CHAPTER ONE
Introduction

1.1 Introduction

Time series can be defined as a sequence of data points, usually numeric in nature, sequentially recorded on a regular or irregular time basis (Castillejos, 2006). With rapid development of technologies, time series data are being produced at an outstanding speed and volume in a wide range of applications. Thus, time series data are becoming important in many organizations such as medical, telecommunication banking and finance. A typical example of time series is daily stock market price (P.Wang, H.Wang, & W.Wang, 2011).

Stock market can be defined as the market of exchanges in which the issuing and trading of stocks takes place. This trade is either through over-the-counter (OTC) marketplace or formal exchanges. Stock market or also can be known as equity market, is one of the most important components in a free-market company as it provides companies with access to stocks in exchange for a slice of ownership.

Stock market price meanwhile is defined as the current price at which an asset or stock can be bought or sold. It is the result of traders, investors and dealers interacting and discussing with each other in a stock market. The main goal of stock trading is to own and maintain a portfolio of assets based on buy and sell orders while the long-term objective is to gain as much profit as possible from these trading activities.

This paper focuses on time series analysis on a stock market price dataset. This paper is organized as follows, section 2 highlights an overview of time series and package in R-Studio to used. Section 3 meanwhile provides an insight of our dataset and the methodology of our research on time series. Section 4 highlights the results and findings of our research. Lastly, discussions for future studies and conclusion are provided in section 5.

1.2 Problem statement

Companies and institutions are often focusing on forecasting because everyone would want to predict the future. However, before forecasting, one needs to perform analysis and it can be difficult for the following reasons:

1. The need to convert time-stamped data to time series
2. The need to understand the time series dataset
3. The need to discover appropriate statistical models for time series analysis

1.3 Research objectives

This paper aims to achieve the following objectives:

1.2.1 To describe the important features of the time series pattern
1.2.2 To provide better analysis for future forecasting

CHAPTER TWO
Literature Review

2.1 Background on Time Series

Time series can be categorized into two categories which are stationary series and non-stationary series. Stationary series can be defined as a time series variable which exhibits no significant downward or upward trend over time. For strong definition, stationary means that all moments of all degrees such as mean and variance are all the same (Nason, 2006). Non-stationary meanwhile can be explained as a time series variable which exhibits a significant downward or upward trend over time. Nason (2016) again stated that most statistical books focused on stationary time series. Is stationarity of a time series important? According to Srivastava (2015), one cannot build a time series model unless your time series dataset is stationary. One way to turn it into stationary is to perform differencing.

Differencing is to compute the differences between successive observations. Transformations such as logarithms can stabilize the variance and mean of a time series dataset by removing changes in the time series level.

2.2 R-Studio on time series

R language provides a variety of statistical such as linear and nonlinear modelling, classification, clustering and also time series analysis. Base R contains an infrastructure for representing and analysing time series data. The function "ts" can represent regularly spaced time series (Hyndman, 2018). A few other time series packages of R include:

- Ts: As explained earlier, it is the basic class for regularly spaced time series using numeric time stamps.
- Zoo: A package that is created for regularly and irregularly spaced time series using arbitrary classes for the time stamps.

- Xts: A package that is based on zoo and provides uniform handling of R's different time-based data classes.
- Timeseries: A package that implements time series with "timeDate" time stamps.

CHAPTER THREE
Research Methodology

3.1 Data Sources

Daily close prices of four companies from Feb 2013 to Feb 2018 are extracted from the website of Yahoo Finance. Table 1 shows the description of the "Stock1" dataset. On the other hand, Table 2 shows the variables available in this study.

Dataset	Number of instances	Number of attributes
Stock1	1259	5

Table 1: Description of dataset.

Attributes	Description
Date	Date of the stock price
Stock_AAL	Close price of AAL stock price
Stock_ADS	Close price of ADS stock price
Stock_M0	Close price of M0 stock price
Stock_PWR	Close price of PWR stock price

Table 2: Attributes of Stock Close Price Dataset

3.2 Analysis

Data, the summary and the description will be shown by using R-studio, which is shown by the codes below:

```
>file.choose()
>stock1 <- read.csv("/Users/zahier/Desktop/stock.csv", header = TRUE)
>stock1$date<- as.Date(stock1$date, format= "%Y-%m-%d")
>attach(stock1)
>str(stock1)
>summary(stock1)
>head(stock1)
>tail(stock1)
>anyNA(stock1)
```

3.3 Time series conversion and analysis

This paper will use library (xts) to convert "travel" dataset into time series because of the irregular time series. Figures will also be created using the base `plot()` function.

```
> library(xts)
> time_series_stock <- as.xts(stock1[,-1], order.by= stock1[,1])

> plot(time_series_stock, col= c("darkgoldenrod3", "cyan3", "darkorchid3",
"darkseagreen4"))
> addLegend("right",
+        bty="n",
+        c("ADL", "ADS", "M0", "PWR"),
+        lty=c(1, 1, 1, 1),
+        lwd=c(2, 2, 2, 2),
+        col=c("darkgoldenrod3", "cyan3",
+              "darkorchid3", "darkseagreen4"))

> plot(time_series_stock$stock_AAL, main= "Close Price AAL", col=
"darkgoldenrod3")

> plot(time_series_stock$stock_ADS, main= "Close Price ADS", col= "cyan3")

> plot(time_series_stock$stock_M0, main= "Close Price M0", col=
"darkorchid3")

> plot(time_series_stock$stock_PWR, main= "Close Price PWR", col=
"darkseagreen4")
```

3.4 Differencing on stock market data

Next, we will turn our time_series_stock into stationary time series using base function diff() to compute differences. We will compute the differences on our original data, logged data and lastly on square-root data. We will also produce graphs to display the differencing on original, logged and Square-root data.

```
> diff_stock_AAL<- diff(stock1$stock_AAL)
> diff_stock_AAL_log=diff(log(stock1$stock_AAL))
> diff_stock_AAL_sqrt=diff(sqrt(stock1$stock_AAL))

>
plot(diff_stock_AAL,type='l',xlab='Time',ylab='Difference',main='Differencing
on Original Data')
>
plot(diff_stock_AAL_log,type='l',xlab='Time',ylab='Difference',main='Differen
cing on Logged Data')
>plot(diff_stock_AAL_sqrt,type='l',xlab='Time',ylab='Difference',main='Differ
encing on Square-root Data')
```

Process is repeated for three other companies; ADS, M0 and PWR. Codes can be seen in the appendix.

