Financial series analysis with R
Rose Bandolo

Introduction

This assessment is based on Lab 1 and is directed at our capability in using R and our ability to follow guidance about R. So in this report we present two main parts. In the first part we will work whith the Ghana Stock Exchange Composite Index ***GSEC5July13-4July18.xlsx*** from 05July2013 – 08July2018, 1232 trading days in all. We will try to obtain the different graphics and interpret one ggplot code for one figure.

In the second part, we follow the same instructions by using another financial series. We decide to use S&P100 Index **S&P500dataset1972-1999.xlsx** from January 1972 to December 1982.

I- Ghana Stock Exchange Composite

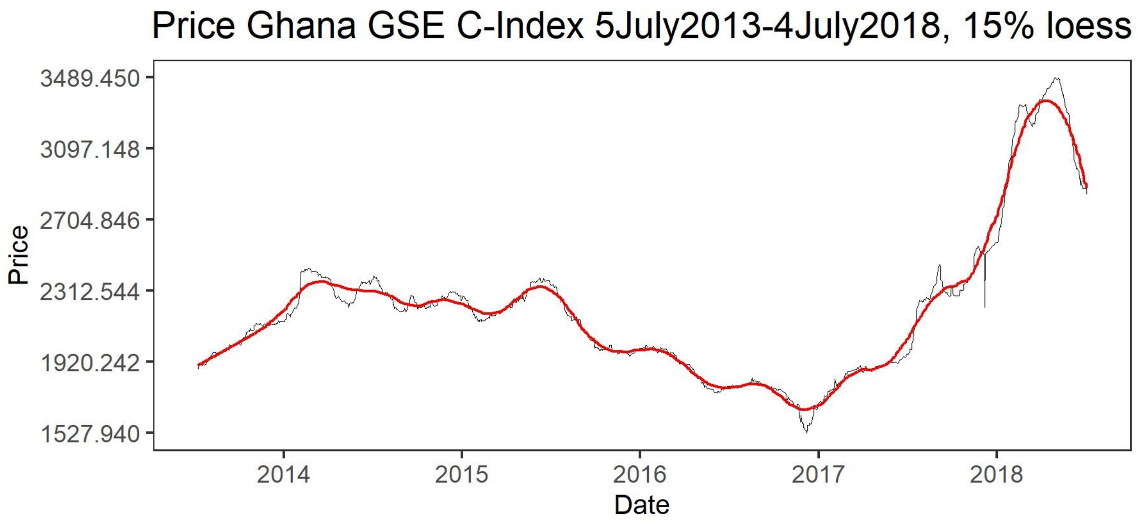
1- output from running *GSEC_FTplotSmoothRfunction.R*

```
Read dateindexpricereturndata file into index-ordered dataframe  df.FT xlsx
file using openxls package

df.FT data frame from original xlsx data file - not usually with date as
date class - review file
'data.frame':	1232 obs. of  2 variables:
 $ Date : chr  "2013-07-05" "2013-07-08" "2013-07-09" "2013-07-10" ...
 $ Price: num  1885 1884 1881 1888 1898 ...

3. For Dates of the form (year-month-day) in xlsx file, conversion code
available here

All dates have been created to be of standard date class form

df.FT data frame   -   Dates of date class
'data.frame':	1232 obs. of  2 variables:
 $ Date : Date, format: "2013-07-05" "2013-07-08" ...
 $ Price: num  1885 1884 1881 1888 1898 ...

[1] 1232
Calulate %returns and name serReturn

 num [1:1232] NA 1885 1884 1881 1888 ...
 num [1:1232] NA -0.0615 -0.1497 0.3881 0.4983 ...
 num [1:1232] NA -0.000615 -0.001497 0.003881 0.004983 ...
Data frame df.FT with % returns as serReturn

Trim df.FT dataframe of initial NA in prevPrice and serReturn
Final form of data frame df.FT
'data.frame':	1231 obs. of  5 variables:
 $ Date     : Date, format: "2013-07-08" "2013-07-09" ...
 $ Price    : num  1884 1881 1888 1898 1911 ...
 $ prevPrice: num  1885 1884 1881 1888 1898 ...
 $ serReturn: num  -0.0615 -0.1497 0.3881 0.4983 0.7019 ...
 $ frcReturn: num  -0.000615 -0.001497 0.003881 0.004983 0.007019 ...

Calculate absReturn and cumulative return in date order and add to df.FT
```

