, or quarterly.
- Stationary : Stationary graphs are relevant to time series analysis, where we seek to understand the changes of a graph over time. With time series analysis, it is expected for data to vary over time, however, it is difficult to figure out the exact pattern by which a graph will change over time.
- Random Walk : A random walk, on the other hand, does not have this same tendency to centralize towards the mean due to the individual points along the walk being dependent on the previous points. This adds variance the more points are included in the walk, which can cause the path of the walk to deviate very far away from the mean.
- White Noise : With a white noise graph, we know that the distribution of the points will be normal and centered around zero with the same variance because the points are independent, so the tendency over time will be towards the mean
- AR (Auto regressive) : In this regression model, the response variable in the previous time period has become the predictor and the errors have our usual assumptions about errors in a simple linear regression model. The order of an autoregression is the number of immediately preceding values in the series that are used to predict the value at the present time. So, the preceding model is a first-order autoregression, written as AR(1).

- MA (Moving Average) :Moving averages are a simple and common type of smoothing used in time series analysis and time series forecasting. Calculating a moving average involves creating a new series where the values are comprised of the average of raw observations in the original time series.
- In time series analysis, the moving-average model (MA model), also known as moving-average process, is a common approach for modeling univariate time series. The moving-average model specifies that the output variable depends linearly on the current and various past values of a stochastic (imperfectly predictable) term.

INRTODUCTION

- We collected data from NYSE from Lat Five year using the API in R
- Also scrapped data of Sectors and Stock so that we can understand the trend by Sector from our data.
- Analyzed 5 year data by Sector and choose one of the stocks from Healthcare sector.
- We partitioned our data in data before 2020 and after 2020.
- Build AR and MA model on data before 2020 and predicted stock value for Year 2020
- We were able to check accuracy of the model by Model comparison and graph .

LITERATURE REVIEW

- One of the interesting works in stocks analysis is "using data mining with time series data in short-term stocks prediction" which explores methodologies similar to our project.
- Their approach uses data mining with time series data using examples related with short-term stocks prediction which is proved to be important to a better understanding of the field
- Specific challenges: developers focus on the issue of representing time series data in order to effectively and efficiently apply data mining.
- Another interesting issue was to find out if different time series or parts of time series have similar behavior.
- This issue can be approached through the use of similarity measures or indexing techniques.
- Over-fitting is a common problem across data mining applications.
- Achievements: A new concept, named as "median strings" is presented as a simple and at the same time powerful representation of time series data.
- Our work/investigation on our project is different from their's by our approach used to solve a similar issue
- Time series models used by us are AR and MA model, as both these models perform with better predictions specifically when market is not stable, which hold true for current covid-19 scenario.
- Link to book we used to learn how other researchers have solved similar issue: Data mining with time series data

METHODOLOGY

- We have used Auto Regressive Integrated Moving Average Model with AR and MA model.
- Together with the autoregressive (AR) model, the moving-average model is a special case and key component of the more general ARMA and ARIMA models of time series, which have a more complicated stochastic structure.
- A time series is a sequence of measurements of the same variable(s) made over time. Usually the measurements are made at evenly spaced times - for example, monthly or yearly. Let us first consider the problem in which we have a y-variable measured as a time series. As an example, we might have y a measure of global temperature, with measurements observed each year. To emphasize that we have measured values over time, we use "t" as a subscript rather than the usual "i," i.e., y_t means y measured in time period t.
- An autoregressive model is when a value from a time series is regressed on previous values from that same time series. for example, y_t on y_{t-1}: $y_t=\beta_0+\beta_1 y_{t-1}+\epsilon_t$.
- In this regression model, the response variable in the previous time period has become the predictor and the errors have our usual assumptions about errors in a simple linear regression model. The order of an autoregression is the number of

immediately preceding values in the series that are used to predict the value at the present time. So, the preceding model is a first-order autoregression, written as AR(1).

OVERVIEW

In this analysis we will analyze Stocks data and check how Covid-19 had impacted it from the beginning of the year 2020. We have collected data from exchange for last Five year starting from year 2015 to Apr 27, 2020. We would be using Moving Average Model and Auto Regressive model to analyze the time series data.

Column Name	Type	Sample Data	Descrption
X1	num	0	Sequence
begins_at	POSIXct,	4/28/2015	Date
open_price	num	24.9	Open Price
close_price	num	24.9	Close Price
high_price	num	24.9	High Price of stock on the given date.
low_price	num	24.9	Loweest price of the stock on the given date
volume	num	0	Volume of the shares trade on the give date
session	chr	reg	Type of the data session Regular /Extended
interpolated:	TRUE	TRUE	Not used
sname	chr	AA	Stock Name

Below is a short snippet of the data in the data set:

X1 <dbl>	begins_at <S3: POSIXct>	open_price <dbl>	close_price <dbl>	high_price <dbl>	low_price <dbl>	volume <dbl>	session <chr>	interpolated <lgl>	sname <chr>
0	2015-04-28	24.8871	24.8871	24.8871	24.8871	0	reg	TRUE	AA
1	2015-04-29	24.8871	24.8871	24.8871	24.8871	0	reg	TRUE	AA
2	2015-04-30	24.8871	24.8871	24.8871	24.8871	0	reg	TRUE	AA
3	2015-05-01	24.8871	24.8871	24.8871	24.8871	0	reg	TRUE	AA
4	2015-05-04	24.8871	24.8871	24.8871	24.8871	0	reg	TRUE	AA
5	2015-05-05	24.8871	24.8871	24.8871	24.8871	0	reg	TRUE	AA

Sector information for the stocks:

```
 [1] "Basic Materials Sector"        "Communication Services Sector" "Consumer Cyclical Sector"
 [4] "Consumer Defensive Sector"     "Energy Sector"                 "Financial Services Sector"
 [7] "Healthcare Sector"             "Industrials Sector"            "Technology Sector"
[10] "Utilities Sector"
```

DELIVERABLES

- A write-up submitted in PDF format. Your write-up should have four sections. Each one is described below. You may assume you are addressing me as a fellow data scientist, so do not need to shy away from technical details.
- Assigned predictions (the number of wins for the team) for the evaluation data set.
- Include your R statistical programming code in an Appendix.

DATA EXPLORATION

We will be exploring the 500 stocks data by sectors and then we will choose one stocks to do further analysis.

We have collected all the stocks from NYSE and corresponding sector info:

- Basic Materials Sector
- Communication Services Sector
- Consumer Cyclical Sector
- Consumer Defensive Sector
- Energy Sector
- Financial Services Sector
- Healthcare Sector
- Industrials Sector
- Technology Sector
- Utilities Sector

Let’s looks at our data structure of the data :

```
Observations: 370,440
Variables: 10
$ X1          <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17,...
$ begins_at   <dttm> 2015-04-28, 2015-04-29, 2015-04-30, 2015-05-01, 2015-05-04, ...
$ open_price  <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871...
$ close_price <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871...
$ high_price  <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871...
$ low_price   <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871...
$ volume      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
$ session     <chr> "reg", "reg", "reg", "reg", "reg", "reg", "reg", "reg", "reg"...
$ interpolated <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, T...
$ sname       <chr> "AA", "AA", "AA", "AA", "AA", "AA", "AA", "AA", "AA", "AA", "...
```

Summary of the data:

```
##       X1          begins_at                  open_price
##  Min.   :   0.0   Min.   :2015-04-28 00:00:00   Min.   :   0.00
##  1st Qu.: 314.8   1st Qu.:2016-07-26 18:00:00   1st Qu.:  13.61
##  Median : 629.5   Median :2017-10-24 12:00:00   Median :  26.33
##  Mean   : 629.5   Mean   :2017-10-25 14:51:25   Mean   :  48.92
##  3rd Qu.: 944.2   3rd Qu.:2019-01-25 18:00:00   3rd Qu.:  56.09
```

```
## Max.   :1259.0   Max.   :2020-04-27 00:00:00   Max.   :1266.56

##  close_price        high_price       low_price        volume
## Min.   :   0.0052   Min.   :   0.00   Min.   :   0.0   Min.   :        0
## 1st Qu.:  13.6100   1st Qu.:  13.72   1st Qu.:  13.5   1st Qu.:   103282
## Median :  26.3300   Median :  26.67   Median :  26.0   Median :   421838
## Mean   :  48.9221   Mean   :  49.43   Mean   :  48.4   Mean   :  1650980
## 3rd Qu.:  56.1200   3rd Qu.:  56.68   3rd Qu.:  55.5   3rd Qu.:  1332573
## Max.   :1250.0000   Max.   :1274.41   Max.   :1232.0   Max.   :375088650

##   session          interpolated       sname
## Length:370440      Mode :logical   Length:370440
## Class :character   FALSE:362193    Class :character
## Mode  :character   TRUE :8247      Mode  :character
```

X1	begins_at	open_price	close_price	high_price	low_price	volume	session	interpolated	sname
Min. : 0.0	Min. :2015-04-28 00:00:00	Min. : 0.00	Min. : 0.0052	Min. : 0.00	Min. : 0.0	Min. : 0	Length:370440	Mode :logical	Length:370440
1st Qu.: 314.8	1st Qu.:2016-07-26 18:00:00	1st Qu.: 13.61	1st Qu.: 13.6100	1st Qu.: 13.72	1st Qu.: 13.5	1st Qu.: 103282	Class :character	FALSE:362193	Class :character
Median : 629.5	Median :2017-10-24 12:00:00	Median : 26.33	Median : 26.3300	Median : 26.67	Median : 26.0	Median : 421838	Mode :character	TRUE :8247	Mode :character
Mean : 629.5	Mean :2017-10-25 14:51:25	Mean : 48.92	Mean : 48.9221	Mean : 49.43	Mean : 48.4	Mean : 1650980	NA	NA	NA
3rd Qu.: 944.2	3rd Qu.:2019-01-25 18:00:00	3rd Qu.: 56.09	3rd Qu.: 56.1200	3rd Qu.: 56.68	3rd Qu.: 55.5	3rd Qu.: 1332573	NA	NA	NA
Max. :1259.0	Max. :2020-04-27 00:00:00	Max. :1266.56	Max. :1250.0000	Max. :1274.41	Max. :1232.0	Max. :375088650	NA	NA	NA

Calculate number of ZERO's in each variable in the main dataset:

variable	n	percent
volume	8625	2.3%
X1	294	0.1%
high_price	1	0%
low_price	1	0%
open_price	1	0%

This is clear from the above stats that in 2.3% of the data points we don’t have any value, 0.1% of the stock has no name which counts 294 in number, there is no zeros for the high price , low price, open price . We will not use volume in our analysis at this time so we will drop those data points latter.

Let’s see how many stocks from 500 stocks we are studying fall under which sectors:

Stocks by Sector:

Sector <chr>	CODE <chr>	NAME <chr>
Basic Materials Sector	GPRE	Green Plains Inc
Basic Materials Sector	BCPC	Balchem Corporation
Basic Materials Sector	STLD	Steel Dynamics, Inc
Basic Materials Sector	METC	Ramaco Resources, Inc
Basic Materials Sector	ASH	Ashland Global Holdings Inc
Basic Materials Sector	TRX	Tanzanian Gold Corporation

Yearly Average Price For The Stocks:

sname <chr>	Avg_price <dbl>
AA	30.50011
AAN	40.21981
AAP	146.20186
AAT	40.64534
AB	26.15712
ABB	21.47852

Monthly Average Price For The Stocks

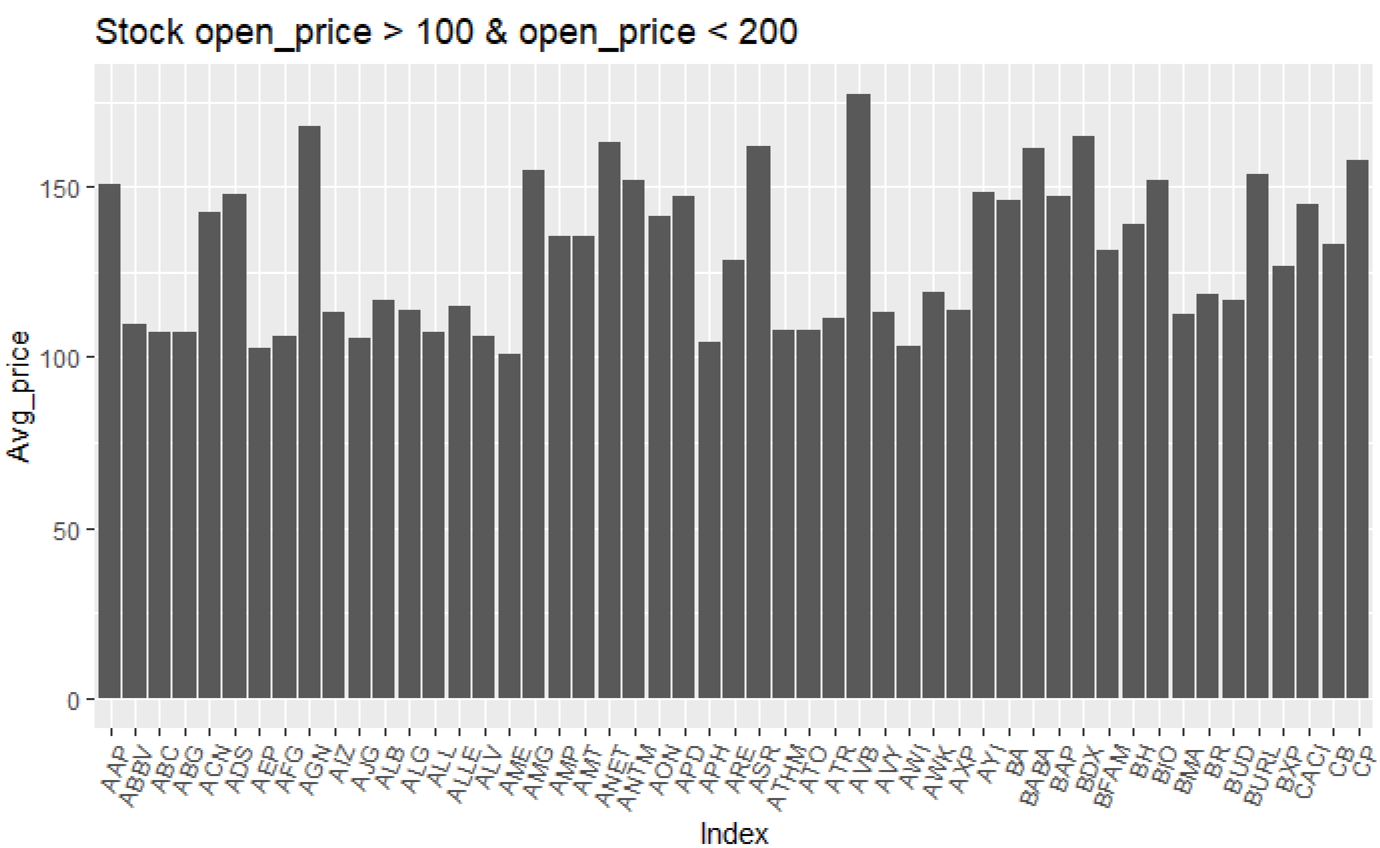
sname <chr>	Avg_price <dbl>
AA	30.50011
AAN	40.21981
AAP	146.20186
AAT	40.64534
AB	26.15712
ABB	21.47852

Average Price For A Stock From 2015 To 2020:

```
data_price_year[which(data_price_year$sname=='AA'),]
```

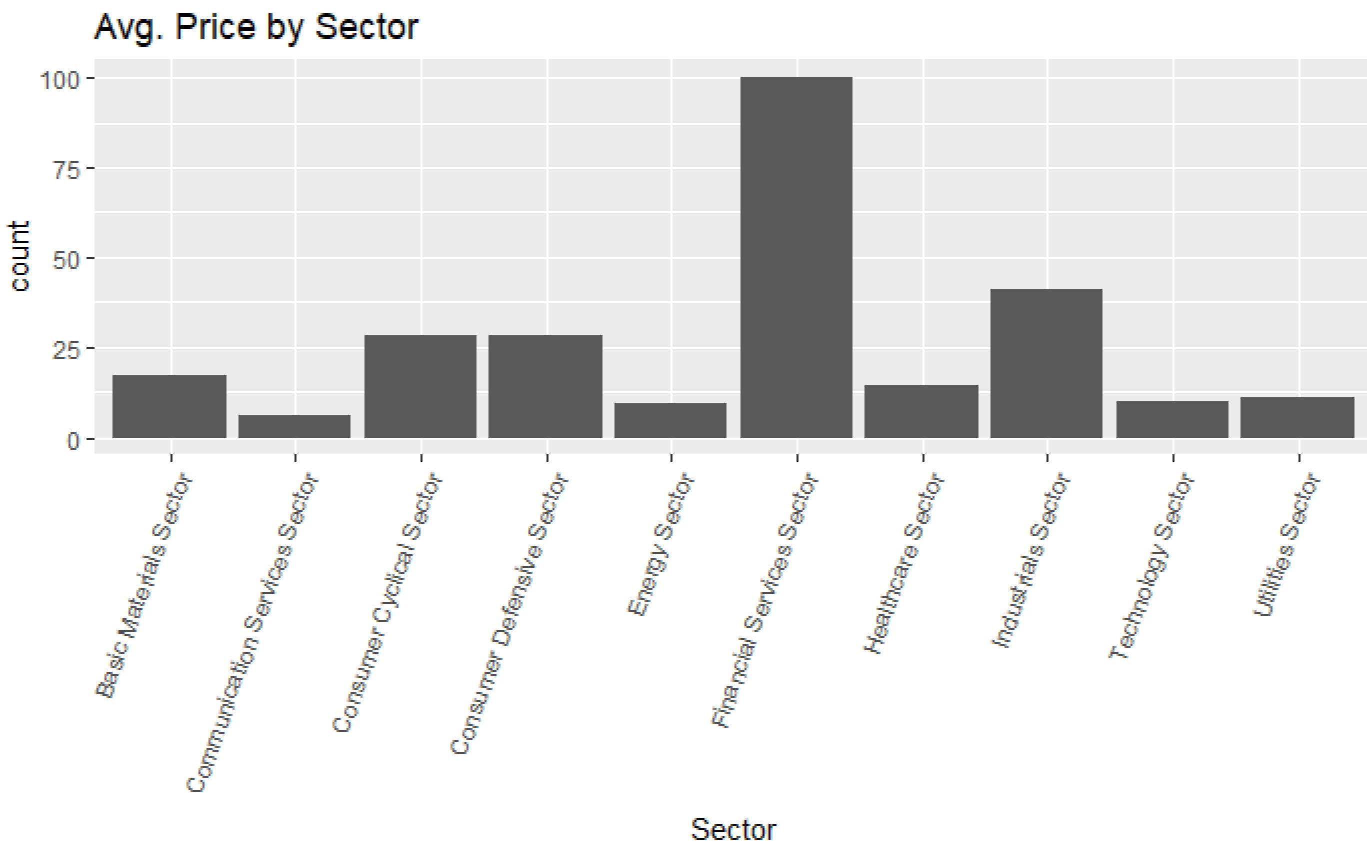
Year	sname	Avg_price
<dbl>	<chr>	<dbl>
2015	AA	24.88710
2016	AA	25.42091
2017	AA	38.25301
2018	AA	44.25964
2019	AA	23.70490
2020	AA	12.37538