3.5 Augment Dickey-Fuller Test

Next, we perform Augment Dickey-Fuller test on our differences data using adf.test() of tseries package. For our data to be stationary, the p-value needs to be lower than 0.05. Codes are shown below:

```
> adf.test(diff_stock_AAL)
> adf.test(diff_stock_AAL_log)
> adf.test(diff_stock_AAL_sqrt)
```

Process is also repeated for three other companies; ADS, M0 and PWR. Codes can be seen in the appendix.

3.6 Aggregate

In this section, we will compute the monthly average of stock price. To do that, we will create a month and a year column by using `lubridate` package in R-studio. Then will use `aggregate()` function to find the month wise average for all four companies:

```
> library(lubridate)
> stock1$month1 <- month(stock1$date)
> stock1$year1 <- year(stock1$date)

> monthyearAAL<- aggregate(stock_AAL ~ month1 + year1,
+                                data=stock1,
+                                FUN=mean)
> monthyearMO<- aggregate(stock_M0 ~ month1 + year1,
+                           data=stock1,
+                           FUN=mean)
> monthyearPWR<- aggregate(stock_PWR ~ month1 + year1,
+                          data=stock1,
+                          FUN=mean)
> monthyearXL<- aggregate(stock_XL ~ month1 + year1,
+                           data=stock1,
+                           FUN=mean)
```

Next, we will find the maximum and minimum month wise average price for every year. For company AAL, codes are shown below

```
monthyearAAL[monthyearAAL$stock_AAL==max(monthyearAAL$stock_AAL[monthyearAAL$
year1==2013]),]

monthyearAAL[monthyearAAL$stock_AAL==min(monthyearAAL$stock_AAL[monthyearAAL$
year1==2013]),]

monthyearAAL[monthyearAAL$stock_AAL==max(monthyearAAL$stock_AAL[monthyearAAL$
year1==2014]),]

monthyearAAL[monthyearAAL$stock_AAL==min(monthyearAAL$stock_AAL[monthyearAAL$
year1==2014]),]

monthyearAAL[monthyearAAL$stock_AAL==max(monthyearAAL$stock_AAL[monthyearAAL$
year1==2015]),]

monthyearAAL[monthyearAAL$stock_AAL==min(monthyearAAL$stock_AAL[monthyearAAL$
year1==2015]),]
```

```
monthyearAAL[monthyearAAL$stock_AAL==max(monthyearAAL$stock_AAL[monthyearAAL$
year1==2016]),]

monthyearAAL[monthyearAAL$stock_AAL==min(monthyearAAL$stock_AAL[monthyearAAL$
year1==2016]),]

monthyearAAL[monthyearAAL$stock_AAL==max(monthyearAAL$stock_AAL[monthyearAAL$
year1==2017]),]

monthyearAAL[monthyearAAL$stock_AAL==min(monthyearAAL$stock_AAL[monthyearAAL$
year1==2017]),]

monthyearAAL[monthyearAAL$stock_AAL==max(monthyearAAL$stock_AAL[monthyearAAL$
year1==2018]),]

monthyearAAL[monthyearAAL$stock_AAL==min(monthyearAAL$stock_AAL[monthyearAAL$
year1==2018]),]
```

Codes are repeated for three other companies; ADS, M0 and PWR. Codes can be seen in the appendix.

CHAPTER 4
DATA ANALYSIS AND FINDINGS

4.1 Analysis

First, we changed the Date column to the format of date. To see the class after the change:

```
> class(stock1$date)
[1] "Date"
```

The structural of the data. There is one variable in a date format and five numerical variables (5 companies stock prices):

```
'data.frame':     1259 obs. of  6 variables:
 $ date     : Date, format: "2013-02-08" "2013-02-11" "2013-02-12" ...
 $ stock_AAL: num  14.8 14.5 14.3 14.7 14 ...
 $ stock_ADS: num  154 153 152 152 152 ...
 $ stock_M0 : num  61.4 61.1 61.5 62.2 62 ...
 $ stock_PWR: num  28.4 28.2 28.6 29 29.1 ...
```

The summary of all variables:

```
      date              stock_AAL       stock_ADS       stock_M0       stock_PWR
 Min.   :2013-02-08   Min.   :13.02   Min.   :152.1   Min.   :19.39   Min.   :17.29
 1st Qu.:2014-05-10   1st Qu.:34.33   1st Qu.:214.4   1st Qu.:26.82   1st Qu.:26.35
 Median :2015-08-10   Median :40.87   Median :242.6   Median :41.19   Median :29.36
 Mean   :2015-08-09   Mean   :38.39   Mean   :241.0   Mean   :38.01   Mean   :29.91
 3rd Qu.:2016-11-05   3rd Qu.:46.53   3rd Qu.:272.0   3rd Qu.:47.10   3rd Qu.:34.73
 Max.   :2018-02-07   Max.   :58.47   Max.   :309.9   Max.   :64.30   Max.   :39.86
```

Figure 1: Summary of the Stock close price dataset

From the summary above, it shows that Stock ADS has the highest close price with an average of 241 while the lowest close price is Stock PWR with an average of 29.91. It also shows that the data is from 08/02/2013 to 07/02/2018. Figure 2 below shows the first six rows of the dataset while Figure 3 shows the last six rows of the dataset.

```
         date stock_AAL stock_ADS stock_M0 stock_PWR
1 2013-02-08     14.75    154.08    61.40     28.42
2 2013-02-11     14.46    153.42    61.14     28.19
3 2013-02-12     14.27    152.32    61.49     28.61
4 2013-02-13     14.66    152.32    62.20     28.99
5 2013-02-14     13.99    152.37    61.98     29.11
6 2013-02-15     14.50    155.41    62.18     28.74
```

Figure 2: Sample of Stock Close price dataset (First five rows)

```
         date stock_AAL stock_ADS stock_M0 stock_PWR
1254 2018-01-31     54.32    256.66    27.30     38.49
1255 2018-02-01     53.88    257.11    26.87     38.26
1256 2018-02-02     52.10    249.91    26.04     36.09
1257 2018-02-05     49.76    238.78    24.78     34.40
1258 2018-02-06     51.18    241.47    25.07     35.18
1259 2018-02-07     51.40    246.75    25.25     34.45
```

Figure 3: Sample of Stock Close price dataset (Last five rows)

Next, it was confirmed by R-studio that no missing values were detected.

```
> anyNA(stock1)
[1] FALSE
```

4.2 Time series conversion and analysis

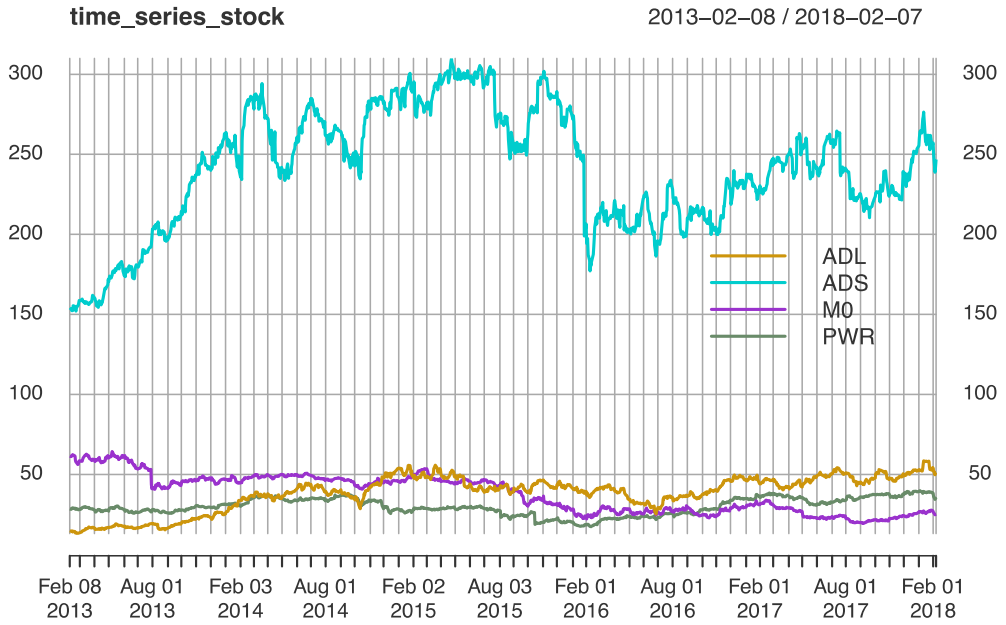

Figure 4: Close price of all companies

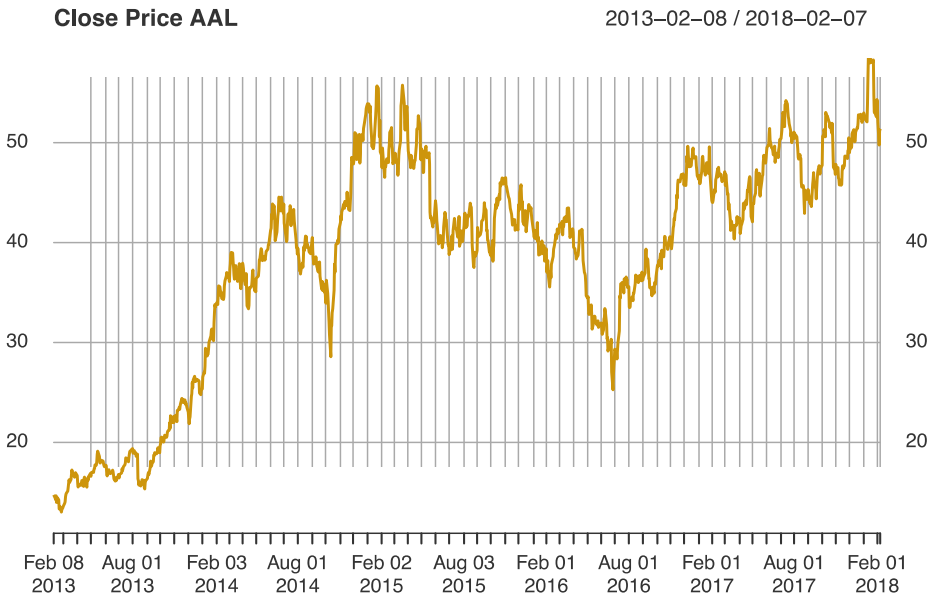

Figure 5: Close price of AAL

The figure displays the close price of AAL increases in general over the past five years. However, there is no seasonal pattern in the fluctuation of stock price. In other words, there is no rise and fall in the dataset that repeats regularly over the same time period. Also, the variance of the stock price AAL seems to be unstable. Thus, we may apply logarithm or square root transformation on our Stock1 dataset to stabilize the variance.

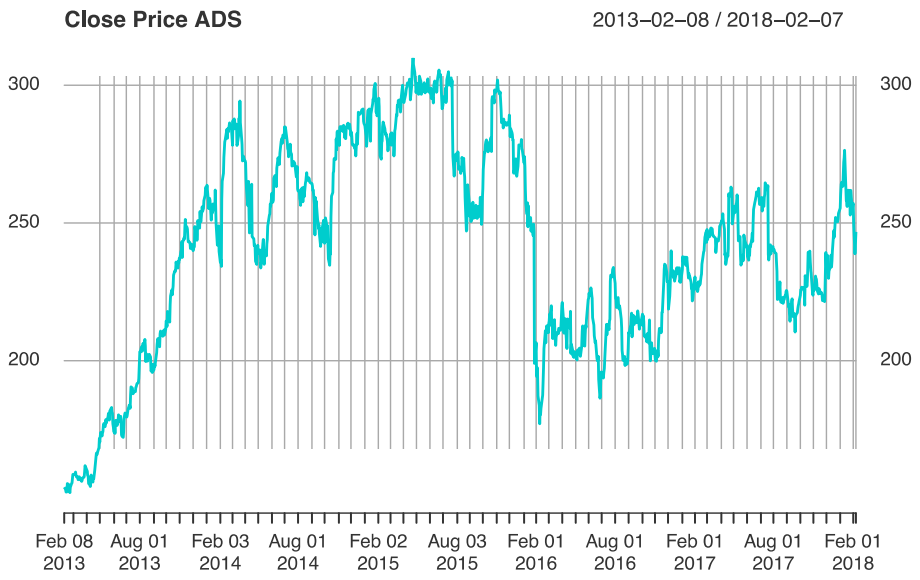

Figure 6: Close price of ADS

Figure 6 displays the close price of ADS which shows a sudden shift between 3rd Aug 2015 and 1st Feb 2016. Similar with Figure 5, there is no obvious pattern in the fluctuation of stock price and unstable variance. Thus, we may use logarithm or square root transformation on original data to stabilize the variance.