Now, let’s only target shares whose open prices are between 100 and 200:



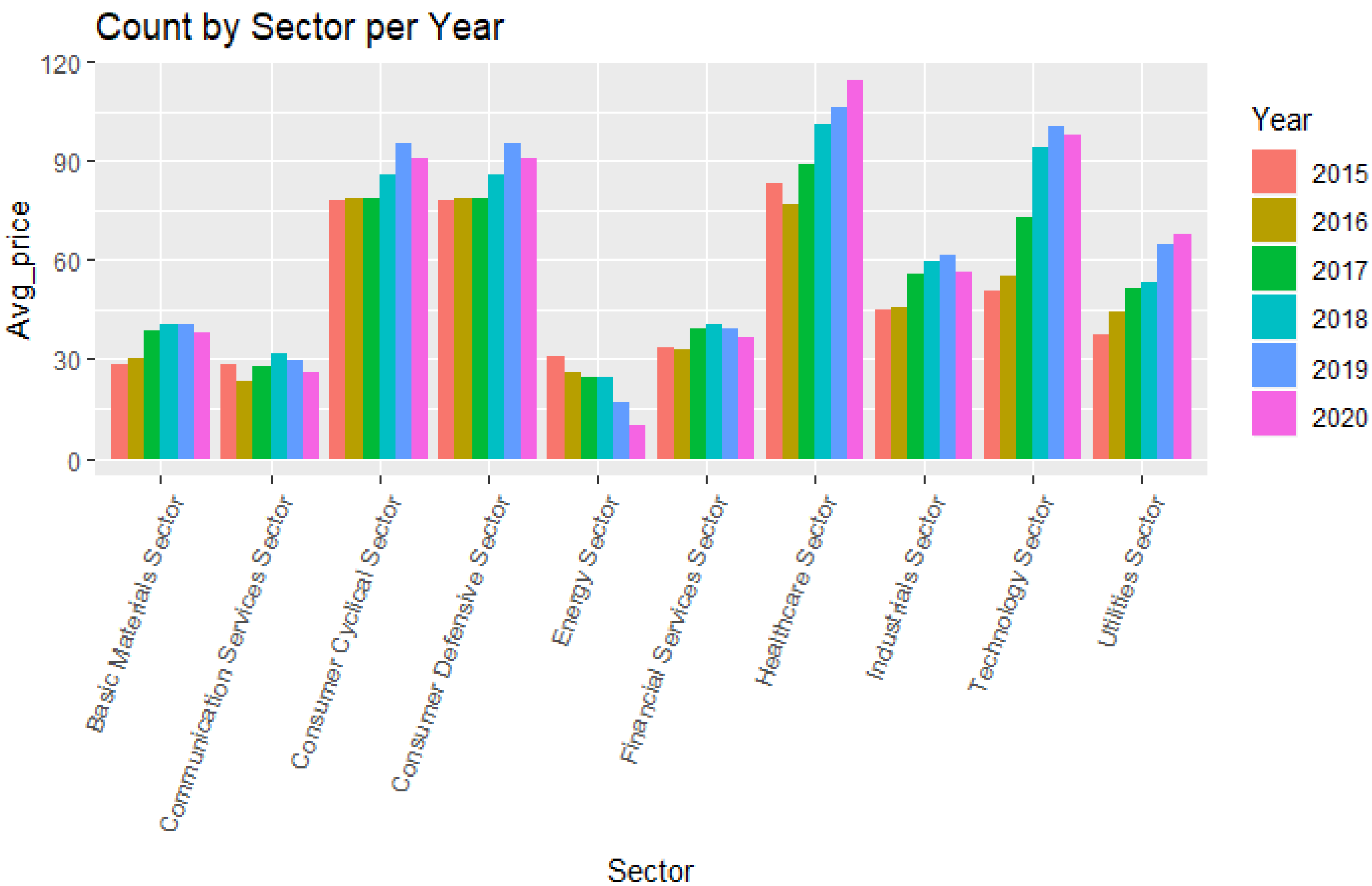
Avg Price By Sector:

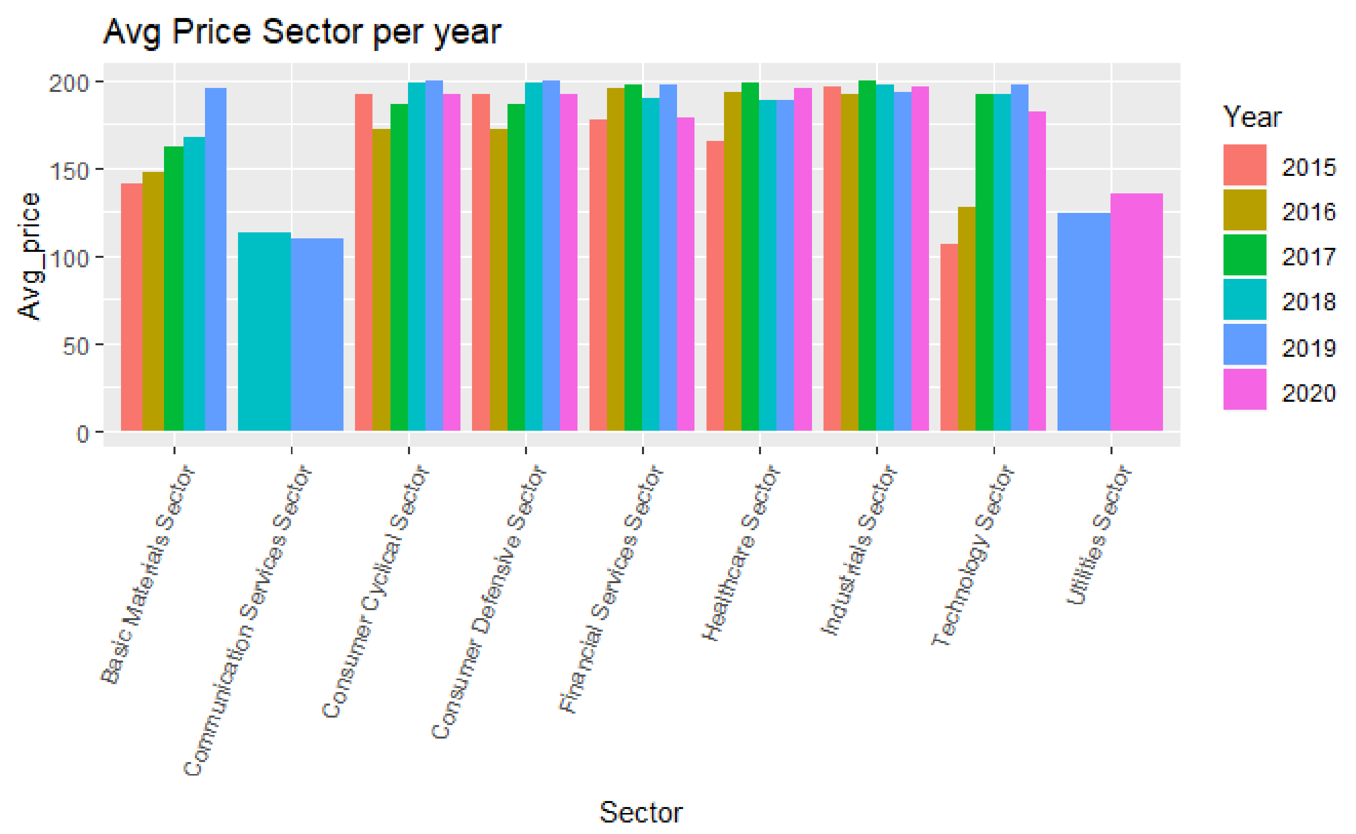
Checking only with 500 stocks data and analyze the distribution of data in each sector.



The graph above tells us about average price of stocks for each sector name.

Sector Per Year:

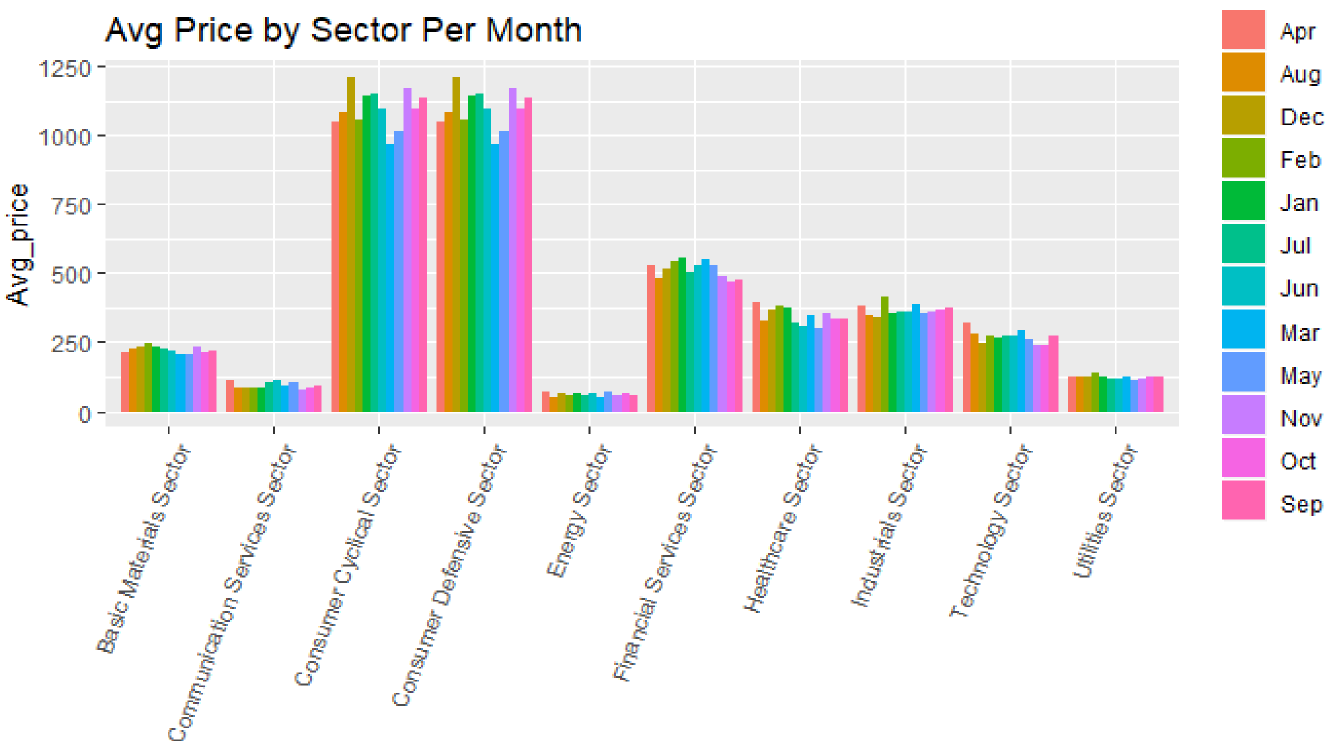




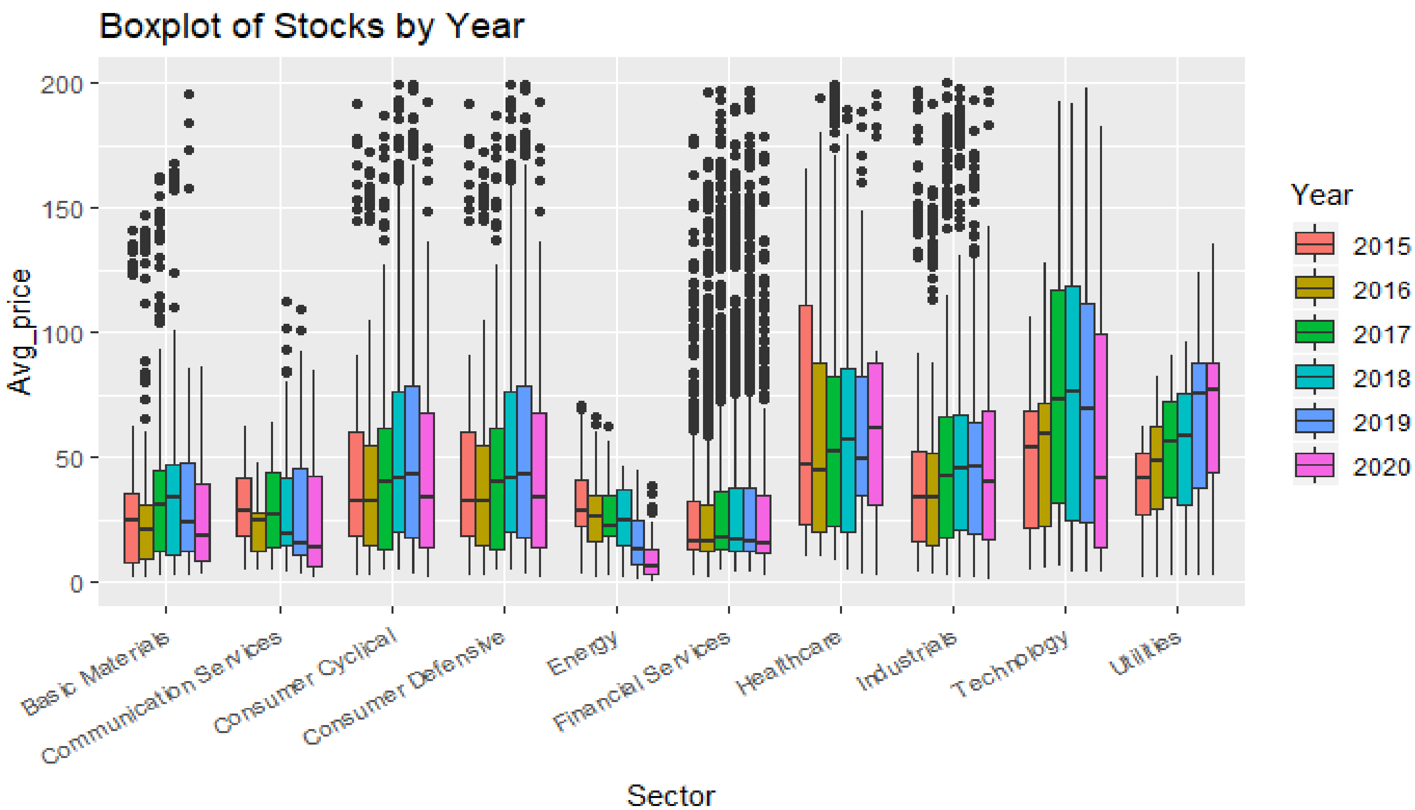
The graph above tells us about yearly increase avg. Price in each sectors.

Below is Graph of the sector by Month and year, which shows some pattern. We will do some analysis to see how stocks from few of these industries fit with `AR(Auto Regression)` and `MA(Moving Average)` model. Analyzing average price of stocks yearly for each sector in the dataset.

Analyzing average price of stocks monthly for each sector in the dataset:



Box plot for year:



From the box plot above we can analyze that mostly all the sectors in our dataset have some outliers throughout 5 years, except two sectors that are: `Technology` and `Utilities`.

DATA PREPARATION

Analyzing Top 3 Stocks In Each Sector:

sname <chr>	Avg <dbl>	Sector <chr>	NAME <chr>
AZO	800.9067	Consumer Cyclical Sector	AutoZone, Inc
AZO	800.9067	Consumer Defensive Sector	AutoZone, Inc
BLK	419.6630	Financial Services Sector	BlackRock, Inc
BH	291.6055	Consumer Cyclical Sector	Biglari Holdings Inc
BH	291.6055	Consumer Defensive Sector	Biglari Holdings Inc
BA	245.1263	Industrials Sector	The Boeing Company
BIO	233.8155	Healthcare Sector	Bio-Rad Laboratories, Inc
AGN	211.6256	Healthcare Sector	Allergan plc
ADS	208.0295	Financial Services Sector	Alliance Data Systems Corporation
ANTM	206.0807	Healthcare Sector	Anthem, Inc

sname <chr>	Avg <dbl>	Sector <chr>	NAME <chr>
BDX	203.2816	Healthcare Sector	Becton, Dickinson and Company
BAP	181.8395	Financial Services Sector	Credicorp Ltd
CP	178.9086	Industrials Sector	Canadian Pacific Railway Limited
AYI	171.8084	Industrials Sector	Acuity Brands, Inc
ANET	169.3343	Technology Sector	Arista Networks, Inc
ASR	163.6174	Industrials Sector	Grupo Aeroportuario del Sureste, S. A. B. de C. V

Building main stock data table to be used for analysis:

sname <chr>	Avg <dbl>	Sector <chr>	NAME <chr>
AZO	800.9067	Consumer Cyclical Sector	AutoZone, Inc
AZO	800.9067	Consumer Defensive Sector	AutoZone, Inc
BLK	419.6630	Financial Services Sector	BlackRock, Inc
BH	291.6055	Consumer Cyclical Sector	Biglari Holdings Inc
BH	291.6055	Consumer Defensive Sector	Biglari Holdings Inc
BA	245.1263	Industrials Sector	The Boeing Company

Building main stock data table to be used for analysis:

sname <chr>	Avg <dbl>	Sector <chr>	NAME <chr>
AZO	800.9067	Consumer Cyclical Sector	AutoZone, Inc
AZO	800.9067	Consumer Defensive Sector	AutoZone, Inc
BLK	419.6630	Financial Services Sector	BlackRock, Inc
BH	291.6055	Consumer Cyclical Sector	Biglari Holdings Inc
BH	291.6055	Consumer Defensive Sector	Biglari Holdings Inc
BA	245.1263	Industrials Sector	The Boeing Company

We will study the flow on some of the stocks from Health and Tech Sectors like:

ANTM Anthem, Inc
ANET Arista Networks, Inc
BA The Boeing Company
Glimpse of data:

Added Month name and converted data to yearly data so that further analysis can be done:

Month <chr>	sname <chr>	2015 <dbl>	2016 <dbl>	2017 <dbl>	2018 <dbl>	2019 <dbl>	2020 <dbl>
Apr	AA	24.887100	24.887100	33.420526	52.222381	28.256190	7.1155556
Apr	AAN	34.206667	26.280000	30.740526	46.094286	53.827619	23.2527778
Apr	AAP	144.643333	158.708571	143.973684	110.712381	174.845238	106.8144444
Apr	AAP	144.643333	158.708571	143.973684	110.712381	174.845238	106.8144444
Apr	AB	31.216667	23.933810	22.884211	26.573810	29.448095	19.6716667
Apr	ABB	21.850000	20.008810	23.278421	23.266190	20.081429	17.4333333
Apr	ABBV	65.230000	59.555714	64.750526	93.173333	81.045714	79.6416667
Apr	ABC	115.026667	87.976190	84.860000	88.779524	75.053333	87.4088889
Apr	ABG	85.996667	58.068571	59.947368	67.826190	74.884286	53.4244444
Apr	ABG	85.996667	58.068571	59.947368	67.826190	74.884286	53.4244444