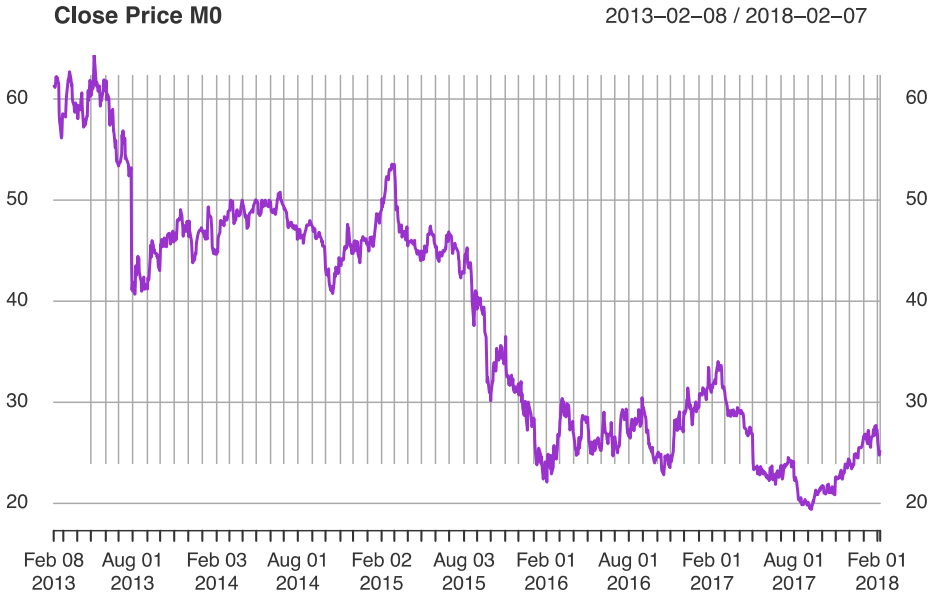

Figure 7: Close price of M0

Contrary to Figure 5, the close price of M0 in Figure 7 decreases in general during the period of Feb 2013 until Feb 2018. Just like previous figures of AAL and ADS, there is no seasonal pattern in the fluctuation of stock price and also unstable variance. Therefore, the application of logarithm or square root transformation on Stock1 dataset might be appropriate to stabilize the variance.

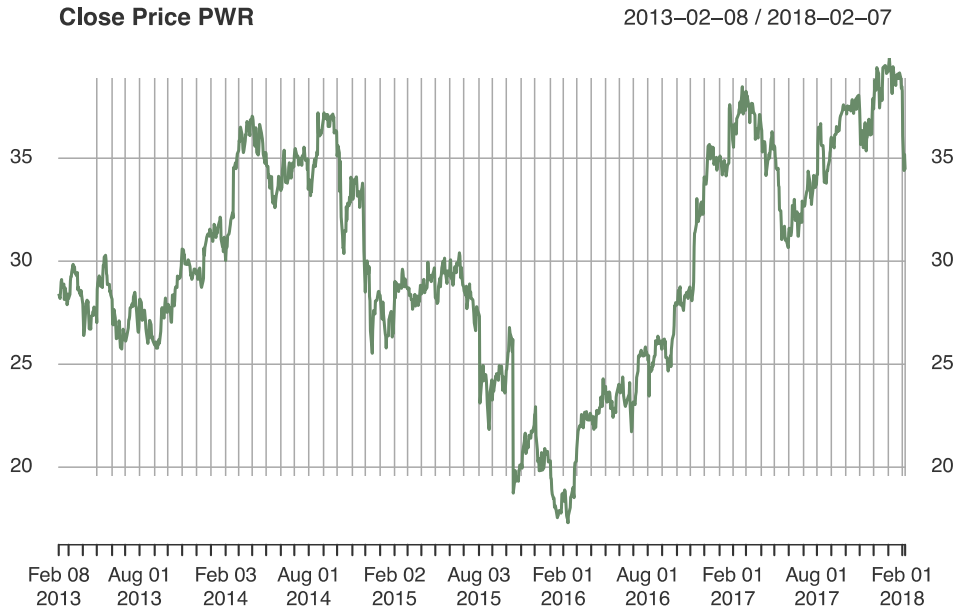

Figure 8: Close price of PWR

Figure 7 displays the close price of PWR during the period of Feb 2013 until Feb 2018. It shows very low prices between 3rd August 2015 and 1st August 2016. Similar with the other figures, there is no seasonal pattern in the fluctuation of stock price and yes, unstable variance. Based on four figures displayed, these observations implied that logarithm or square root transformation on original data would be appropriate.

4.3 Differencing time series data

In this section, we computed the differences to stabilize the variance and results are shown in Figures 9-12. It is clear that the differences in those three figures show reasonably stable variance over time. Comparing all the three plots, it seems that they did a good job in stabilizing the variance and also in tuning down some the extreme values.

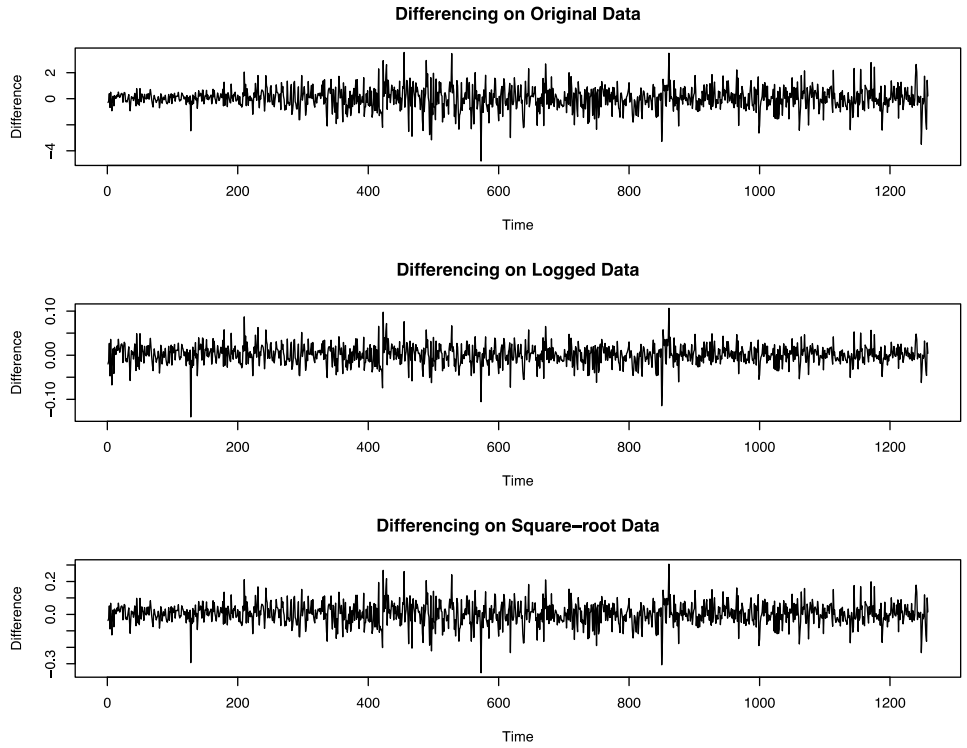

Figure 9: Differencing on original, logged and square-root data of AAL.

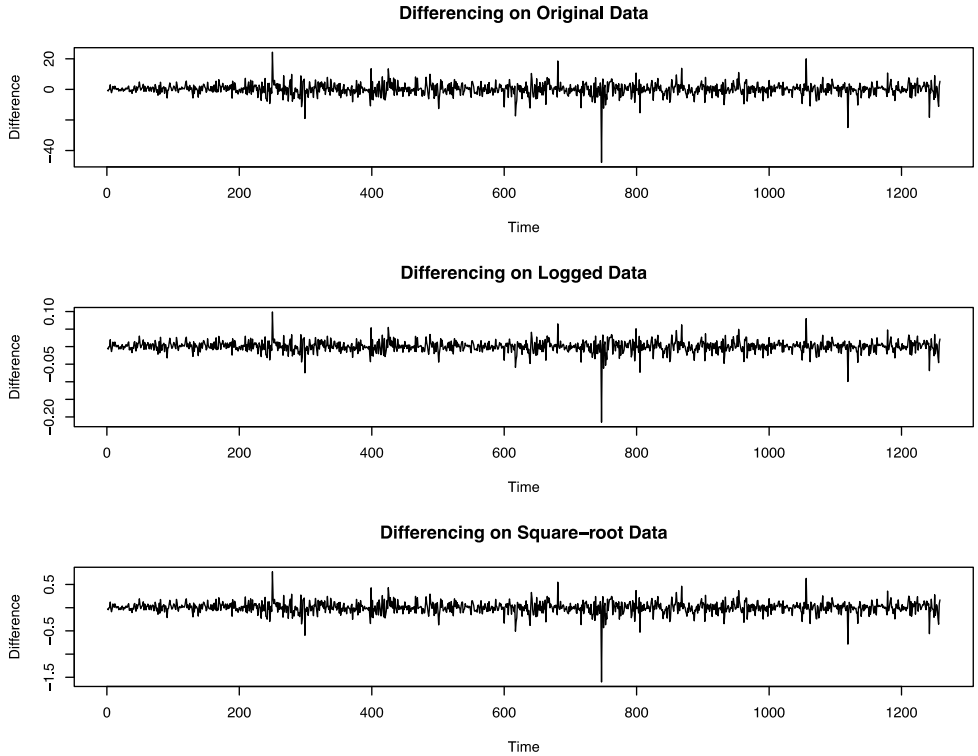

Figure 10: Differencing on original, logged and square-root data of ADS

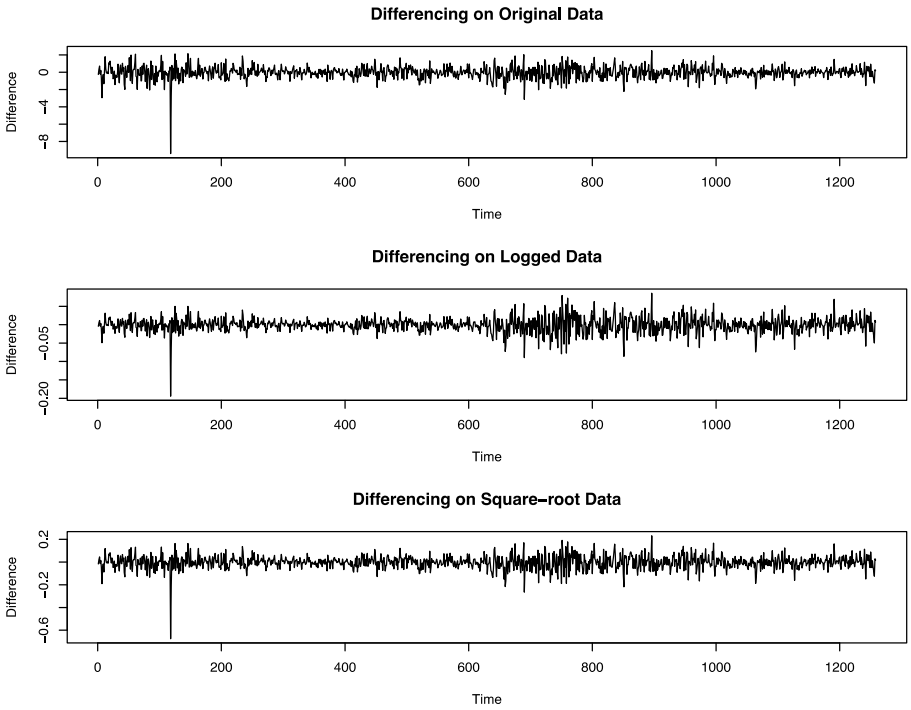

Figure 11: Differencing on original, logged and square-root data of M0

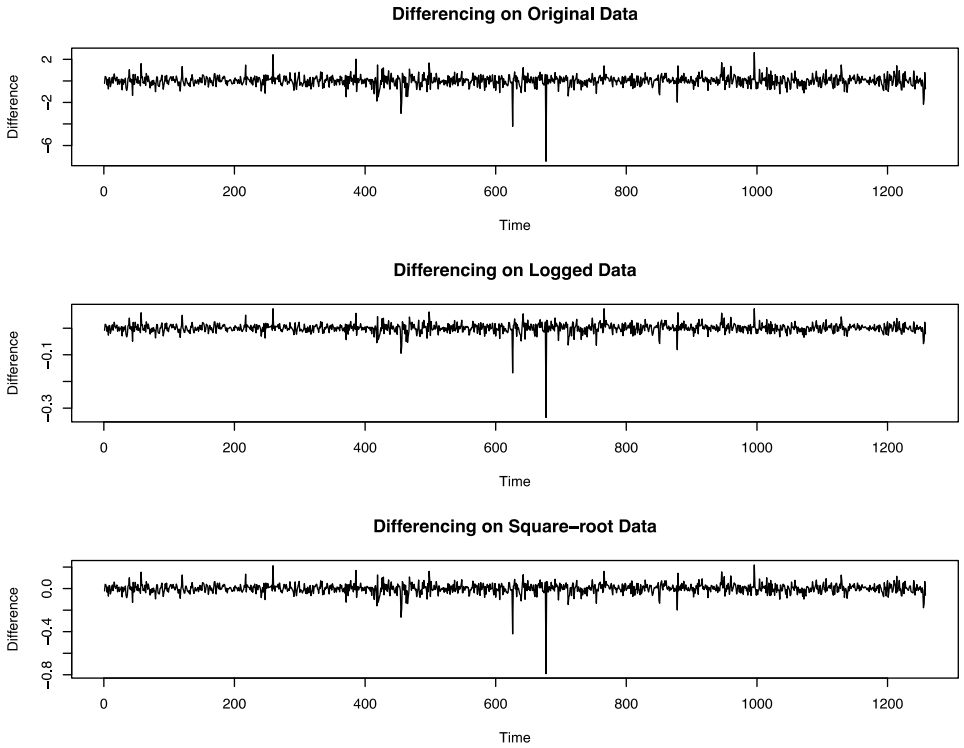

Figure 12: Differencing on original, logged and square-root data of PWR

4.3 Augmented Dickey-Fuller Test

To test whether it is a stationary time series or not, we did Augmented Dickey-Fuller test. Results on every differences data are producing the same p-value of 0.01, lower than 0.05 which means every differences data is now a stationary time series.

```
        Augmented Dickey-Fuller Test

data:  diff_stock_AAL
Dickey-Fuller = -11.389, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Figure 13: Dickey-Fuller Test on differences of AAL stock price

4.4 Aggregate

In this section, we computed the monthwise average for all four companies; AAL, ADS, M0 and PWR. Table 1 shows the month wise average while Table 2 shows the maximum and minimum average for all companies. Regarding Table 2, the average price for company AAL and company PWR generally increase over time; 24.97 to 51.66 for AAL, 29.99 to 35.68 for PWR. The average close stock price for company Stock ADS meanwhile has been consistent. On the other hand, the average price for company M0 generally decreases over time, starting from an average of 61.30 in 2013 to 25.60 in 2018.

Year	Stock AAL	Stock ADS	Stock M0	Stock PWR
2013				
Jan	-	-	-	-
Feb	13.877	154.372	59.623	28.480
March	15.777	158.012	60.196	29.047
Apr	16.109	160.616	59.172	27.393
May	17.811	177.283	61.301	29.091
June	16.839	177.020	57.889	26.691
July	17.929	187.741	53.140	27.312
Aug	17.034	202.260	42.178	26.998
Sep	18.184	206.498	44.117	26.823
Oct	21.060	225.737	45.959	28.471
Nov	23.482	243.603	47.405	29.813
Dec	24.965	249.750	45.915	29.992
2014				
Jan	29.716	253.841	46.683	31.343
Feb	35.347	271.915	47.295	32.585
March	37.180	282.001	48.894	36.122
Apr	35.862	251.989	48.862	35.880

May	38.392	242.761	49.403	33.702
June	42.764	270.565	49.544	34.265
July	41.841	274.228	47.719	34.956
Aug	38.820	262.854	46.995	35.029
Sep	37.254	252.054	46.321	36.821
Oct	35.548	257.582	42.445	32.897
Nov	43.857	283.643	45.235	33.346
Dec	50.273	283.472	45.446	28.143
2015				
Jan	52.296	289.057	46.853	27.006
Feb	48.895	282.032	51.675	28.871
March	51.002	287.330	48.121	28.265
Apr	49.323	300.770	45.266	29.065
May	46.404	299.351	45.797	29.068
June	41.302	298.993	45.077	29.549
July	40.671	291.304	44.640	28.006
Aug	41.284	266.172	42.296	24.055
Sep	41.241	254.080	36.695	24.218
Oct	43.315	282.634	33.867	22.680
Nov	43.480	291.302	32.312	21.129
Dec	43.221	276.964	29.633	20.721
2016				
Jan	39.957	249.424	24.673	18.368
Feb	38.721	196.573	24.525	18.577
March	41.883	212.525	28.774	22.171
Apr	38.883	209.982	26.860	22.778
May	32.529	206.770	26.057	23.236
June	30.149	208.525	26.676	23.331
July	33.888	215.246	27.492	24.985
Aug	35.544	210.965	28.070	25.450
Sep	36.456	214.040	26.170	25.970
Oct	39.044	208.187	24.197	28.226
Nov	43.777	215.411	26.883	31.146
Dec	47.823	230.371	29.416	34.839
2017				
Jan	47.139	231.431	31.215	35.326
Feb	46.007	233.056	32.483	37.074