1-10 of 3,168 rows | 1-8 of 10 columns   Previous 1 2 3 4 5 6 ... 317 Next

Summary of the working data :

```
##        X1            begins_at                    open_price
##  Min.   :   0.0   Min.   :2015-04-28 00:00:00   Min.   :   0.00
##  1st Qu.: 314.8   1st Qu.:2016-07-26 18:00:00   1st Qu.:  13.61
##  Median : 629.5   Median :2017-10-24 12:00:00   Median :  26.33
##  Mean   : 629.5   Mean   :2017-10-25 14:51:25   Mean   :  48.92
##  3rd Qu.: 944.2   3rd Qu.:2019-01-25 18:00:00   3rd Qu.:  56.09
##  Max.   :1259.0   Max.   :2020-04-27 00:00:00   Max.   :1266.56
##   close_price        high_price        low_price         volume
##  Min.   :   0.0052   Min.   :   0.00   Min.   :   0.0   Min.   :        0
##  1st Qu.:  13.6100   1st Qu.:  13.72   1st Qu.:  13.5   1st Qu.:   103282
##  Median :  26.3300   Median :  26.67   Median :  26.0   Median :   421838
##  Mean   :  48.9221   Mean   :  49.43   Mean   :  48.4   Mean   :  1650980
##  3rd Qu.:  56.1200   3rd Qu.:  56.68   3rd Qu.:  55.5   3rd Qu.:  1332573
##  Max.   :1250.0000   Max.   :1274.41   Max.   :1232.0   Max.   :375088650
##    session          interpolated        sname
##  Length:370440      Mode :logical   Length:370440
##  Class :character   FALSE:362193    Class :character
##  Mode  :character   TRUE :8247      Mode  :character
##
```

```
# Only Keeping Date, open_price ,      sname  , interpolated = FALSE

glimpse((data_pro))
## Observations: 370,440
## Variables: 10
## $ X1           <dbl> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...
```

```
## $ begins_at    <dttm> 2015-04-28, 2015-04-29, 2015-04-30, 2015-05-01, 2015-...
## $ open_price   <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, ...
## $ close_price  <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, ...
## $ high_price   <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, ...
## $ low_price    <dbl> 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, 24.8871, ...
## $ volume       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ session      <chr> "reg", "reg", "reg", "reg", "reg", "reg", "reg", "reg"...
## $ interpolated <lgl> TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, TRUE, ...
## $ sname        <chr> "AA", "AA", "AA", "AA", "AA", "AA", "AA", "AA", "AA", ...
```

We will only be Keeping data which satisfies Date, open_price ,   sname (Name of stock) , interpolated = FALSE. Below is final data for further analysis .

begins_at <S3: POSIXct>	open_price <dbl>	sname <chr>
2015-04-28	145.00	AAP
2015-04-29	144.50	AAP
2015-04-30	144.43	AAP
2015-05-01	143.09	AAP
2015-05-04	145.25	AAP
2015-05-05	145.45	AAP

Converting the data of stocks in wide format:

begins_at <S3: POSIXct>	AAP <dbl>	ADS <dbl>	AGN <dbl>	ANET <dbl>	ANTM <dbl>	APD <dbl>	ASR <dbl>	AYI <dbl>	AZO <dbl>
2015-04-28	145.00	303.00	281.90	64.82	150.15	139.4615	151.5279	166.54	691.68
2015-04-29	144.50	301.27	286.35	65.67	155.96	140.3583	150.7279	164.61	690.00
2015-04-30	144.43	299.54	287.27	64.56	151.83	141.7728	149.6180	168.95	682.63
2015-05-01	143.09	300.06	285.17	64.08	151.92	132.8975	145.1780	167.22	675.06
2015-05-04	145.25	299.77	290.19	64.24	153.45	136.7619	146.6580	170.07	683.00
2015-05-05	145.45	300.00	290.45	64.66	154.67	137.0023	143.9980	172.83	682.00

6 rows | 1-10 of 18 columns

Build Data

Fit an AR model to the following stocks, we would be using xts package in R to convert our data set to required time series data format.:

ANTM Anthem, Inc

ANET Arista Networks, Inc

BA The Boeing Company

Creating Time Series Object

```
# wide_data_Main$begins_at  <- as_datetime(wide_data_Main$begins_at)
stocks_ANTM <-xts(wide_data_Main$ANTM,  order.by=as.Date(wide_data_Main$begins_at))
stocks_ANET <- xts(wide_data_Main$ANET, order.by=as.Date(wide_data_Main$begins_at))
stocks_BA <- xts(wide_data_Main$BA,  order.by=as.Date(wide_data_Main$begins_at))


# Data for only 2020 data
wide_data_Main_20 <- wide_data_Main[which(year(wide_data_Main$begins_at) %in% c("2020")),]
# Data for only Rest than 2020 data
wide_data_Main_Old <- wide_data_Main[-which(year(wide_data_Main$begins_at) %in% c("2020")),]


stocks_ANTM_MY  <-xts(wide_data_Main_20$ANTM,  order.by=as.Date(wide_data_Main_20$begins_at))
stocks_ANET_MY <- xts(wide_data_Main_20$ANET, order.by=as.Date(wide_data_Main_20$begins_at))
stocks_BA_MY <- xts(wide_data_Main_20$BA,  order.by=as.Date(wide_data_Main_20$begins_at))


stocks_ANTM_old  <-xts(wide_data_Main_Old$ANTM,  order.by=as.Date(wide_data_Main_Old$begins_at))
stocks_ANET_old <- xts(wide_data_Main_Old$ANET, order.by=as.Date(wide_data_Main_Old$begins_at))
stocks_BA_old <- xts(wide_data_Main_Old$BA,  order.by=as.Date(wide_data_Main_Old$begins_at))
```

Checking the index of xts object:

```
index(stocks_ANTM_MY)

##  [1] "2020-01-02" "2020-01-03" "2020-01-06" "2020-01-07" "2020-01-08"
##  [6] "2020-01-09" "2020-01-10" "2020-01-13" "2020-01-14" "2020-01-15"
## [11] "2020-01-16" "2020-01-17" "2020-01-21" "2020-01-22" "2020-01-23"
## [16] "2020-01-24" "2020-01-27" "2020-01-28" "2020-01-29" "2020-01-30"
## [21] "2020-01-31" "2020-02-03" "2020-02-04" "2020-02-05" "2020-02-06"
## [26] "2020-02-07" "2020-02-10" "2020-02-11" "2020-02-12" "2020-02-13"
## [31] "2020-02-14" "2020-02-18" "2020-02-19" "2020-02-20" "2020-02-21"
```

Checking the data of xts object:

```
coredata(stocks_ANTM_MY)

##        [,1]
##  [1,] 302.67
##  [2,] 293.68
##  [3,] 295.75
##  [4,] 299.20
##  [5,] 300.89
##  [6,] 307.83
##  [7,] 307.94
##  [8,] 306.88
```

View some data of stocks_ANTM_MY xts object, Tail data of the xts object:

```
tail(stocks_ANTM_MY,n=10)
##              [,1]
## 2020-04-14 245.76
## 2020-04-15 249.15
## 2020-04-16 254.51
## 2020-04-17 279.01
```

From xts object we can get value from specific date:

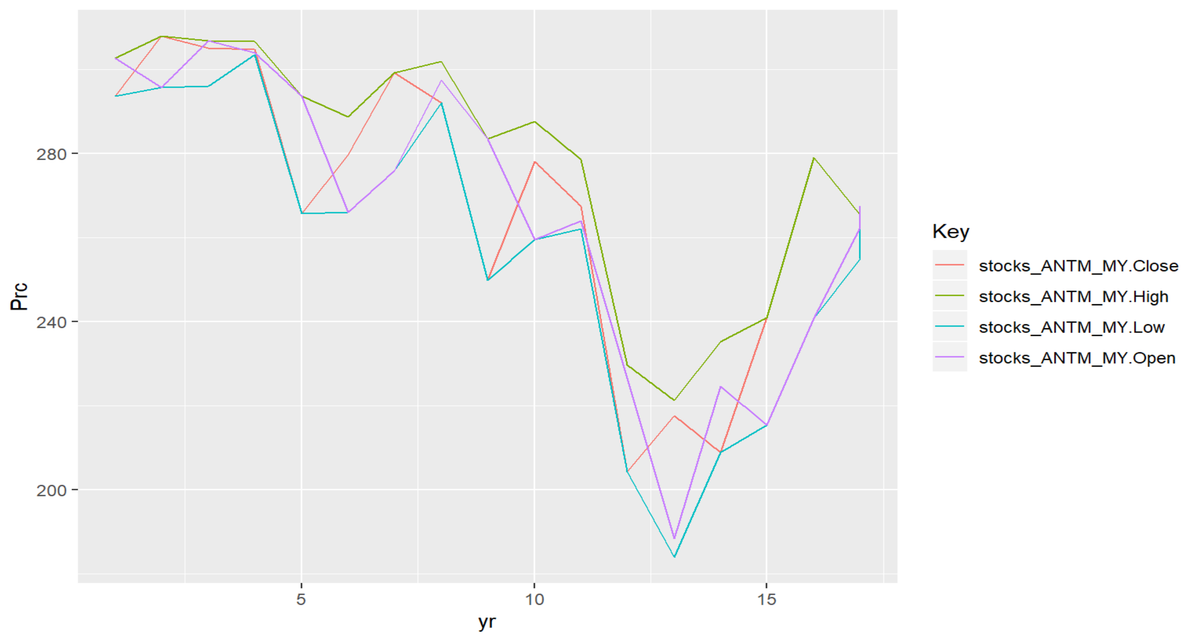
```
stocks_ANTM_MY['2020-04-14']

##              [,1]
## 2020-04-14 245.76
```

By using month on xts object we can get months information in the dataset:

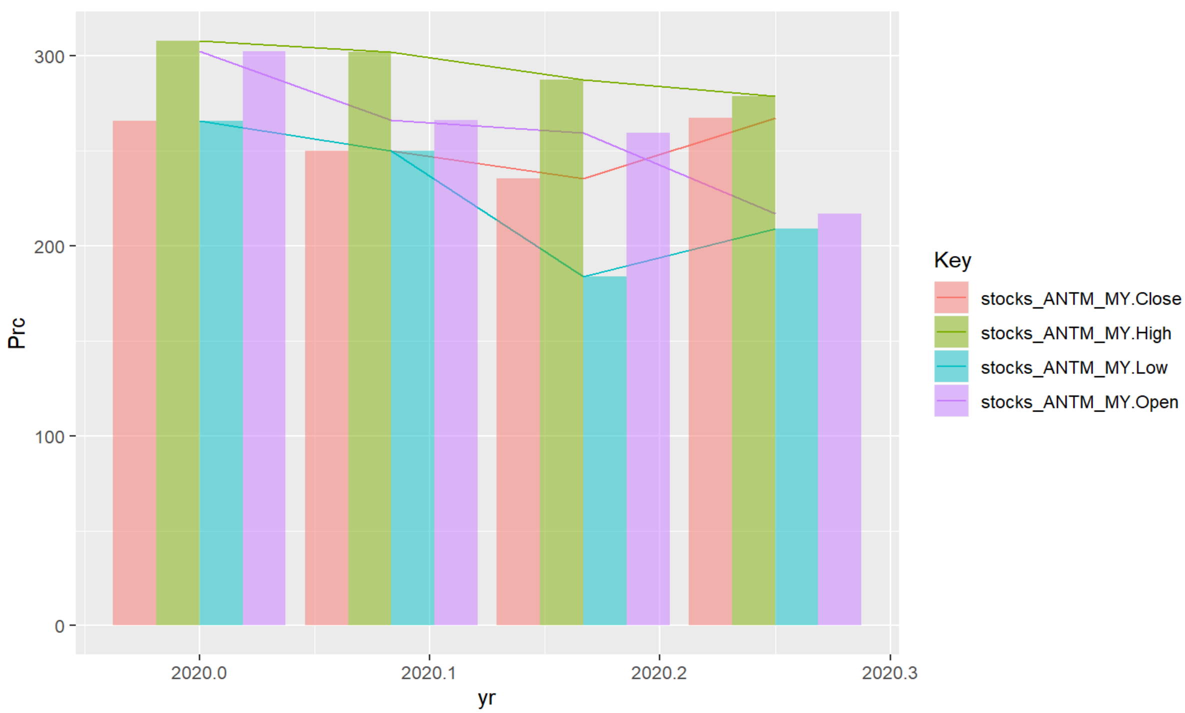
```
month.abb[month(index(stocks_ANTM_MY))]

##  [1] "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan"
## [13] "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Jan" "Feb" "Feb" "Feb"
## [25] "Feb" "Feb" "Feb" "Feb" "Feb" "Feb" "Feb" "Feb" "Feb" "Feb" "Feb" "Feb"
```

Since this is just 4 months of data , ie. Jan 2020 to April 2020 xts object shows the same info:

```
nmonths(stocks_ANTM_MY)

## [1] 4
```



Above graph shows how price of the stocks moved in Year 2020. In the above graph, it shows weekly change in Anthem stock data i.e. open price, close price, high price, low price.



In the above graph, it shows monthly change in Anthem stock data i.e. open price, close price, high price, low price.
Periodicity of Anthem Stocks data

```
periodicity(stocks_ANTM_MY)

## Daily periodicity from 2020-01-02 to 2020-04-27
```

Structure of xts object:

```
str(stocks_ANTM_MY)
## An 'xts' object on 2020-01-02/2020-04-27 containing:
##   Data: num [1:80, 1] 303 294 296 299 301 ...
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
##  NULL
```

With the commands head() and tail() we can see the first and last 6 lines of the base. There are 6 columns with: opening price, maximum and minimum prices, closing price, volume of transactions and adjusted price. Using the command summary() we verify the descriptive statistics of each price series and volume. The command str() returns the object structure. In this case, it's a xts object, a time series.

Time Series Forecasting

Data From Apr 2015 To 2020

'stocks_ANTM' hold data of the Anthem stocks from April 28 2015 to April 27, 2020. Head and tail for the this stock can be seen below:

```
library(forecast)

head(stocks_ANTM)
##              [,1]
## 2015-04-28 150.15
## 2015-04-29 155.96
## 2015-04-30 151.83
## 2015-05-01 151.92
## 2015-05-04 153.45
## 2015-05-05 154.67
```

```
tail(stocks_ANTM)

##              [,1]
## 2020-04-20 262.48
## 2020-04-21 255.00
## 2020-04-22 255.09
## 2020-04-23 264.71
## 2020-04-24 265.48
## 2020-04-27 267.56
```

Summary of the data can be seen as below :

```
summary(stocks_ANTM)
##      Index             stocks_ANTM
##  Min.   :2015-04-28   Min.   :117.0
##  1st Qu.:2016-07-26   1st Qu.:145.2
##  Median :2017-10-24   Median :195.9
##  Mean   :2017-10-25   Mean   :206.5
##  3rd Qu.:2019-01-26   3rd Qu.:262.8
##  Max.   :2020-04-27   Max.   :317.6
```

Structure of the xts object, it has data and Index which can be used to plot and perform data operation.

```
str(stocks_ANTM)
```

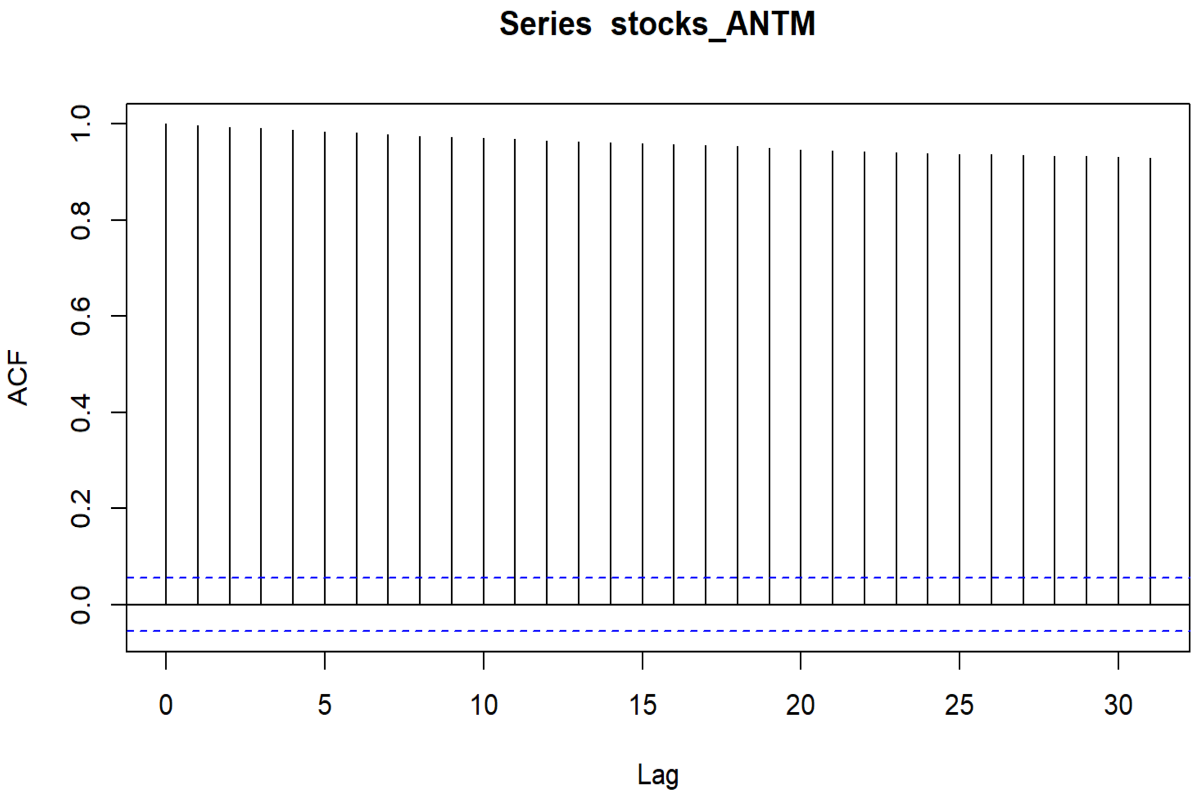
```
## An 'xts' object on 2015-04-28/2020-04-27 containing:
##   Data: num [1:1259, 1] 150 156 152 152 153 ...
##   Indexed by objects of class: [Date] TZ: UTC
##   xts Attributes:
##  NULL
```

Plot below shown how Boing share has moved along the time :

```
plot(stocks_BA)
```

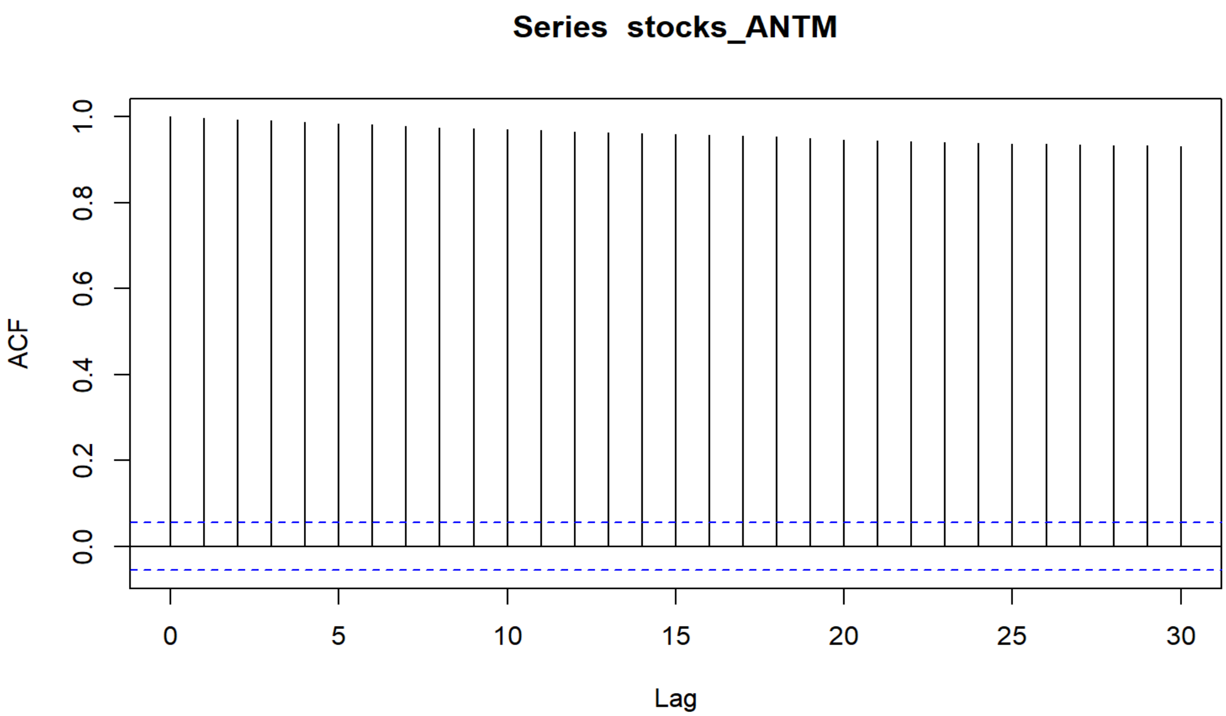
Auto correlation frequency plot for Anthem data from for first 30 lags, going be 1 lag default check.

```
acf(stocks_ANTM)
```



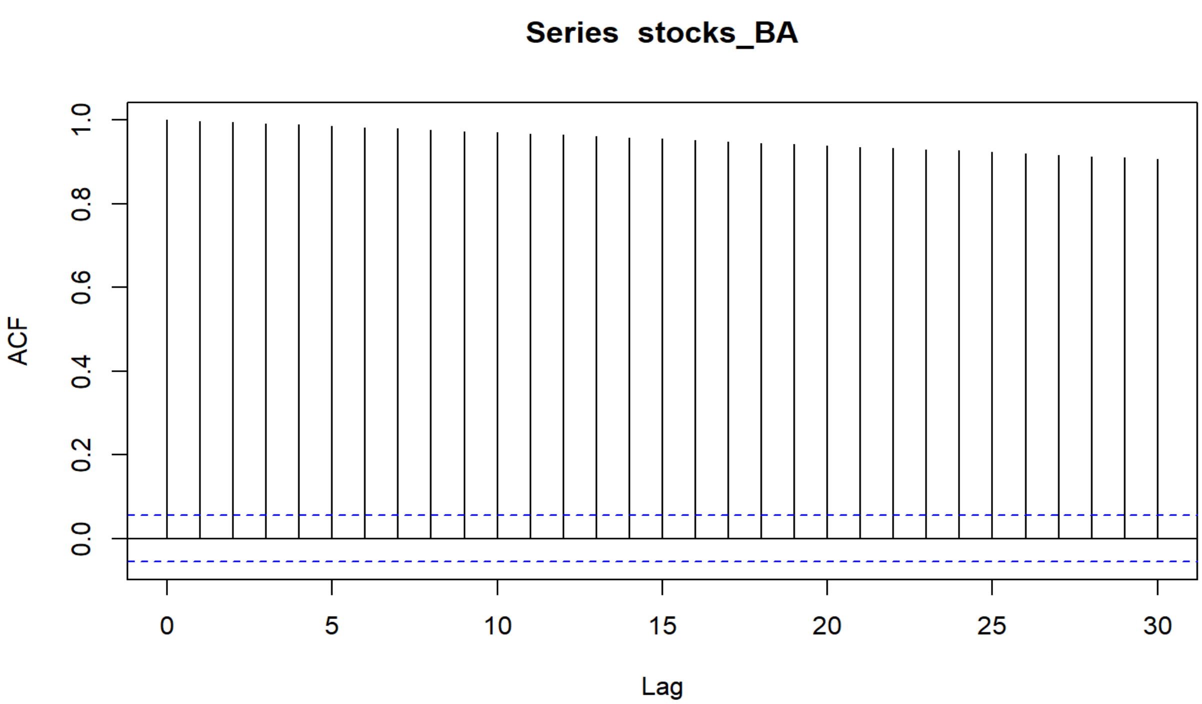
ACF Plot for Anthem:

```
acf_ANTM = acf(stocks_ANTM,lag.max = 30)
```



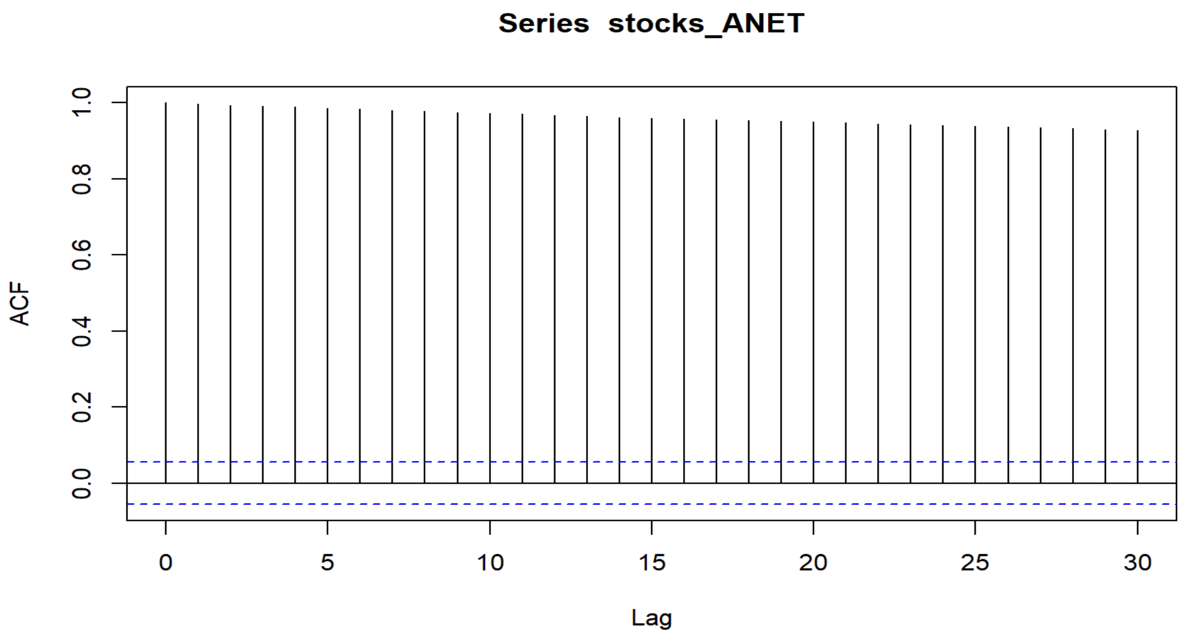
ACF Plot for Boing:

```
acf_BA = acf(stocks_BA,lag.max = 30)
```



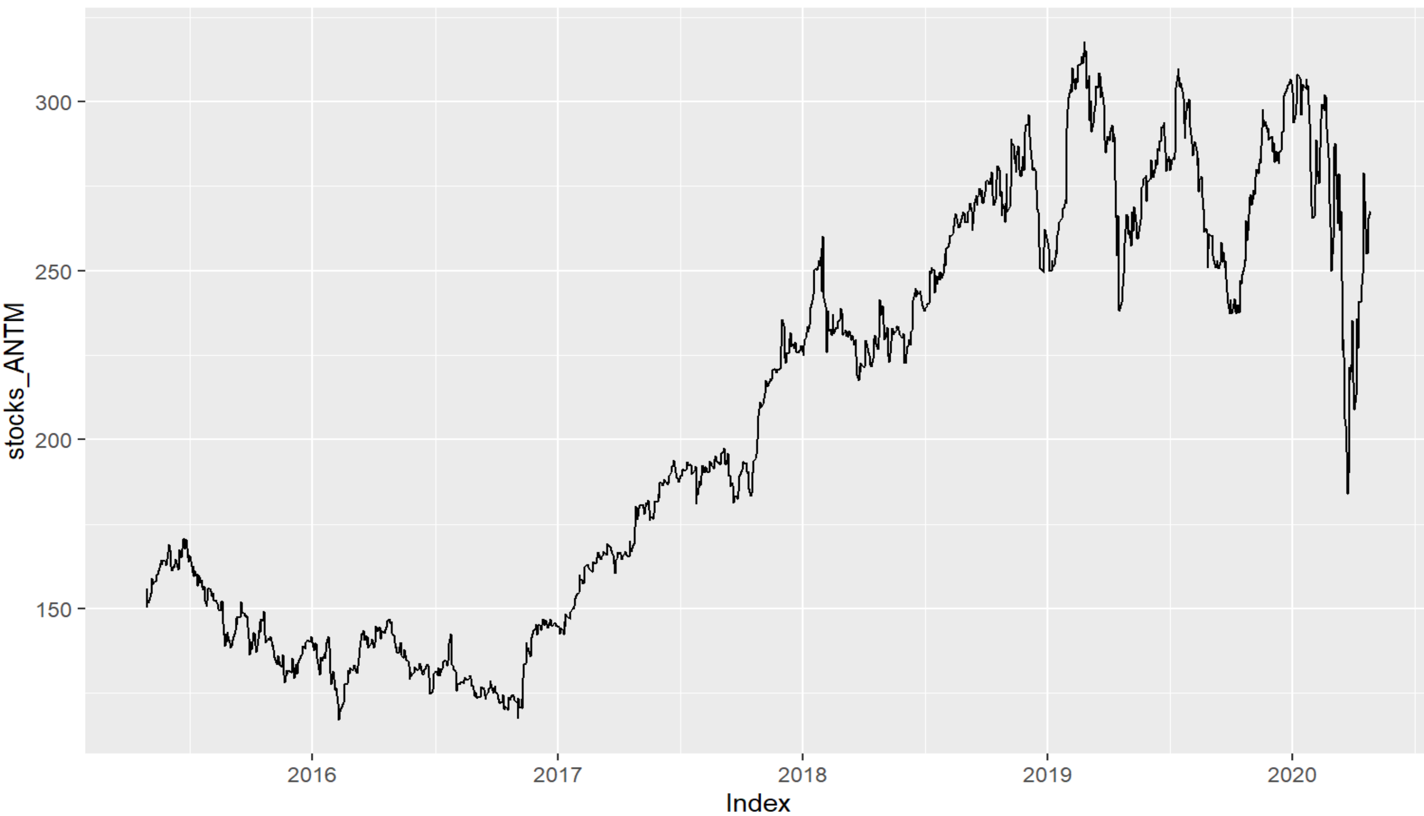
ACF Plot for ANET:

```
acf_ANET = acf(stocks_ANET,lag.max = 30)
```



Follow shows how Anthem data is distributed over years :

```
autoplot(stocks_ANTM)
```



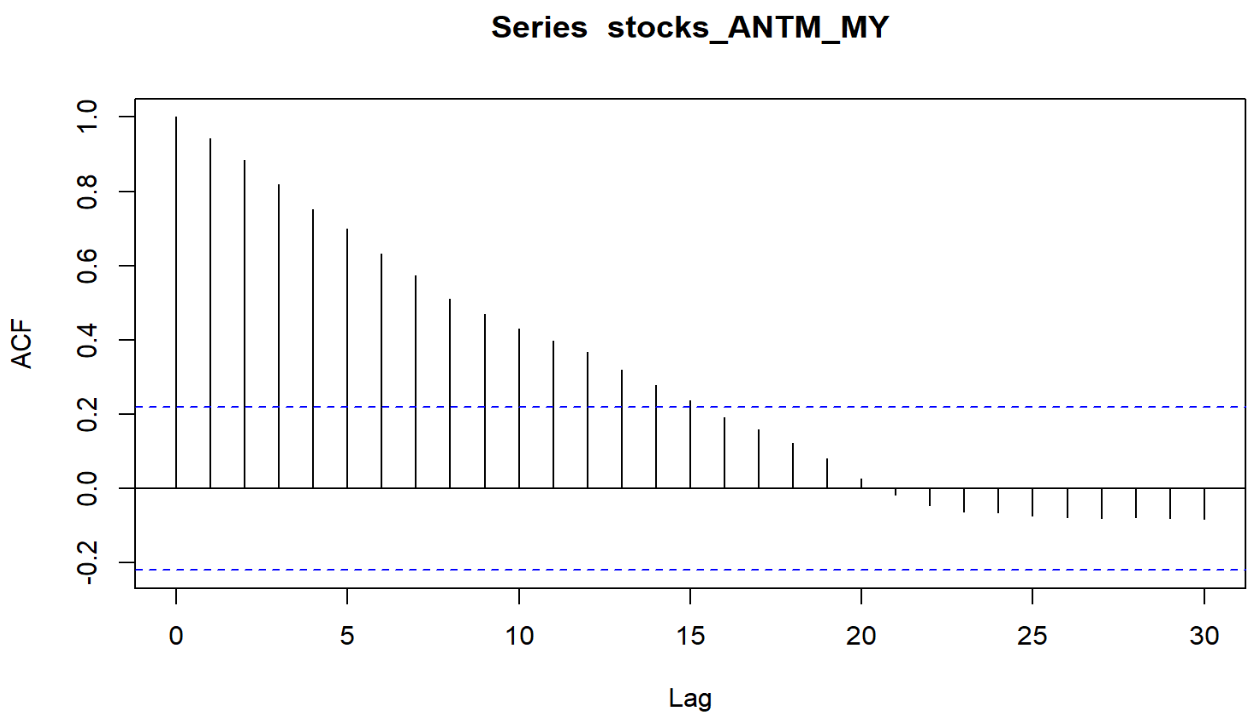
Frequency of data is 1 , which mean that we have data mostly from each day in the data set.

```
frequency(stocks_ANTM)

## [1] 1
```

Current Year Data

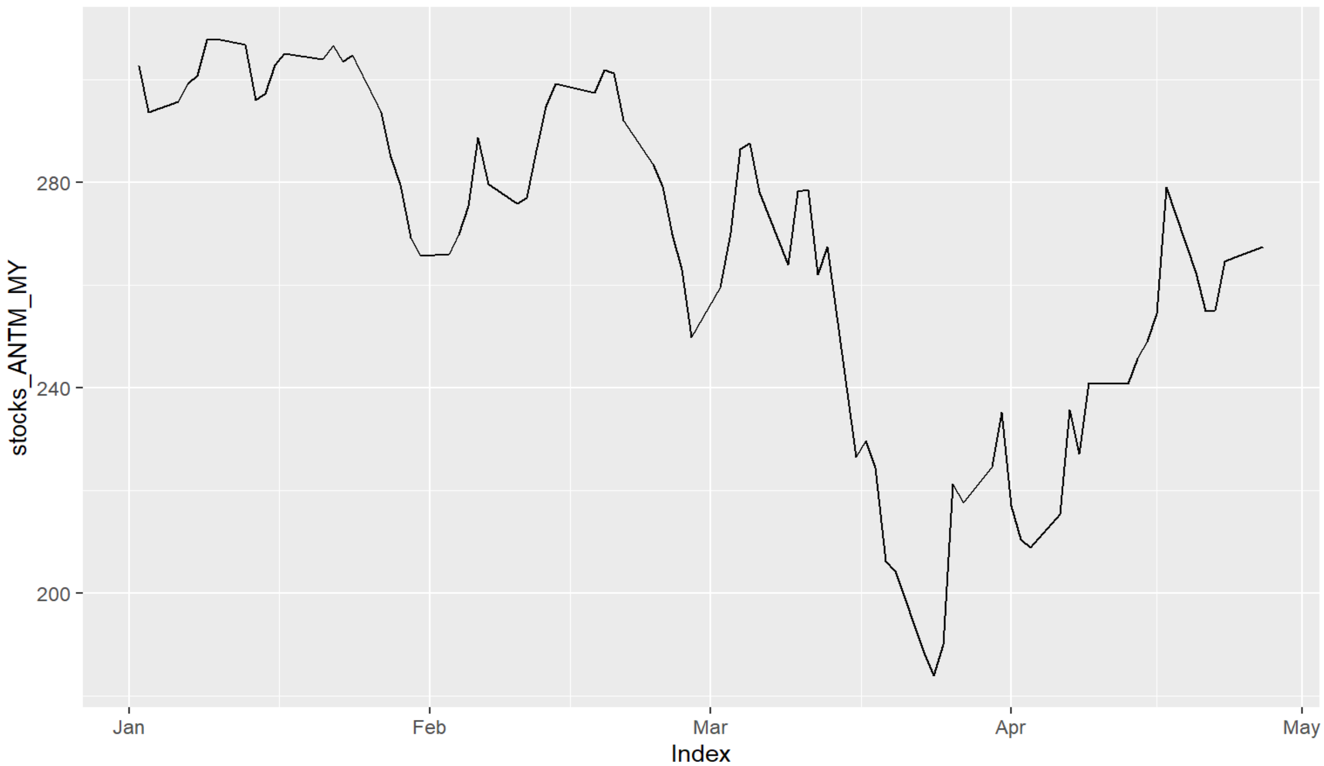
Now we will only analyze the data set which only contains data for 2020:
Plot for 2020 Data only
ACF plots below shows it's not a stationary, as it shows some seasonality in the data in latter part of data. This seasonality is due to COVID-19 impact on the stocks. If you compare the same plot for full ANTHEM data , its create that this stock prices follows some study random walk model and is non-stationary data.

```
acf(stocks_ANTM_MY,lag.max = 30)
```



Below graph shows how ANTHEM Stock moved in the year 2020. From the tred , its very easy to see sharp decline in March and little recovery in April 2020.

```
autoplot(stocks_ANTM_MY)
```



Head of Data from 2020:

```
head(stocks_ANTM_MY)
##              [,1]
## 2020-01-02 302.67
## 2020-01-03 293.68
## 2020-01-06 295.75
## 2020-01-07 299.20
## 2020-01-08 300.89
## 2020-01-09 307.83
```