March	42.964	245.250	29.243	37.092
Apr	43.716	249.970	27.883	35.516
May	45.962	246.663	23.485	32.747
June	49.410	250.475	22.725	31.967
July	52.027	251.964	23.785	33.458
Aug	47.343	227.858	20.730	35.109
Sep	45.895	218.352	20.578	36.614
Oct	50.756	229.120	21.364	37.535
Nov	47.587	226.568	23.106	36.395
Dec	51.151	241.483	24.617	38.672
2018				
Jan	54.903	261.725	26.713	38.962

Feb	51.664	246.804	25.602	35.676

Table 1: Monthwise Average for all companies

Year	Stock AAL	Stock ADS	Stock M0	Stock PWR
2013				
Max	24.97	249.75	61.30	29.99
Min	13.88	154.37	42.178	26.69
2014				
Max	50.27	283.64	49.54	36.82
Min	29.72	242.76	42.45	28.14
2015				
Max	52.30	300.77	51.68	29.55
Min	40.67	254.08	29.63	20.72
2016				
Max	47.82	249.42	29.42	34.84
Min	30.15	196.57	24.20	18.37
2017				
Max	52.03	251.96	32.48	38.67
Min	42.96	218.35	20.58	31.97
2018				
Max	54.90	261.72	26.71	38.96
Min	51.66	246.80	25.60	35.68

Table 2: Maximum and minimum average for all companies

CHAPTER FIVE
Conclusions

5.1 Conclusion

As a conclusion for this study, we have identified the important features of our stock1 dataset. Plots of close prices for every company were compared with each having their own unique features but one thing they have in common, which is the instability of their variance. To stabilize the variance, we perform first degree differencing on original, logged and square root data.

We also calculate the month wise average of every company's close price for 2013 until 2018 using the base `aggregate()` function in R. Instead of looking at the trend like we did earlier, we were looking at the average price for every month, too see how well every company performed.

Statistics showed that things are not looking good for Company M0, while company ADS has been consistent. Lastly, average close price for companies AAL and PWR are generally increasing over time.

5.2 Recommendation

For future research, the first-degree differencing can be used for forecasting. A stationarized series makes it easy to predict as statistical properties such as mean and variance are all constant over time (Duke, n.d.). Besides, possible improvements could be done such as more companies to be included and also to consider some analysis on High or Close price. Proper technique of analysis and forecasting is required to be made by future researchers.

References

Castillejos, A. M. (2006). Management of Time Series Data. Canberra, Australia: University of Canberra.

Duke (n.d.). Available at: https://people.duke.edu/~rnau/411diff.htm

Hyndman, R., J. (2018). Available at: https://cran.r-project.org/web/views/TimeSeries.html

Minitab (n.d.). Available at: https://support.minitab.com/en-us/minitab-express/1/help-and-how-to/graphs/time-series-plot/interpret-the-results/key-results/

Nason, G. P. 2006. Stationary and non-stationary times series. In: MADER, H. M., COLES, S. G., CONNOR, C. B. & CONNOR, L. J. (eds) Statistics in Volcanology. Special Publications of IAVCEI, 1. Geological Society, London, 129 – 142

P.Wang, H.Wang, & W.Wang. (2011). Finding Semantics in Time Series. Microsoft Research Asia.

Appendix

Differencing ADS company