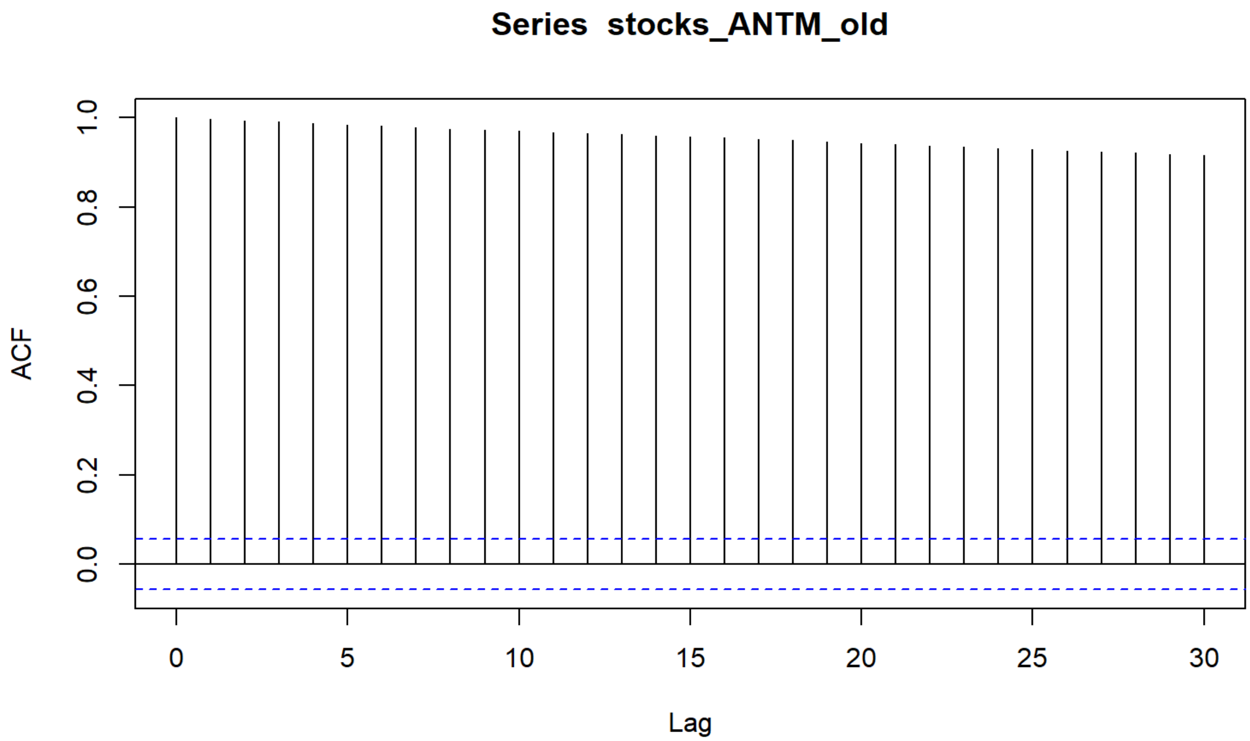
Frequency of the data is clearly daily.

```
frequency(stocks_ANTM_MY)

## [1] 1
```

Data Till Dec 2019

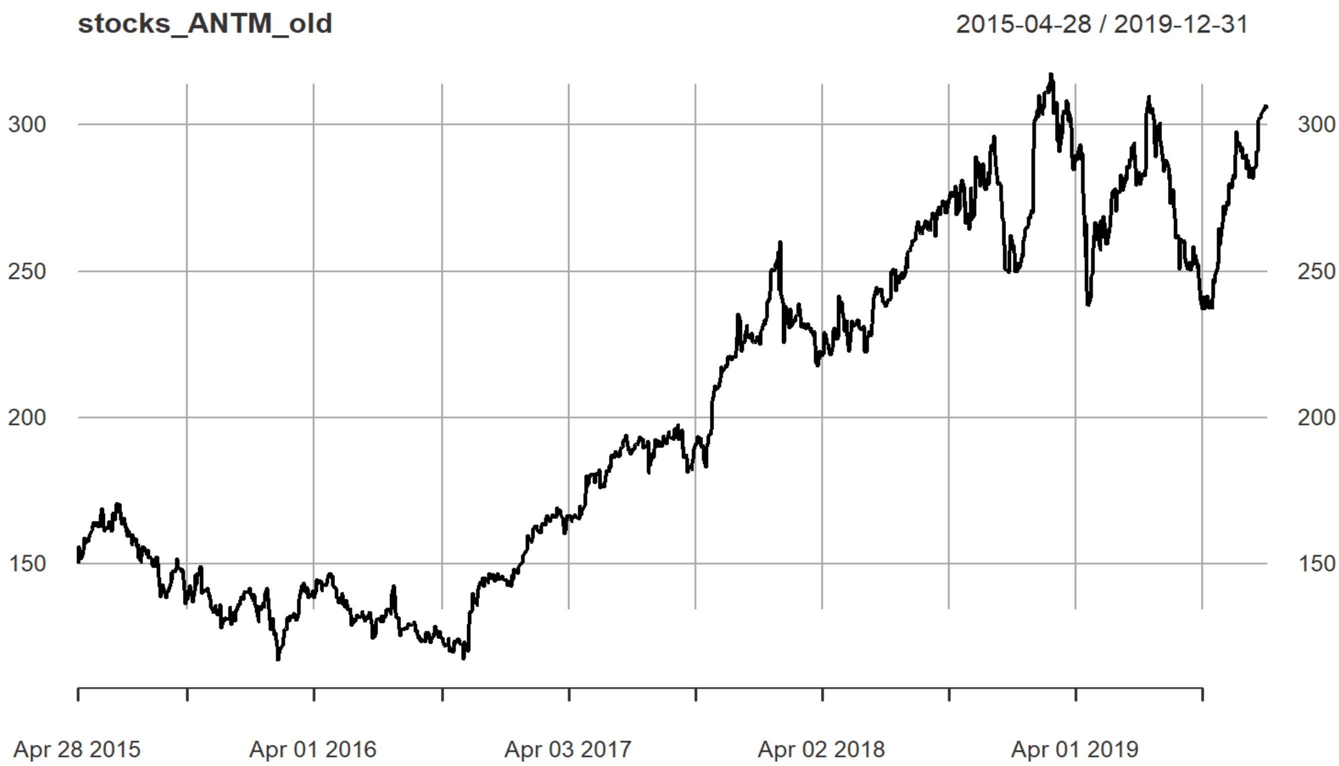
Now Let's check the same stocks data only until Dec 2019.
For rest of the data before 2020:
Auto correlation plot doesn't show any difference from whole data , we will be using this data set to build our model and predict stock price in Year 2020 for stock ANTM.

```
acf(stocks_ANTM_old)
```



Data flow over the year until dec 2019.

```
plot(stocks_ANTM_old)
```




```
head(stocks_ANTM_old)
##              [,1]
## 2015-04-28 150.15
## 2015-04-29 155.96
## 2015-04-30 151.83
## 2015-05-01 151.92
## 2015-05-04 153.45
## 2015-05-05 154.67
```

Frequency of the data is daily.

```
frequency(stocks_ANTM_old)

## [1] 1
```

White Noise & Autoplots

All the above plots doesn't show any white noise, The white noise (WN) and random walk (RW) models are very closely related. However, only the RW is always non-stationary. And we do see that our dataset is supporting random walk and its not stationary.
The ACF plots test if an individual lag autocorrelation is different than zero. An alternative approach is to use the Ljung-Box test, which tests whether any of a group of autocorrelations of a time series are different from zero.
In essence it tests the "overall randomness" based on a number of lags. If the result is a small p-value than it indicates the data are probably not white noise.
For 2020 Data we will see if it's while noise or not:

```
Box.test(wide_data_Main_20$ANTM, lag = 30, fitdf = 0, type = "Lj")
##
##  Box-Ljung test
##
## data:  wide_data_Main_20$ANTM
## X-squared = 480.01, df = 30, p-value < 2.2e-16

Box.test(wide_data_Main$ANTM, lag = 4, fitdf = 0, type = "Lj")
##
##  Box-Ljung test
##
## data:  wide_data_Main$ANTM
## X-squared = 4976, df = 4, p-value < 2.2e-16
```

Here, we perform a Ljung-Box test on the first 24 lag autocorrelations. The resulting p-value is significant at $p < .001$, so this supports our ACF plot consideration above where we stated it's likely this is not purely white noise and that some time series information exists in this data.

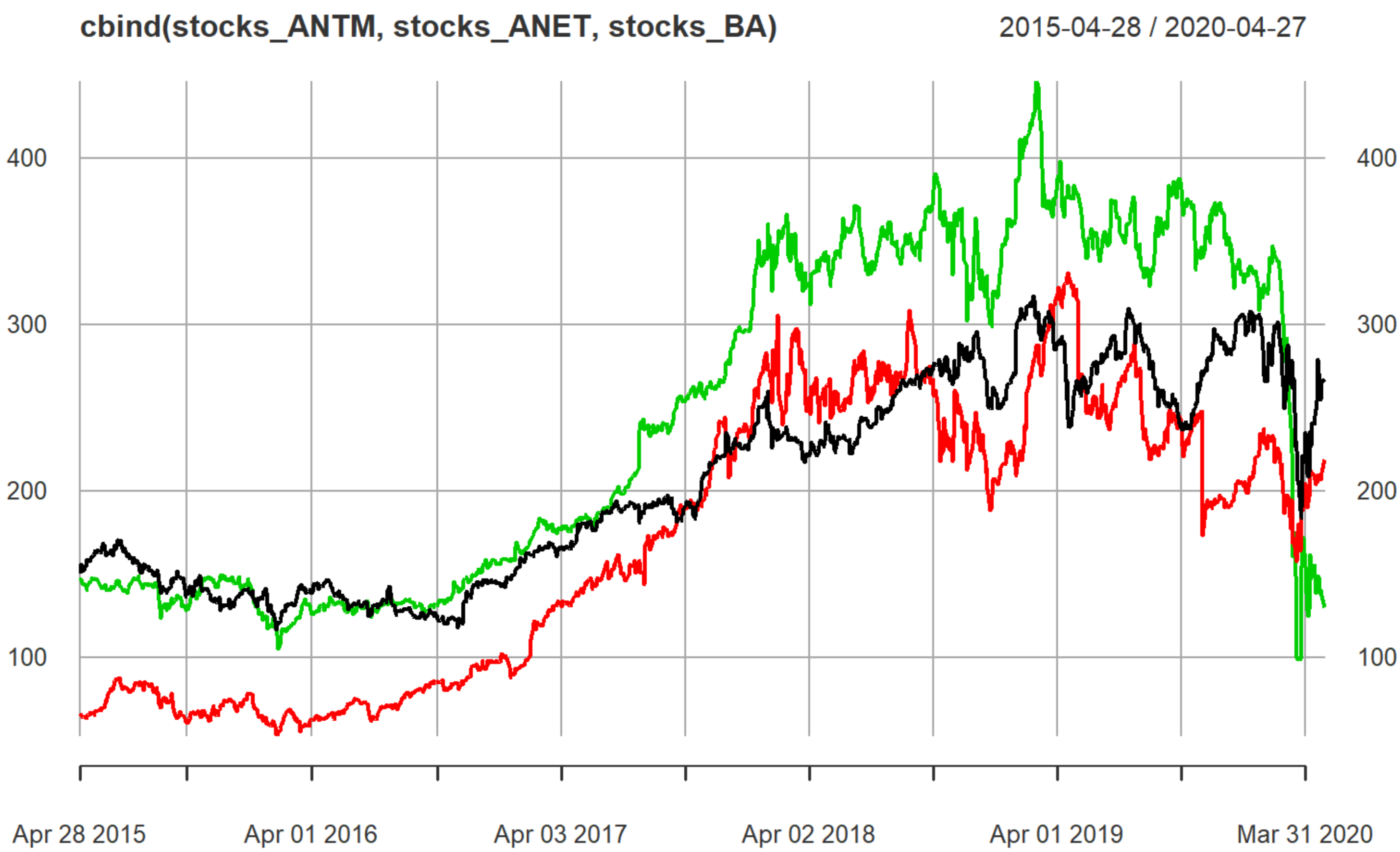
Following Plots of all the stocks together shows correlation in the stock, we will see how much correlation exists among these stocks:

```
plot(cbind(stocks_ANTM,stocks_ANET,stocks_BA))
```

BLACK- >ANTM

RED-> ANET

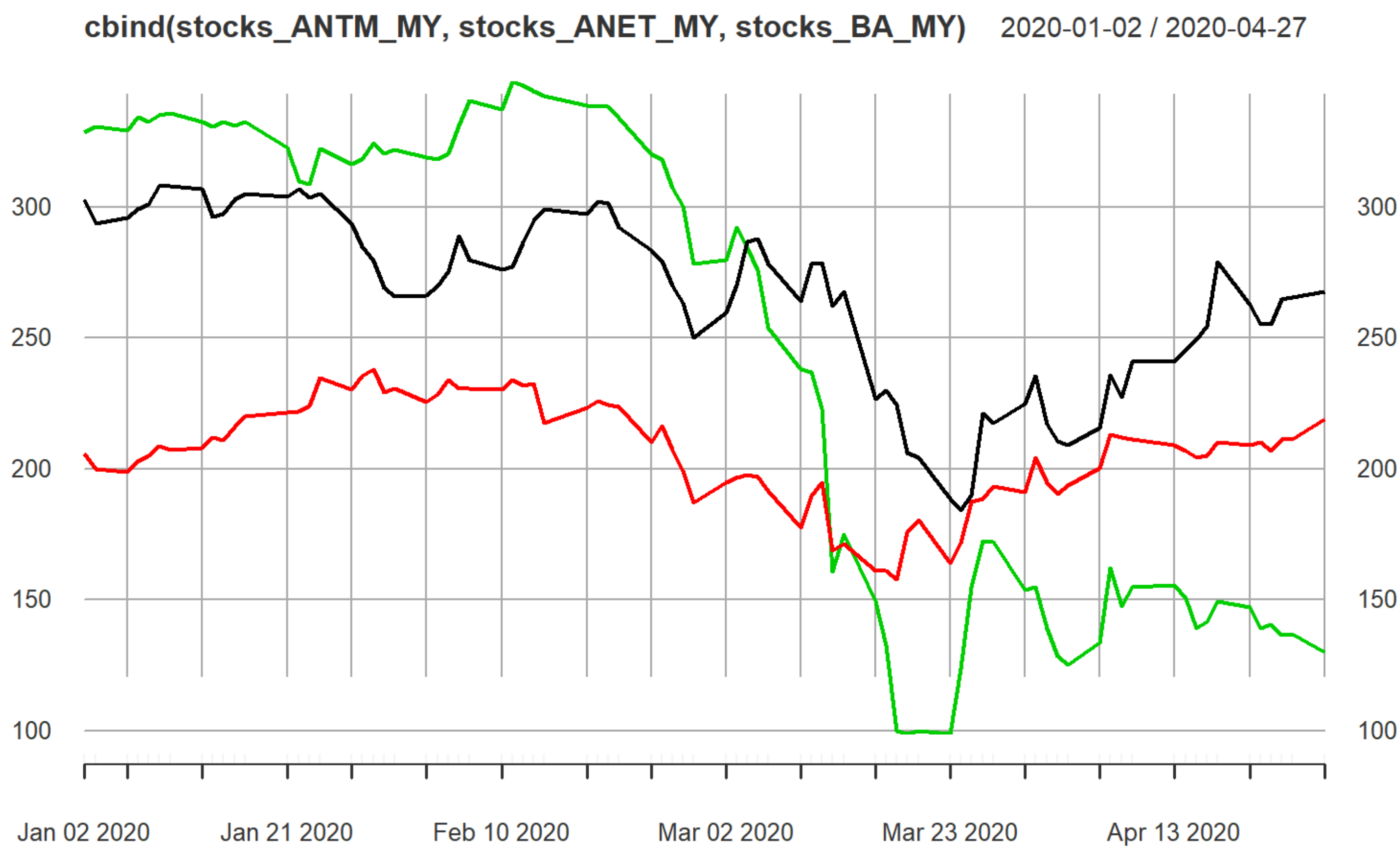
GREEN -> BA



Correlation of the 2020 only data is as below:

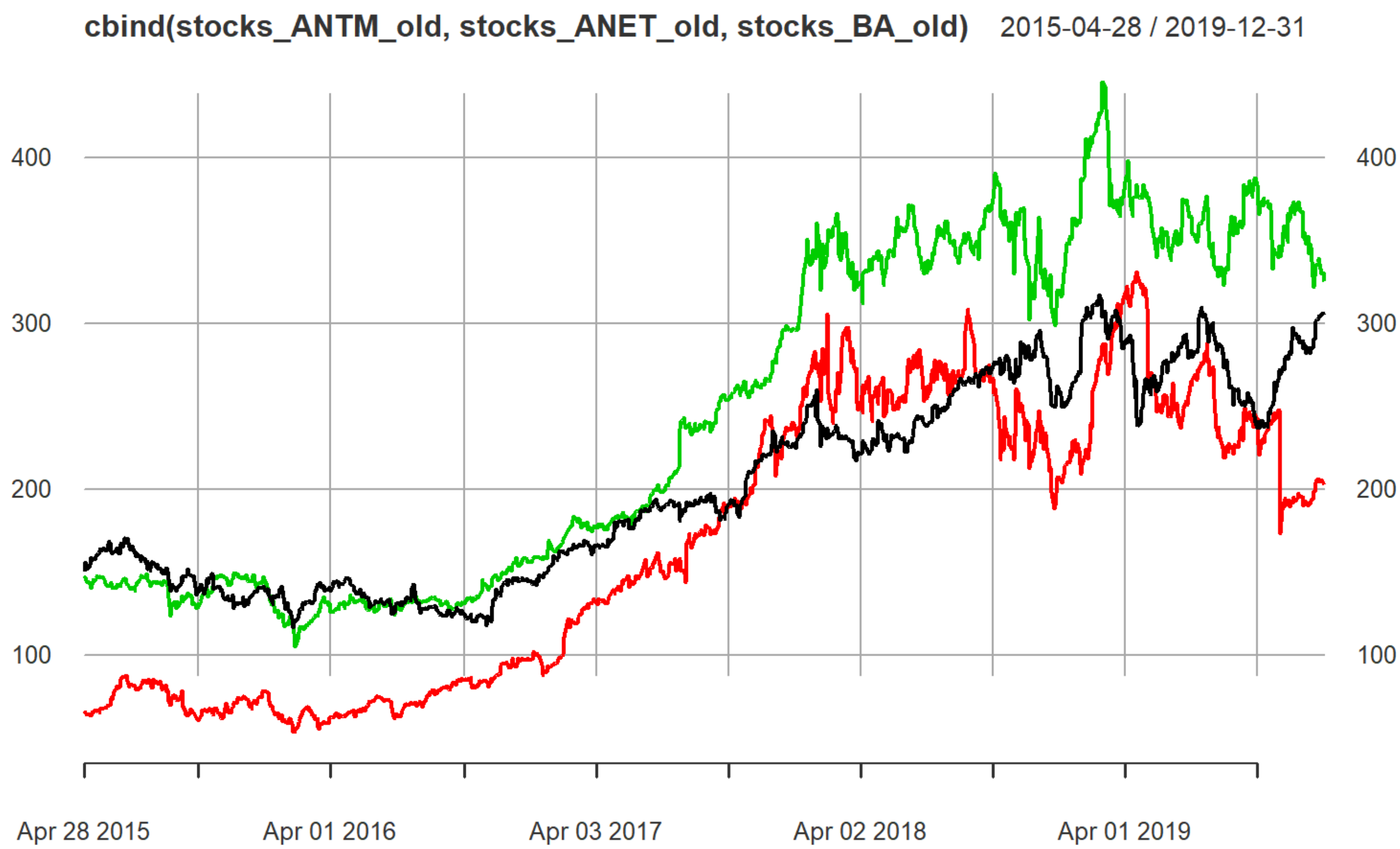
```
plot(cbind(stocks_ANTM_MY,stocks_ANET_MY,stocks_BA_MY))
```

BLACK- >ANTM
RED-> ANET
GREEN -> BA



Same plot for data only till Dec 2019.

```
plot(cbind(stocks_ANTM_old,stocks_ANET_old,stocks_BA_old))
```