```
> diff_stock_ADS<- diff(stock1$stock_ADS)
> diff_stock_ADS_log=diff(log(stock1$stock_ADS))
> diff_stock_ADS_sqrt=diff(sqrt(stock1$stock_ADS))
>
plot(diff_stock_ADS,type='l',xlab='Time',ylab='Difference',main='Differencing
on Original Data')
```

```
>
plot(diff_stock_ADS_log,type='l',xlab='Time',ylab='Difference',main='Differen
cing on Logged Data')
>
plot(diff_stock_ADS_sqrt,type='l',xlab='Time',ylab='Difference',main='Differe
ncing on Square-root Data')
```

Differencing M0 company

```
> diff_stock_M0<- diff(stock1$stock_M0)
> diff_stock_M0_log=diff(log(stock1$stock_M0))
> diff_stock_M0_sqrt=diff(sqrt(stock1$stock_M0))
>
plot(diff_stock_M0,type='l',xlab='Time',ylab='Difference',main='Differencing
on Original Data')
>
plot(diff_stock_M0_log,type='l',xlab='Time',ylab='Difference',main='Differenc
ing on Logged Data')
>
plot(diff_stock_M0_sqrt,type='l',xlab='Time',ylab='Difference',main='Differen
cing on Square-root Data')
```

Differencing PWR company

```
> diff_stock_PWR<- diff(stock1$stock_PWR)
> diff_stock_PWR_log=diff(log(stock1$stock_PWR))
> diff_stock_PWR_sqrt=diff(sqrt(stock1$stock_PWR))
>
plot(diff_stock_PWR,type='l',xlab='Time',ylab='Difference',main='Differencing
on Original Data')
>
plot(diff_stock_PWR_log,type='l',xlab='Time',ylab='Difference',main='Differen
cing on Logged Data')
>
plot(diff_stock_PWR_sqrt,type='l',xlab='Time',ylab='Difference',main='Differe
ncing on Square-root Data')
```

Augmented Dickey-Fuller Test

Ads company

```
> adf.test(diff_stock_ADS)

      Augmented Dickey-Fuller Test

data:  diff_stock_ADS
Dickey-Fuller = -10.105, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

> adf.test(diff_stock_ADS_log)

      Augmented Dickey-Fuller Test

data:  diff_stock_ADS_log
Dickey-Fuller = -10.121, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

> adf.test(diff_stock_ADS_sqrt)
```

```
      Augmented Dickey-Fuller Test

data:  diff_stock_ADS_sqrt
Dickey-Fuller = -10.107, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Mo company

```
> adf.test(diff_stock_MO)

      Augmented Dickey-Fuller Test

data:  diff_stock_MO
Dickey-Fuller = -11.841, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

> adf.test(diff_stock_MO_log)

      Augmented Dickey-Fuller Test

data:  diff_stock_MO_log
Dickey-Fuller = -11.395, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
> adf.test(diff_stock_MO_sqrt)

      Augmented Dickey-Fuller Test

data:  diff_stock_MO_sqrt
Dickey-Fuller = -11.627, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

PWR company

```
> adf.test(diff_stock_PWR)

      Augmented Dickey-Fuller Test

data:  diff_stock_PWR
Dickey-Fuller = -10.102, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary

> adf.test(diff_stock_PWR_log)

      Augmented Dickey-Fuller Test

data:  diff_stock_PWR_log
Dickey-Fuller = -10.569, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary


> adf.test(diff_stock_PWR_sqrt)

      Augmented Dickey-Fuller Test

data:  diff_stock_PWR_sqrt
Dickey-Fuller = -10.339, Lag order = 10, p-value = 0.01
alternative hypothesis: stationary
```

Aggregate

ADS company

```
>monthyearADS[monthyearADS$stock_ADS==max(monthyearADS$stock_ADS[monthyearADS
$year1==2013]),]
   month1 year1 stock_ADS
11     12  2013    249.75
>monthyearADS[monthyearADS$stock_ADS==min(monthyearADS$stock_ADS[monthyearADS
$year1==2013]),]
  month1 year1 stock_ADS
1      2  2013  154.3721
>monthyearADS[monthyearADS$stock_ADS==max(monthyearADS$stock_ADS[monthyearADS
$year1==2014]),]
   month1 year1 stock_ADS
22     11  2014  283.6426
>monthyearADS[monthyearADS$stock_ADS==min(monthyearADS$stock_ADS[monthyearADS
$year1==2014]),]
   month1 year1 stock_ADS
16      5  2014   242.761
>monthyearADS[monthyearADS$stock_ADS==max(monthyearADS$stock_ADS[monthyearADS
$year1==2015]),]
   month1 year1 stock_ADS
27      4  2015  300.7695
>monthyearADS[monthyearADS$stock_ADS==min(monthyearADS$stock_ADS[monthyearADS
$year1==2015]),]
   month1 year1 stock_ADS
32      9  2015  254.0805
>monthyearADS[monthyearADS$stock_ADS==max(monthyearADS$stock_ADS[monthyearADS
$year1==2016]),]
   month1 year1 stock_ADS
36      1  2016  249.4237
>monthyearADS[monthyearADS$stock_ADS==min(monthyearADS$stock_ADS[monthyearADS
$year1==2016]),]
   month1 year1 stock_ADS
37      2  2016   196.573
>monthyearADS[monthyearADS$stock_ADS==max(monthyearADS$stock_ADS[monthyearADS
$year1==2017]),]
   month1 year1 stock_ADS
54      7  2017   251.964
>monthyearADS[monthyearADS$stock_ADS==min(monthyearADS$stock_ADS[monthyearADS
$year1==2017]),]
   month1 year1 stock_ADS
56      9  2017   218.352
>monthyearADS[monthyearADS$stock_ADS==max(monthyearADS$stock_ADS[monthyearADS
$year1==2018]),]
   month1 year1 stock_ADS
60      1  2018  261.7248>
monthyearADS[monthyearADS$stock_ADS==min(monthyearADS$stock_ADS[monthyearADS$
year1==2018]),]
   month1 year1 stock_ADS
61      2  2018   246.804
```

M0 company

```
>monthyearM0[monthyearM0$stock_M0==max(monthyearM0$stock_M0[monthyearM0$year1
==2013]),]
  month1 year1 stock_M0
4      5  2013 61.30136
>monthyearM0[monthyearM0$stock_M0==min(monthyearM0$stock_M0[monthyearM0$year1
==2013]),]
```

```
  month1 year1 stock_MO
7      8  2013 42.17773
>monthyearMO[monthyearMO$stock_MO==max(monthyearMO$stock_MO[monthyearMO$year1
==2014]),]
   month1 year1 stock_MO
17      6  2014 49.54381
>monthyearMO[monthyearMO$stock_MO==min(monthyearMO$stock_MO[monthyearMO$year1
==2014]),]
   month1 year1 stock_MO
21     10  2014 42.44522
>monthyearMO[monthyearMO$stock_MO==max(monthyearMO$stock_MO[monthyearMO$year1
==2015]),]
   month1 year1 stock_MO
25      2  2015 51.67526
>monthyearMO[monthyearMO$stock_MO==min(monthyearMO$stock_MO[monthyearMO$year1
==2015]),]
   month1 year1 stock_MO
35     12  2015 29.63273
>monthyearMO[monthyearMO$stock_MO==max(monthyearMO$stock_MO[monthyearMO$year1
==2016]),]
   month1 year1 stock_MO
47     12  2016 29.41571
>monthyearMO[monthyearMO$stock_MO==min(monthyearMO$stock_MO[monthyearMO$year1
==2016]),]
   month1 year1 stock_MO
45     10  2016 24.19714
>monthyearMO[monthyearMO$stock_MO==max(monthyearMO$stock_MO[monthyearMO$year1
==2017]),]
   month1 year1 stock_MO
49      2  2017 32.48316
>monthyearMO[monthyearMO$stock_MO==min(monthyearMO$stock_MO[monthyearMO$year1
==2017]),]
   month1 year1 stock_MO
56      9  2017  20.5775
>monthyearMO[monthyearMO$stock_MO==max(monthyearMO$stock_MO[monthyearMO$year1
==2018]),]
   month1 year1 stock_MO
60      1  2018 26.71286
>monthyearMO[monthyearMO$stock_MO==min(monthyearMO$stock_MO[monthyearMO$year1
==2018]),]
   month1 year1 stock_MO
61      2  2018   25.602
```

M0 company

```
>monthyearPWR[monthyearPWR$stock_PWR==max(monthyearPWR$stock_PWR[monthyearPWR
$year1==2013]),]
   month1 year1 stock_PWR
11     12  2013   29.9919
>monthyearPWR[monthyearPWR$stock_PWR==min(monthyearPWR$stock_PWR[monthyearPWR
$year1==2013]),]
  month1 year1 stock_PWR
5      6  2013    26.691
>monthyearPWR[monthyearPWR$stock_PWR==max(monthyearPWR$stock_PWR[monthyearPWR
$year1==2014]),]
   month1 year1 stock_PWR
20      9  2014  36.82095
>monthyearPWR[monthyearPWR$stock_PWR==min(monthyearPWR$stock_PWR[monthyearPWR
$year1==2014]),]
   month1 year1 stock_PWR
23     12  2014  28.14273
```

```
>monthyearPWR[monthyearPWR$stock_PWR==max(monthyearPWR$stock_PWR[monthyearPWR
$year1==2015]),]
   month1 year1 stock_PWR
29      6  2015  29.54864
>monthyearPWR[monthyearPWR$stock_PWR==min(monthyearPWR$stock_PWR[monthyearPWR
$year1==2015]),]
   month1 year1 stock_PWR
35     12  2015  20.72136
>monthyearPWR[monthyearPWR$stock_PWR==max(monthyearPWR$stock_PWR[monthyearPWR
$year1==2016]),]
   month1 year1 stock_PWR
47     12  2016  34.83857
>monthyearPWR[monthyearPWR$stock_PWR==min(monthyearPWR$stock_PWR[monthyearPWR
$year1==2016]),]
   month1 year1 stock_PWR
36      1  2016  18.36789
>monthyearPWR[monthyearPWR$stock_PWR==max(monthyearPWR$stock_PWR[monthyearPWR
$year1==2017]),]
   month1 year1 stock_PWR
59     12  2017    38.672
>monthyearPWR[monthyearPWR$stock_PWR==min(monthyearPWR$stock_PWR[monthyearPWR
$year1==2017]),]
   month1 year1 stock_PWR
53      6  2017  31.96727
>monthyearPWR[monthyearPWR$stock_PWR==max(monthyearPWR$stock_PWR[monthyearPWR
$year1==2018]),]
   month1 year1 stock_PWR
60      1  2018  38.96238
>monthyearPWR[monthyearPWR$stock_PWR==min(monthyearPWR$stock_PWR[monthyearPWR
$year1==2018]),]
   month1 year1 stock_PWR
61      2  2018    35.676
```