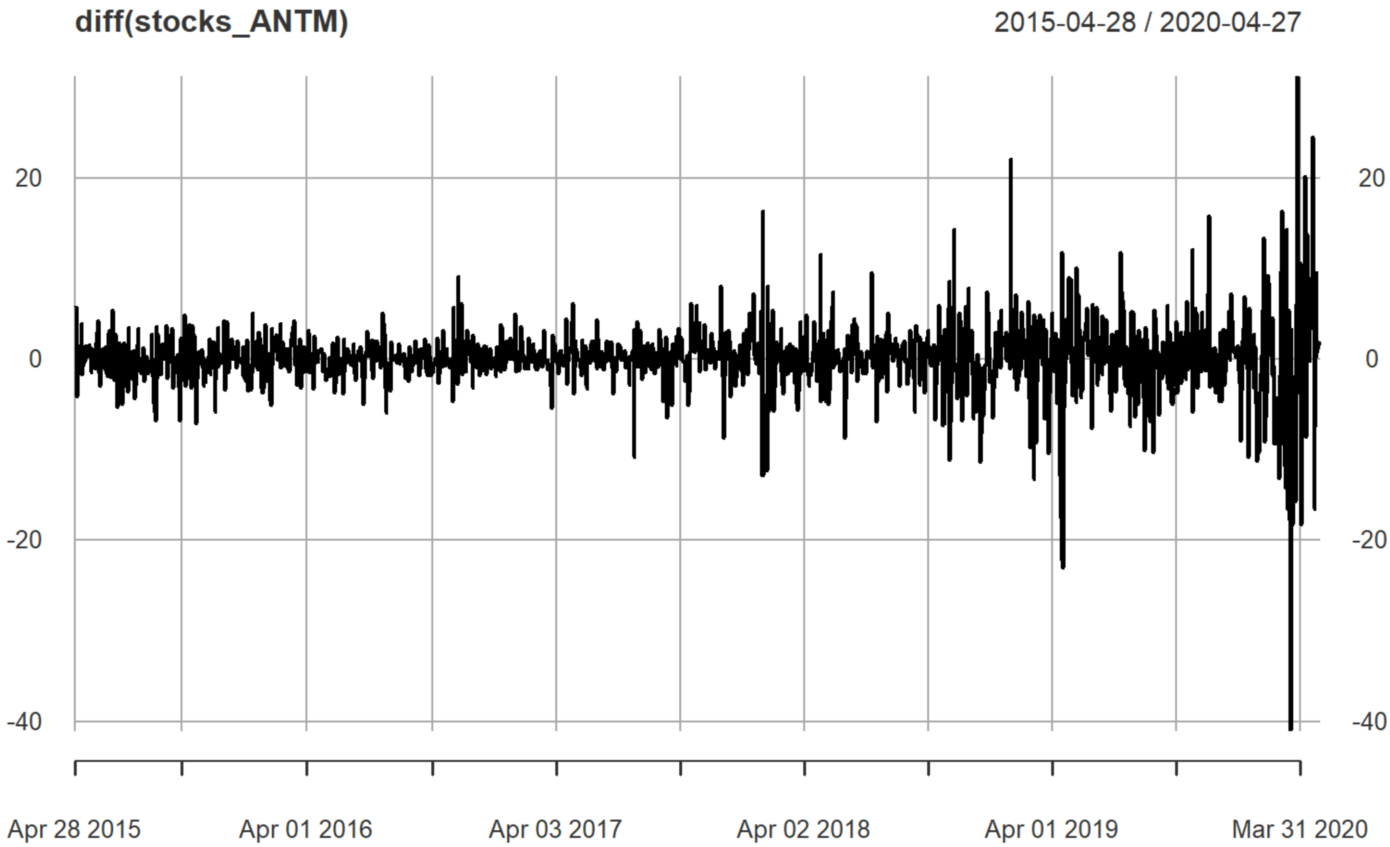


These plots suggest that these slots the stocks improved from their position from mid of 2016 though 2018, and then it remained constant in progress until Late 2019 and early 2020.
The trend is the long-term increase or decrease in the data. There is an increasing trend in the cement data. the seasonal pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. The daily data of the stocks_ANTM doesn't show any seasonality in the graph.
the cycle occurs when the data exhibit rises and falls that are not of a fixed period. These fluctuations are usually due to economic conditions and are often related to the "business cycle". We can see a few cycles in our in stocks_ANTM data from 2015 to 2018 and then in 2020 we have sudden drop due to covid 19.

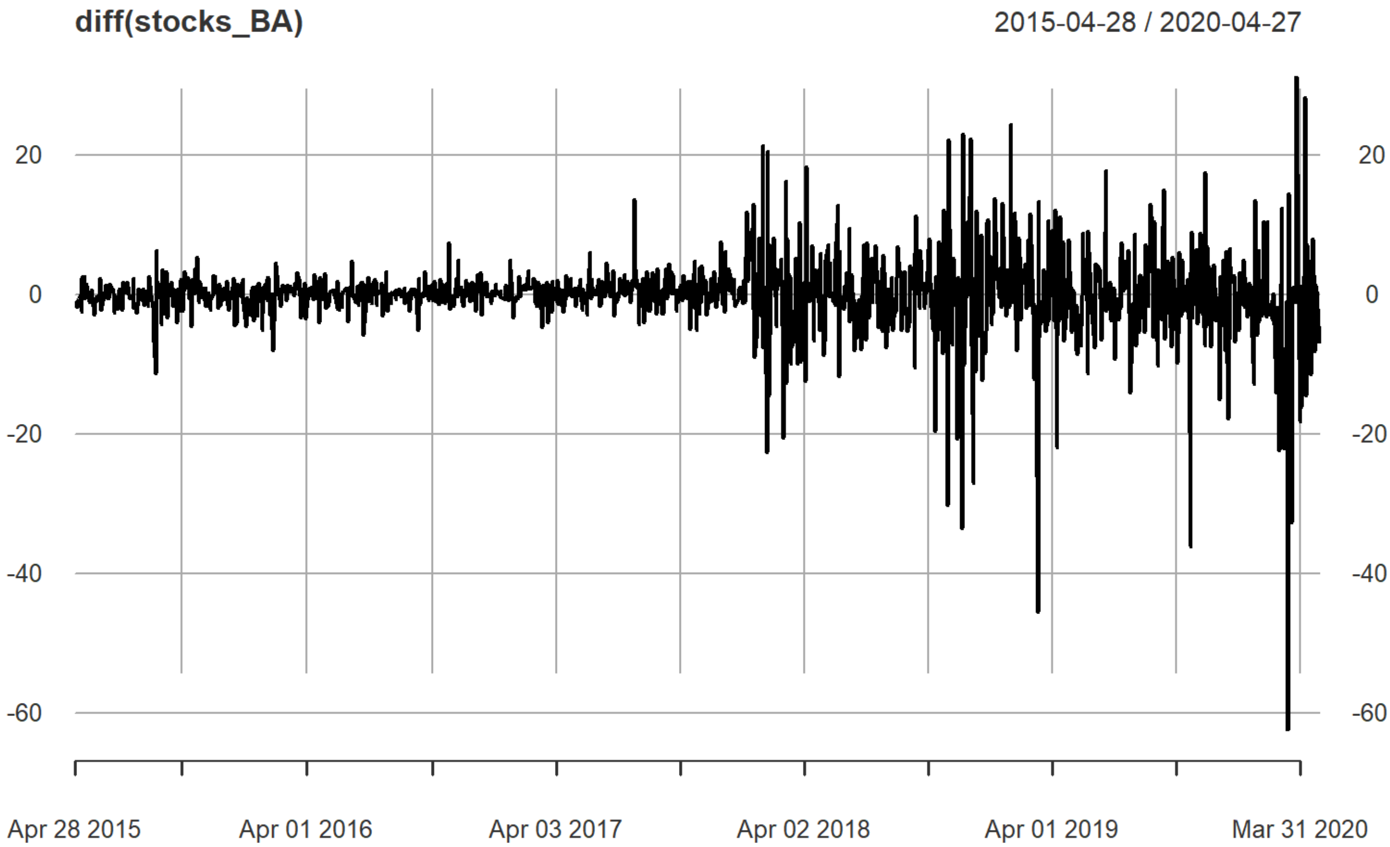
Autocorrelation Of Time Series

Another way to look at time series data is to plot each observation against another observation that occurred sometime previously. For example, we could plot yt against yt-1 . This is called a lag plot because you are plotting the time series against lags of itself.
Below lag plots shows how data was very moving along mean zero but April 2019 and Mar 2020 shows some spiked and random changes in the lags. The Lags in the Latter month of March 2020 is more prominent and can be seen very clearly how it has impacted the stock price.

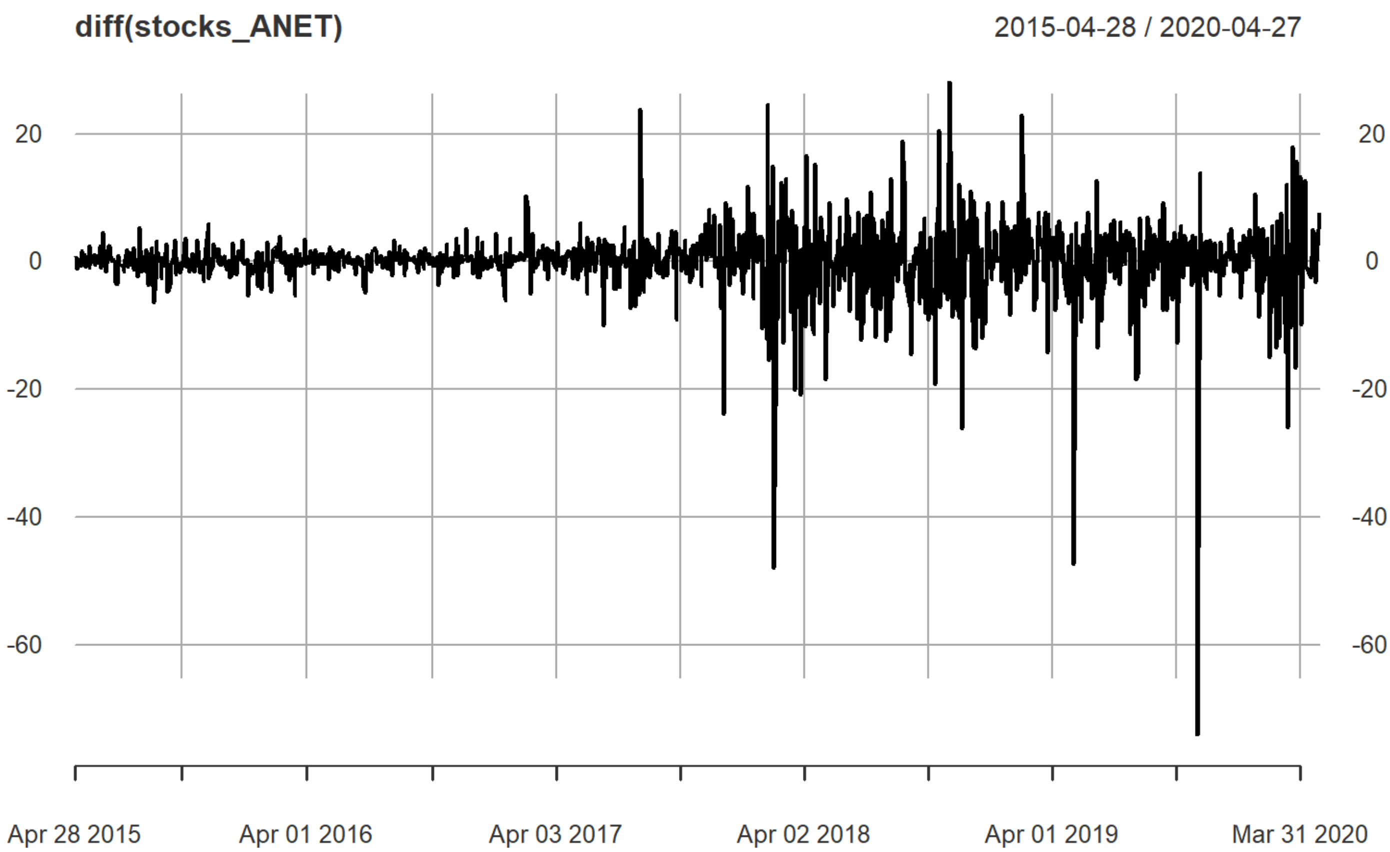
```
plot(diff(stocks_ANTM))
```




```
plot(diff(stocks_BA))
```

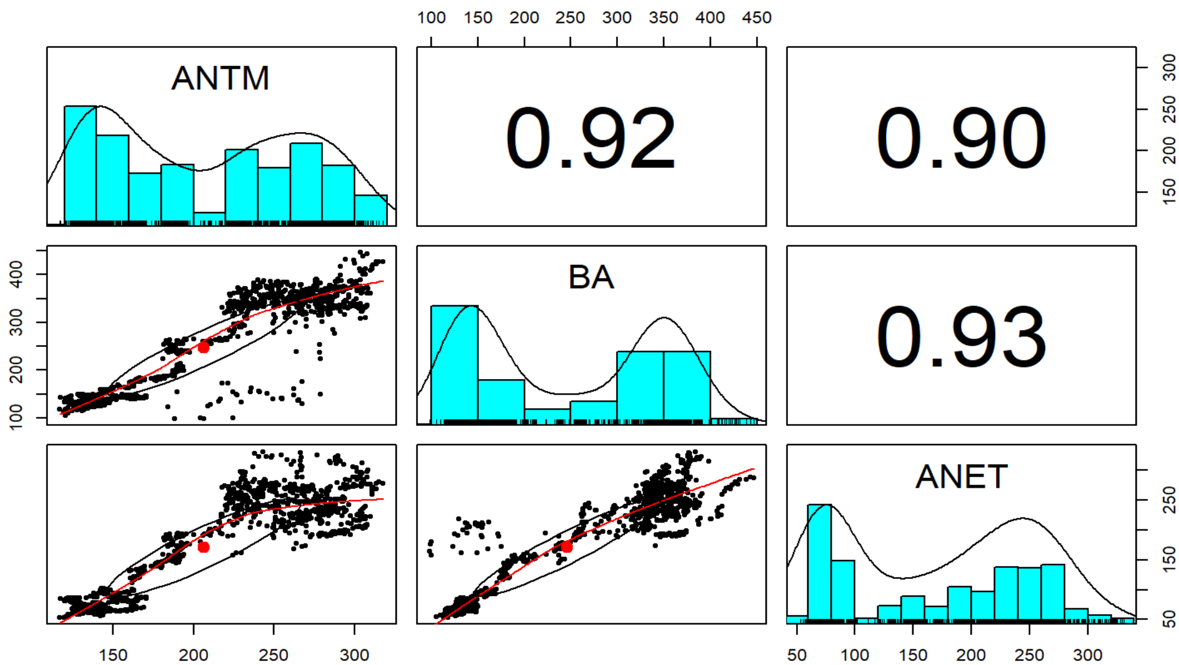



```
plot(diff(stocks_ANET))
```



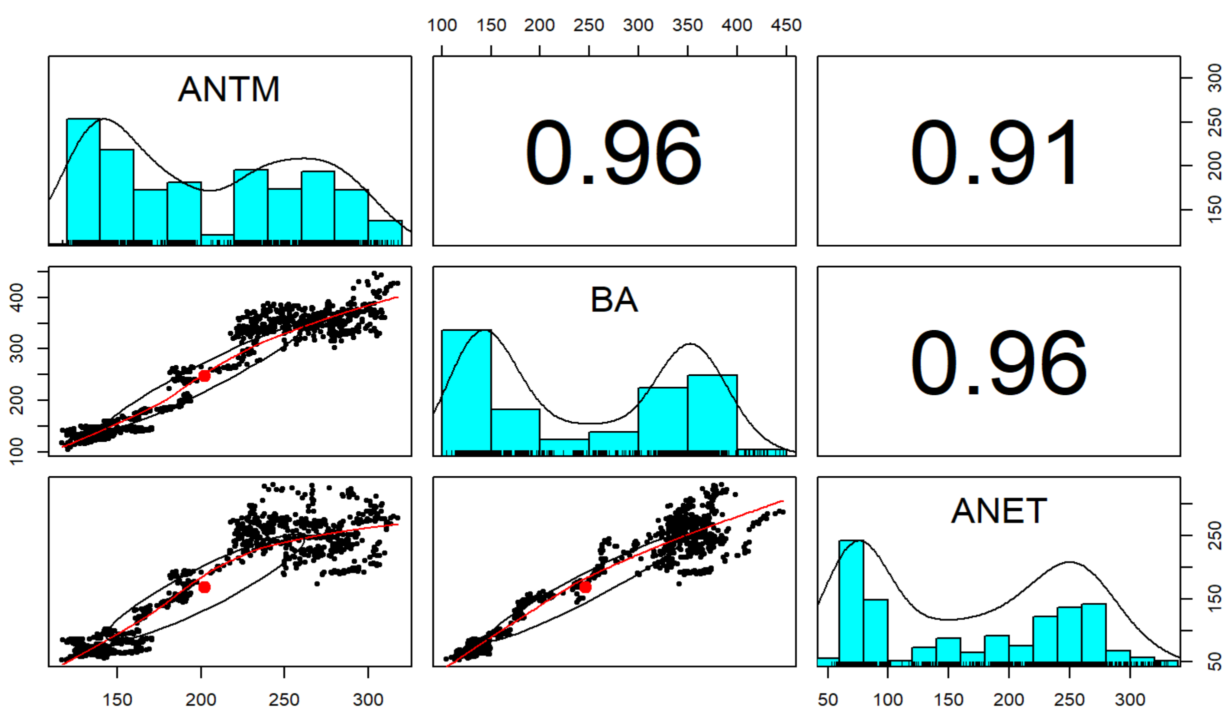
Correlation in the stocks is very high and COR function also shows the same. Correlation on Full Stock data from 2015- 2020:

```
# Correlation Between the stocks
psych::pairs.panels(as.matrix(wide_data_Main[,c('ANTM','BA','ANET')]))
```



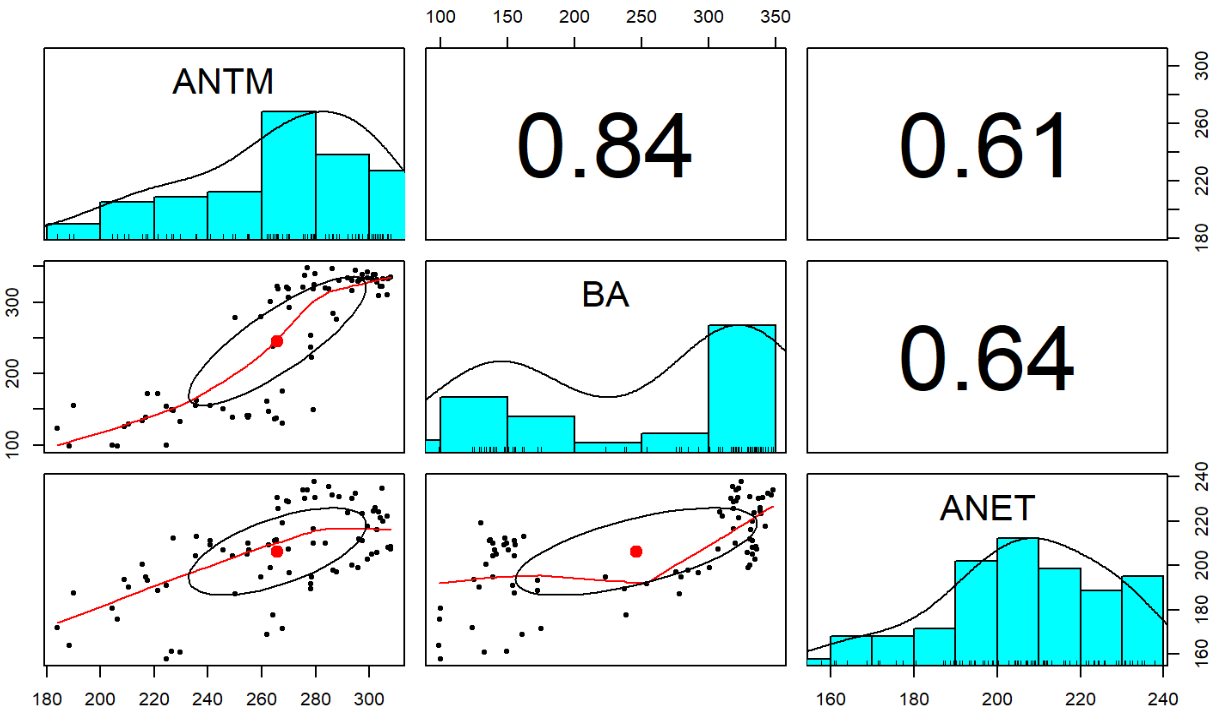
Correlation among data before 2020:

```
psych::pairs.panels(as.matrix(wide_data_Main_Old[,c('ANTM','BA','ANET')]))
```



Correlation only for 2020 data:

```
psych::pairs.panels(as.matrix(wide_data_Main_20[,c('ANTM','BA','ANET')]))
```




```
class(stocks_ANTM)

## [1] "xts" "zoo"
```

White Noise : Time series that show no autocorrelation are called “white noise”. Above plots shows that its of type of Random Walk model , and the (MA Model) Moving Average model should give better estimates of this index.

BUILDING MODEL

Fitting Data

For a given time series x we can fit the autoregressive (AR) model using the arima() command and setting order equal to c(1, 0, 0). Note for reference that an AR model is an ARIMA(1, 0, 0) model.
Fitting Full data :
Fitting Anthem full year data from 2015 to 2020 using AR Model, and MA Model.

```
# Fit with Full Data

AR_ANTM <- arima(stocks_ANTM, order  = c(1,0,0))
MA_ANTM <- arima(stocks_ANTM, order  = c(0,0,1))
AR_ANTM_fit <- as.ts(stocks_ANTM) - resid(AR_ANTM)
MA_ANTM_fit <- as.ts(stocks_ANTM) - resid(MA_ANTM)

summary(AR_ANTM)
```

```
##
## Call:
## arima(x = stocks_ANTM, order = c(1, 0, 0))
##
## Coefficients:
##          ar1  intercept
##       0.9978   222.5894
## s.e.  0.0018    45.1308
##
## sigma^2 estimated as 16.88:  log likelihood = -3568.12,  aic = 7142.25
##
## Training set error measures:
##                     ME     RMSE      MAE         MPE    MAPE     MASE
## Training set 0.0542719 4.108244 2.529343 -0.01210249 1.19899 1.002039
##                    ACF1
## Training set -0.0209907
```

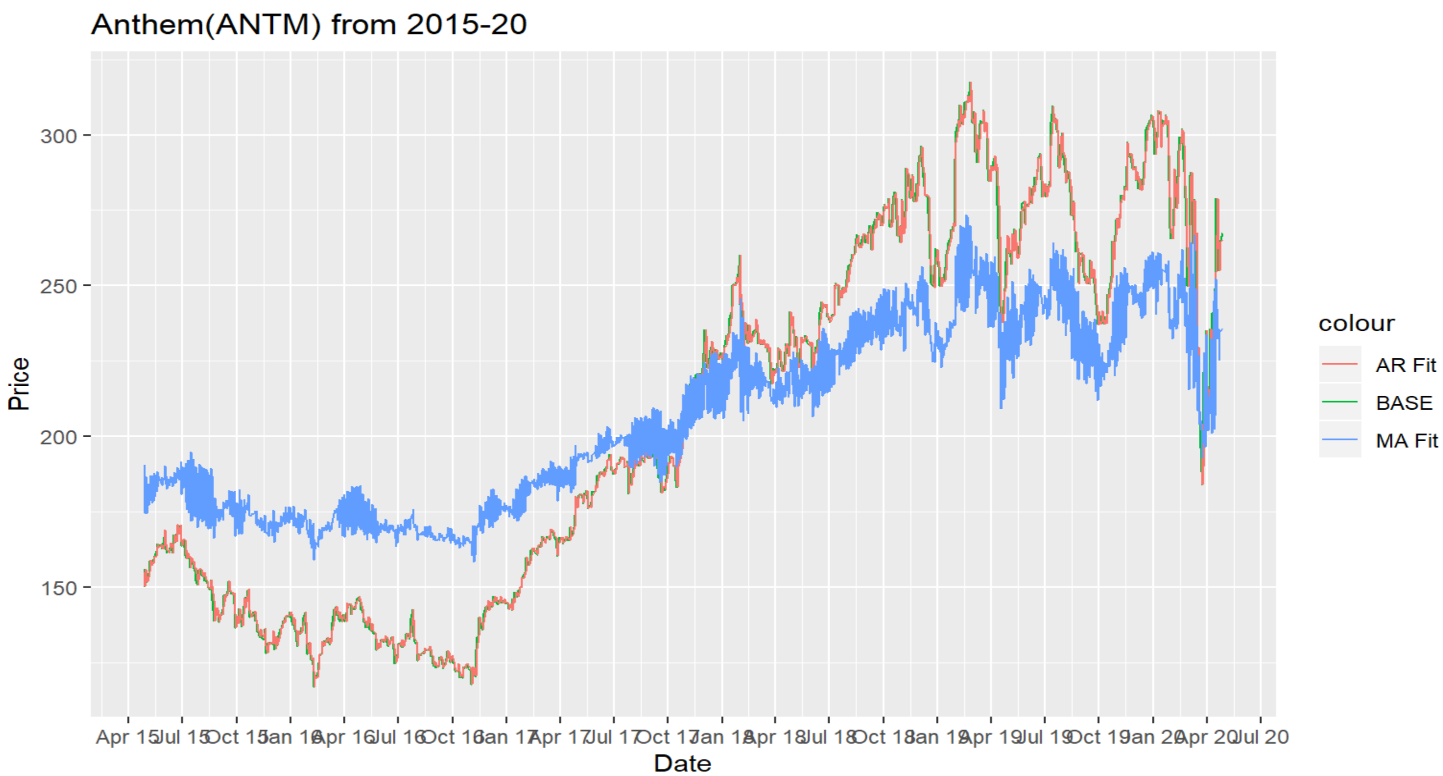
```
summary(MA_ANTM)
##
## Call:
## arima(x = stocks_ANTM, order = c(0, 0, 1))
##
## Coefficients:
##          ma1  intercept
##       0.9678   206.4969
## s.e.  0.0058     1.7300
##
## sigma^2 estimated as 973.9:  log likelihood = -6119.59,  aic = 12245.19
##
## Training set error measures:
##                      ME     RMSE      MAE       MPE     MAPE     MASE      ACF1
## Training set 0.01566924 31.20706 27.96412 -4.741534 15.01558 11.07842 0.9108069
```

Sigma^2 is equivalent of R2 shows AR Model is explaining more variability of the data compared with MA model.

sigma^2 estimated as 16.88

sigma^2 estimated as 973.9:

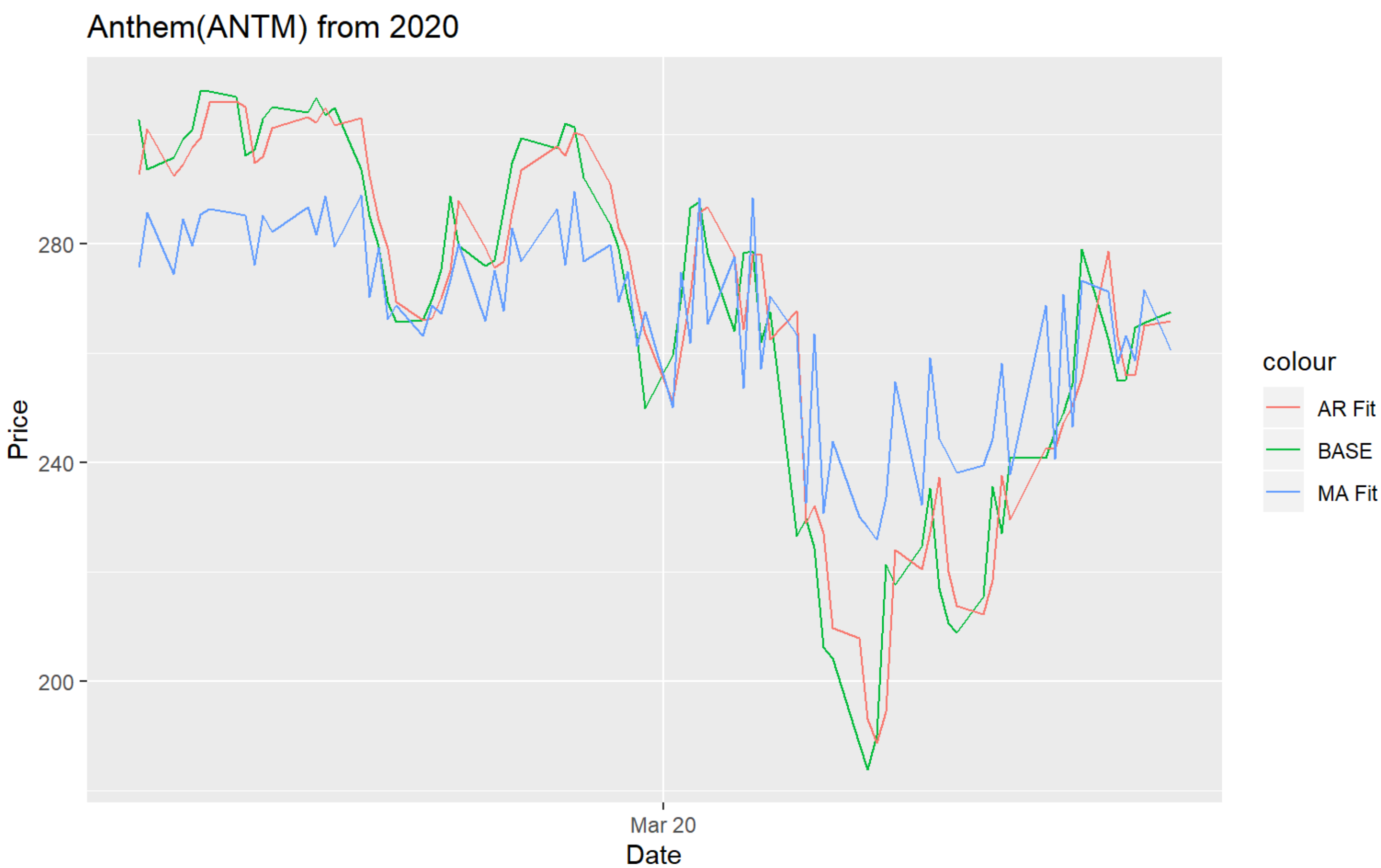
Below plot suggest how actual data fits with AR Model and MA model. WE can see that AR Model data fits very well with actual data, where as Any model data does not fits well with the base model data.



Fit With Only Data After 2020:

We will use the same AR and MA model , and we will try to predict stock data for year 2020, from the fitted model of the data tell December 2019.

```
# plot.ts(stocks_ANTM_MY)

AR_ANTM_MY <- arima(stocks_ANTM_MY, order  = c(1,0,0))
MA_ANTM_MY <- arima(stocks_ANTM_MY, order  = c(0,0,1))
AR_ANTM_MY_fit <- as.ts(stocks_ANTM_MY) - resid(AR_ANTM_MY)
MA_ANTM_MY_fit <- as.ts(stocks_ANTM_MY) - resid(MA_ANTM_MY)
# points(AR_ANTM_MY_fit, type = "l", col = 4, lty = 2)
# points(MA_ANTM_MY_fit, type = "l", col = 3, lty = 3)
ggplot(stocks_ANTM_MY, aes(x = index(stocks_ANTM_MY))) +
  geom_line(aes(y= coredata(stocks_ANTM_MY) , color="BASE")) +
  geom_line(aes(y = AR_ANTM_MY_fit, color = "AR Fit")) +
   geom_line(aes(y = MA_ANTM_MY_fit, color = "MA Fit"))+
   ggtitle("Anthem(ANTM) from 2020") +
  scale_x_date(date_labels = "%b %y", date_breaks = "3 months")+
  xlab("Date") + ylab("Price")
```

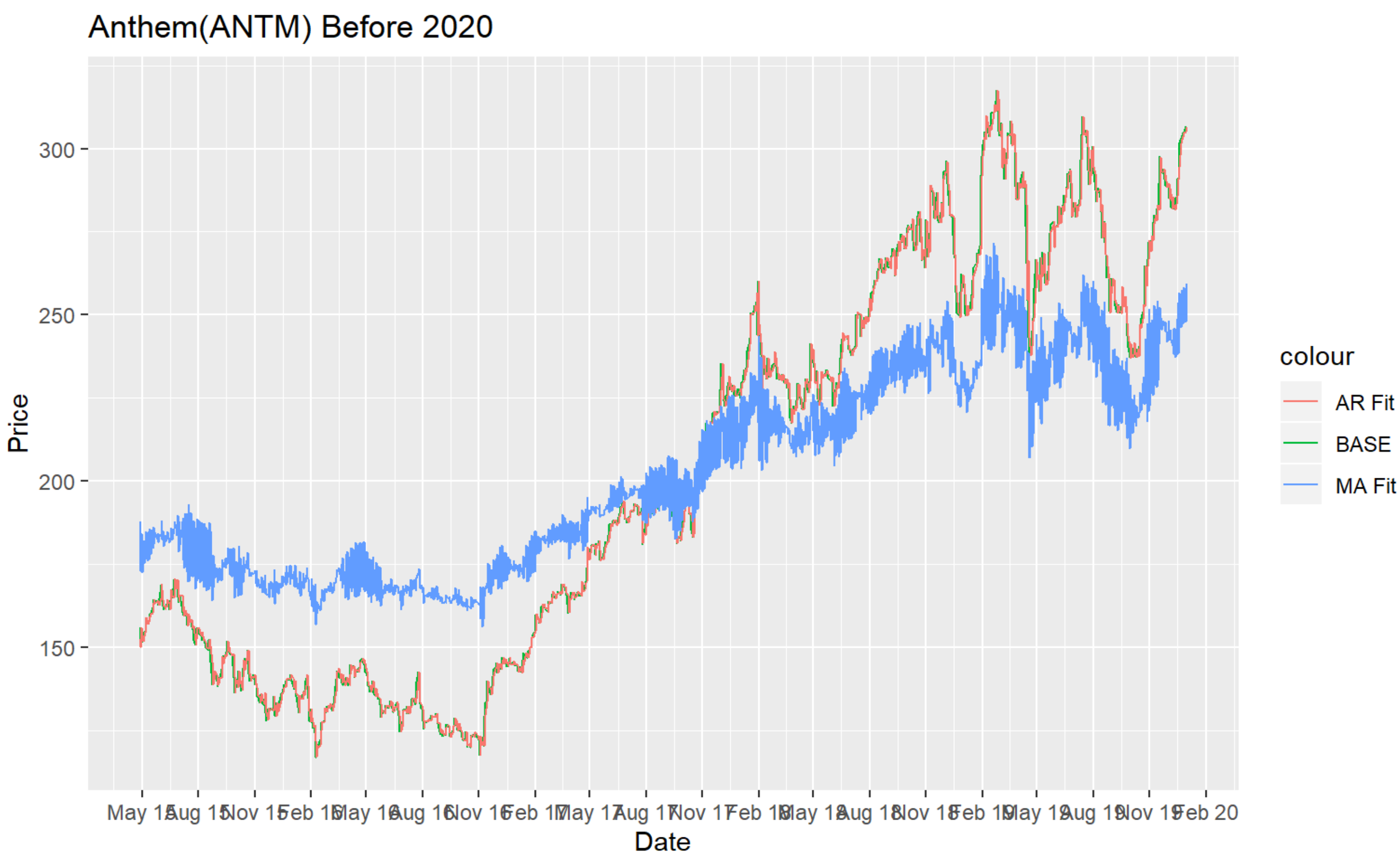


From the above plot we can very clearly see that The red line which indicates the autoregressive model is very close to the baseline which represents the actual data movement, when compared with moving average MA model the blue line is very far from the actual data and hence does not represent a right prediction off the stock price in year 2020.

Fit With Data Before 2020

Now we will use the data Until December 2019 and see how that fits with the data off AR and MA model. We can very clearly see that AR model is doing better and is very close to the base line, whereas MA model is not staying close to the actual data.

```
# Fit with Data before 2020
AR_ANTM_old <- arima(stocks_ANTM_old, order  = c(1,0,0))
MA_ANTM_old <- arima(stocks_ANTM_old, order  = c(0,0,1))
AR_ANTM_old_fit <- as.ts(stocks_ANTM_old) - resid(AR_ANTM_old)
MA_ANTM_old_fit <- as.ts(stocks_ANTM_old) - resid(MA_ANTM_old)

ggplot(stocks_ANTM_old, aes(x = index(stocks_ANTM_old))) +
  geom_line(aes(y= coredata(stocks_ANTM_old) , color="BASE")) +
  geom_line(aes(y = AR_ANTM_old_fit, color = "AR Fit")) +
   geom_line(aes(y = MA_ANTM_old_fit, color = "MA Fit"))+
  ggtitle("Anthem(ANTM) Before 2020") +
  scale_x_date(date_labels = "%b %y", date_breaks = "3 months")+
  xlab("Date") + ylab("Price")
```



Predicting Time Series data

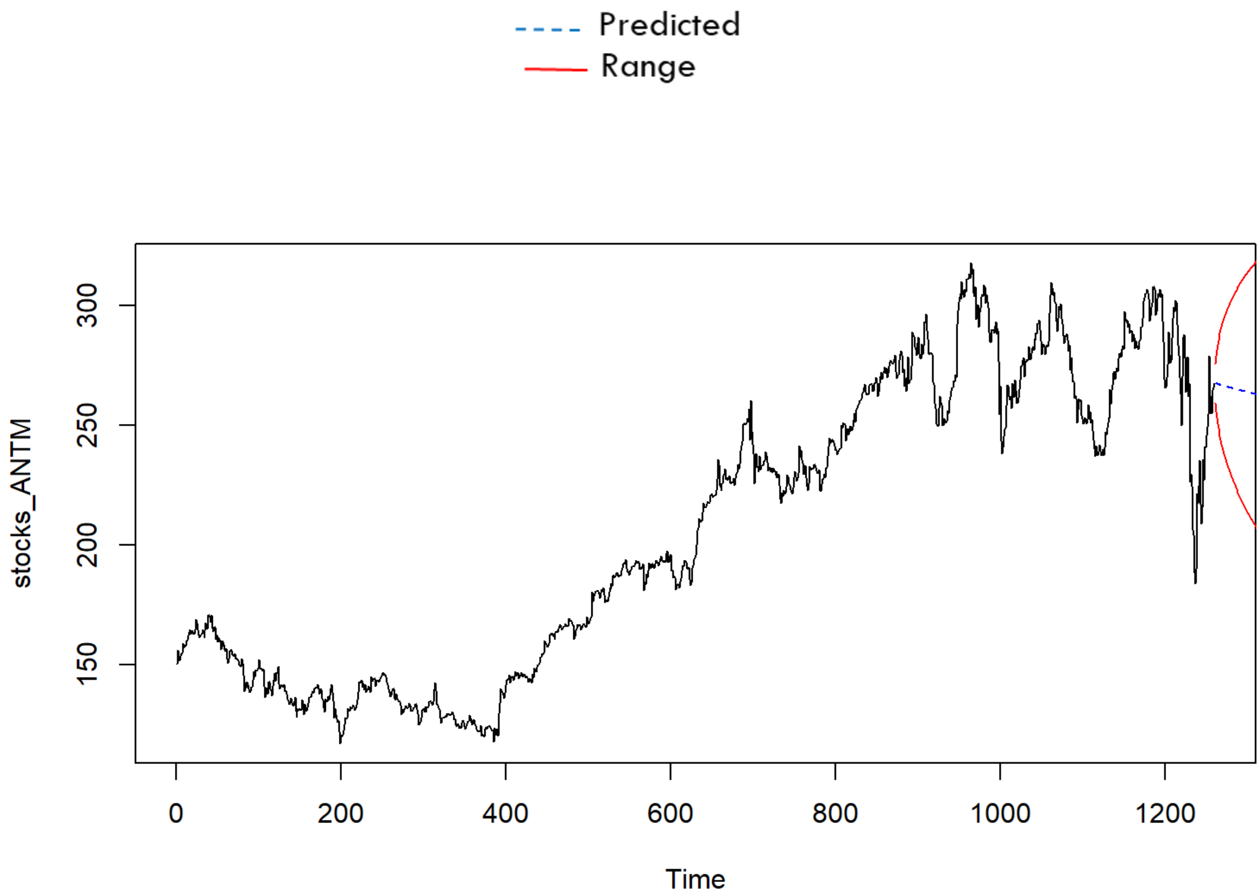
We will evaluate all the data models and see its prediction using both the models with Current Years data.
We can use predict command to predict the future days price by using n.ahead command. Here AR(1) model is the so-called "random walk" model (without drift): it assumes that, from one period to the next, the original time series merely takes a random "step" away from its last recorded position.

```
# Make a 1-step through 10-step forecast based on MA
predict(AR_ANTM,n.ahead = 10)
## $pred
## Time Series:
## Start = 1260
## End = 1269
## Frequency = 1
##  [1] 267.4619 267.3640 267.2663 267.1688 267.0716 266.9745 266.8777 266.7811
##  [9] 266.6846 266.5884
##
## $se
## Time Series:
## Start = 1260
## End = 1269
## Frequency = 1
##  [1]  4.108244  5.803600  7.100186  8.189663  9.146360 10.008444 10.798612
##  [8] 11.531670 12.217916 12.864855
```

Using Full Data

Below graph shows how the price would be if we use AR(1) model to prodict. We are using full data set to predict future price after April 27 2020.

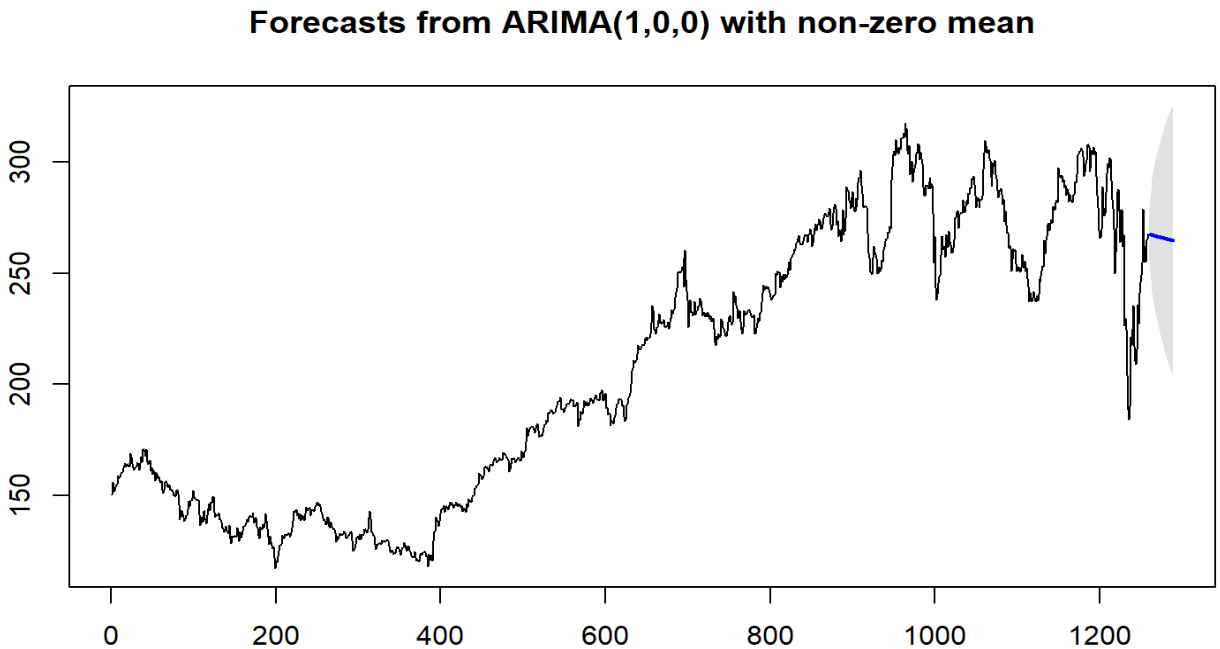
```
# Plot the  series plus the forecast and 95% prediction intervals

AR_forecasts <- predict(AR_ANTM, n.ahead = 300)$pred
AR_forecast_se <- predict(AR_ANTM, n.ahead = 300)$se
plot.ts(stocks_ANTM)
points(AR_forecasts, type = "l", col = 4,lty=2)
# points(AR_forecasts - AR_forecast_se, type = "l", col = 2, lty = 1)
points(AR_forecasts - 2*AR_forecast_se, type = "l", col = 2, lty = 1)
points(AR_forecasts + 2*AR_forecast_se, type = "l", col = 2, lty = 1)
```



We can also use forecast command from forecast package to achieve this.

```
#--------------------
library(forecast)
# We can then use the ARIMA model to make forecasts for future values of the time series, using the "forecast.
AR_ANTM_forcast <- forecast(AR_ANTM, h=30,level=c(99.5))
# We can plot the observed value of stock for the , as well as the predicted that would be predicted for these and for the next 30 days us-
ing our ARIMA(0,0,1) model,
```

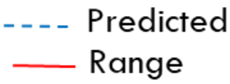
```
plot(AR_ANTM_forcast)
```

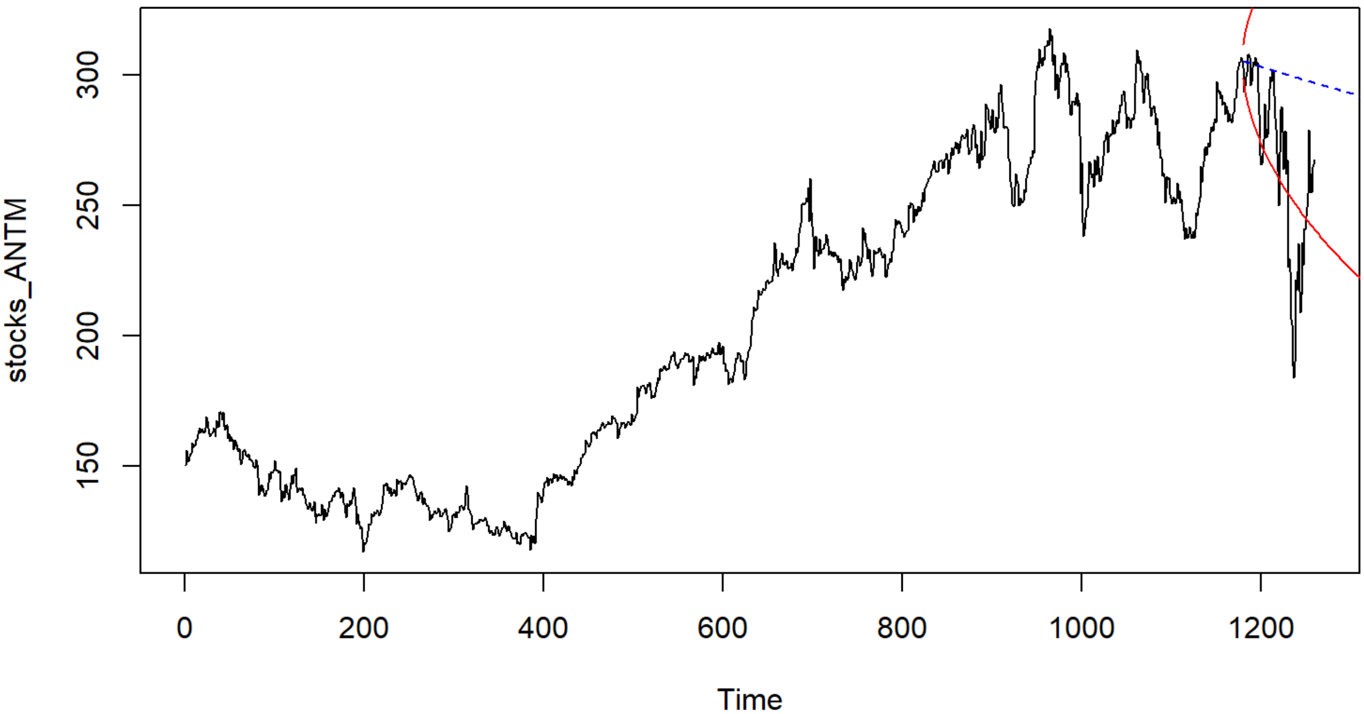


Using Data Fit Till Dec 2019

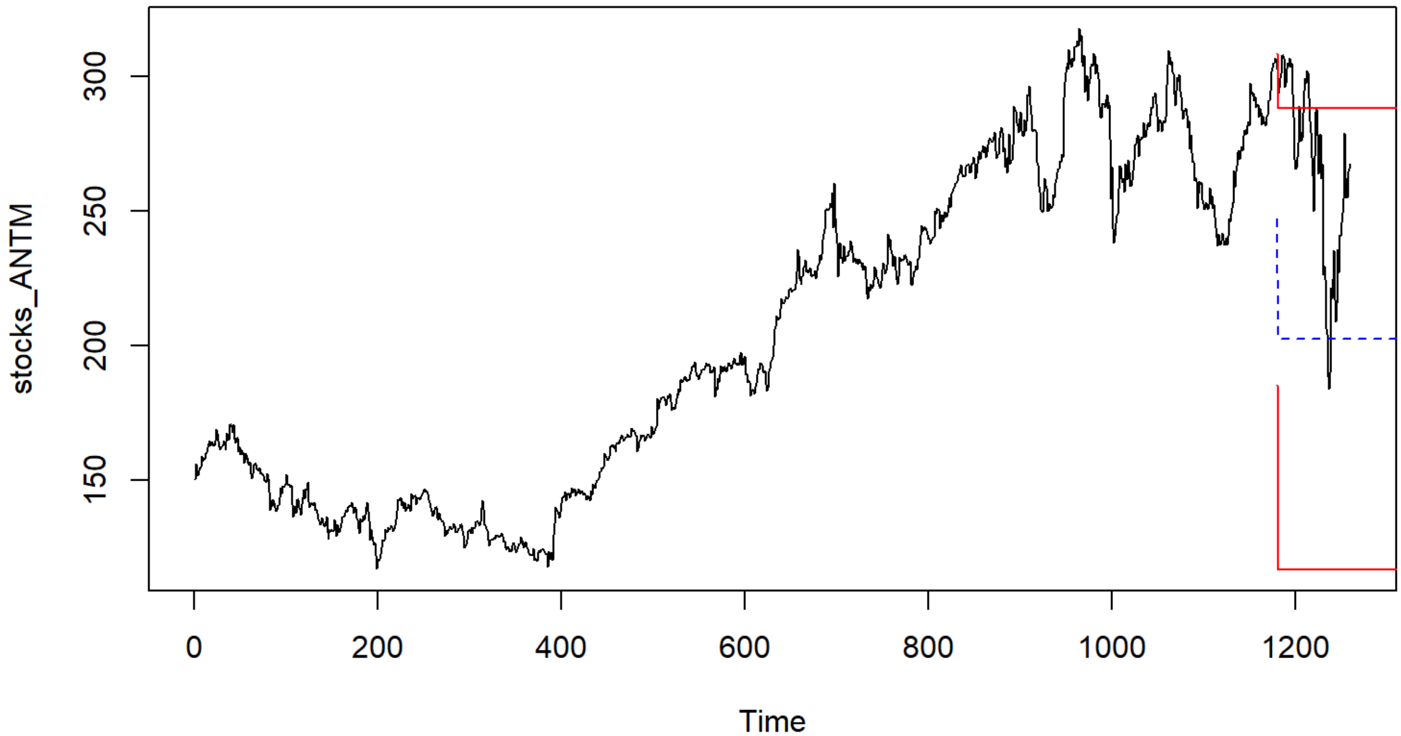
Now let use AR(1) model fit from data of only dec 2019, and try to predict 2020 stock price. Predicting next 300 days stock prices for the Anthem stock , based on AR(1) and MA(1) model.

```
#--------------------
# Plot of original data set and predication  of 2020 based on old data

AR_old_forecasts <- predict(AR_ANTM_old, n.ahead = 300)$pred
AR_old_forecast_se <- predict(AR_ANTM_old, n.ahead = 300)$se
plot.ts(stocks_ANTM)
points(AR_old_forecasts, type = "l", col = 4,lty=2)
points(AR_old_forecasts - 2*AR_old_forecast_se, type = "l", col = 2, lty = 1)
points(AR_old_forecasts + 2*AR_old_forecast_se, type = "l", col = 2, lty = 1)
```






```r
# MA Moving Average Modege

MA_old_forecasts <- predict(MA_ANTM_old, n.ahead = 300)$pred
MA_old_forecast_se <- predict(MA_ANTM_old, n.ahead = 300)$se
plot.ts(stocks_ANTM)
points(MA_old_forecasts, type = "l", col = 4,lty=2)
points(MA_old_forecasts - 2*MA_old_forecast_se, type = "l", col = 2, lty = 1)
points(MA_old_forecasts + 2*MA_old_forecast_se, type = "l", col = 2, lty = 1)
```



As we can very clearly see the from the figure in right:

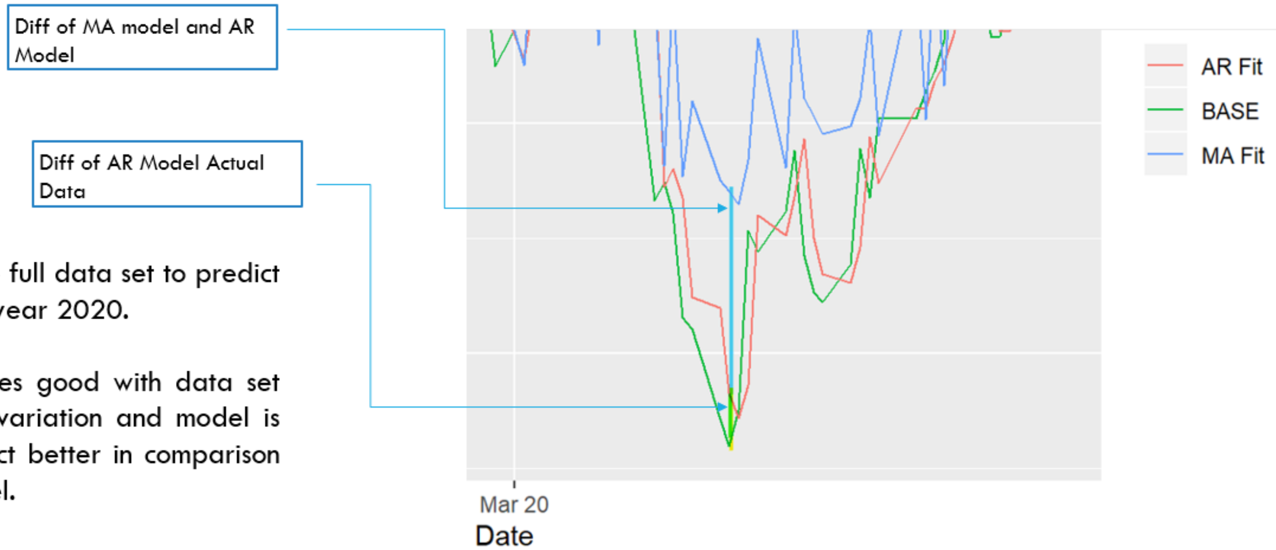
- MA Model has predicted best in case of Year 2020 from the model which was fitted on data until Year Dec-2019.
- AR model prediction is way off from actual value, even though its close to negative range of the prediction.

EXPERIMENTATION AND RESULTS

Check Model

If we compare the data and results from AR and MA Model , we noted that:

- MA model seems to be doing better with predication when we used data till Dec 2019.
- For instance from the figure in right we can see very clearly a sharp drop in price, due to COVID-19 Pandemic. Lets see how these two Model predicted if we go by full data:
- When we use full data set to predict the value for year 2020.
- AR Model does good with data set with random variation and model can predict better in comparison with MA model.



When we use full data set to predict the value for year 2020.

AR Model does good with data set with random variation and model is able to predict better in comparison with MA model.

Compare The Model

Our Model comparison shows that AR(1) model is predicting the price correctly compare with MA model.

Data Group	Model	df	AIC	BIC
Ful Data	AR_ANTM	3	7142.246	7157.66
	MA_ANTM	3	12245.19	12260.6
Till Jan- Apr 2020	AR_ANTM_MY	3	608.5171	615.6632
	MA_ANTM_MY	3	707.9615	715.1076
Til Dec 2019	AR_ANTM_old	3	6151.07	6166.287
	MA_ANTM_old	3	11435.55	11450.77

DISCUSSION AND CONCLUSIONS

Based upon our underacting of this time series analysis we noted that:

- Different model can be used to better predict same set of time series
- AR model is would always perform better for few predictions if market is not stable
- MA model may give better predication when market is very unstable
- Training and testing in Time series data depends on portioning data by date, Random selection of such data may not be accurate choice to better check the efficiency of the model.

REFERENCES

- [Data Camp R cheat-sheet]

- Introduction to Stock Analysis
- R for Data Science cheat-sheet
- A little book of R for Time Series
- Applied Time Series Analysis for Fisheries and Environmental Sciences
- Autoregressive Models
- Moving-average model

APPENDIX

Final_project.html

Github Link

THANK YOU