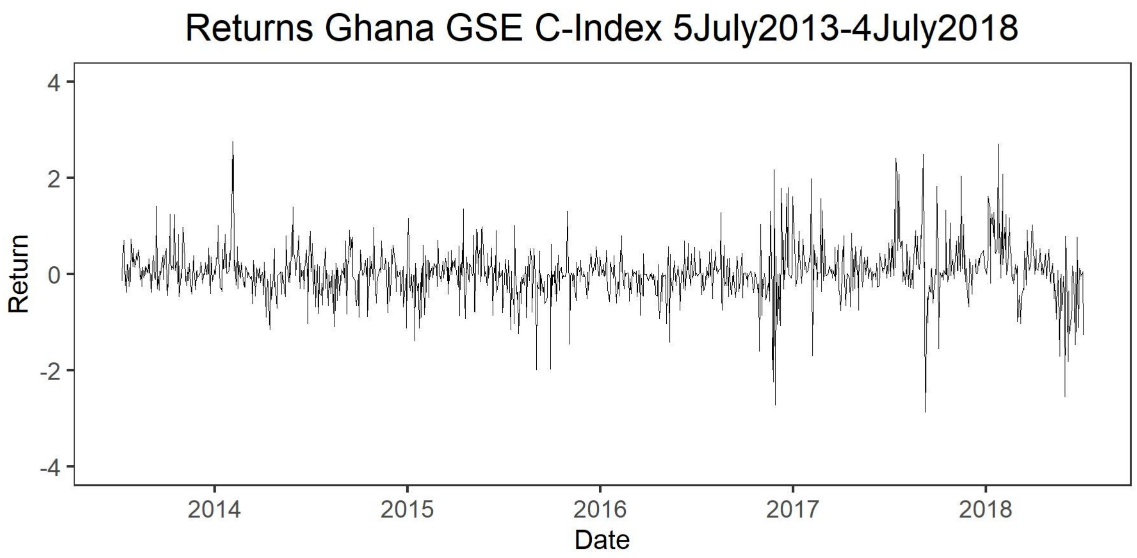
```
Structure of df.FT
'data.frame':	1231 obs. of  7 variables:
 $ Date     : Date, format: "2013-07-08" "2013-07-09" ...
 $ Price    : num  1884 1881 1888 1898 1911 ...
 $ prevPrice: num  1885 1884 1881 1888 1898 ...
 $ serReturn: num  -0.0615 -0.1497 0.3881 0.4983 0.7019 ...
 $ frcReturn: num  -0.000615 -0.001497 0.003881 0.004983 0.007019 ...
 $ absReturn: num  0.0615 0.1497 0.3881 0.4983 0.7019 ...
 $ cumReturn: num  0 -0.15 0.238 0.737 1.444 ...

Final trimmed form of df.FT with index, data lengths FTlengths = 1230
'data.frame':	1230 obs. of  8 variables:
 $ Date     : Date, format: "2013-07-09" "2013-07-10" ...
 $ Price    : num  1881 1888 1898 1911 1906 ...
 $ prevPrice: num  1884 1881 1888 1898 1911 ...
 $ serReturn: num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ frcReturn: num  -0.0015 0.00388 0.00498 0.00702 -0.00235 ...
 $ absReturn: num  0.15 0.388 0.498 0.702 0.235 ...
 $ cumReturn: num  -0.15 0.238 0.737 1.444 1.206 ...
 $ index    : num  1 2 3 4 5 6 7 8 9 10 ...

Write from df.FT to FTfullData.xlsx

Add index to df.FT)
Current form of df.FT)
'data.frame':	1230 obs. of  8 variables:
 $ Date     : Date, format: "2013-07-09" "2013-07-10" ...
 $ Price    : num  1881 1888 1898 1911 1906 ...
 $ prevPrice: num  1884 1881 1888 1898 1911 ...
 $ serReturn: num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ frcReturn: num  -0.0015 0.00388 0.00498 0.00702 -0.00235 ...
 $ absReturn: num  0.15 0.388 0.498 0.702 0.235 ...
 $ cumReturn: num  -0.15 0.238 0.737 1.444 1.206 ...
 $ index    : num  1 2 3 4 5 6 7 8 9 10 ...

write from df.FT with selected time period to xlsx file FTdataplotfile.xlsx

Add smooth Prices to df.FT

str(df.FT) with loess smooth prices added to df.FT
'data.frame':	1230 obs. of  9 variables:
 $ Date        : Date, format: "2013-07-09" "2013-07-10" ...
 $ Price       : num  1881 1888 1898 1911 1906 ...
 $ prevPrice   : num  1884 1881 1888 1898 1911 ...
 $ serReturn   : num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ frcReturn   : num  -0.0015 0.00388 0.00498 0.00702 -0.00235 ...
 $ absReturn   : num  0.15 0.388 0.498 0.702 0.235 ...
 $ cumReturn   : num  -0.15 0.238 0.737 1.444 1.206 ...
 $ index       : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Price_Smooth: num  1901 1903 1905 1908 1910 ...

str(df.FTzero, data frame for needle plot, and df.FT$Zeros
'data.frame':	1230 obs. of  3 variables:
 $ Yrtn : num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ Y0   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis: num  1 2 3 4 5 6 7 8 9 10 ...

Time Series of Volatiliy and its Smooth

Structure of df.FT with serVolatility
'data.frame':	1230 obs. of  11 variables:
 $ Date         : Date, format: "2013-07-09" "2013-07-10" ...
 $ Price        : num  1881 1888 1898 1911 1906 ...
 $ prevPrice    : num  1884 1881 1888 1898 1911 ...
 $ serReturn    : num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ frcReturn    : num  -0.0015 0.00388 0.00498 0.00702 -0.00235 ...
 $ absReturn    : num  0.15 0.388 0.498 0.702 0.235 ...
 $ cumReturn    : num  -0.15 0.238 0.737 1.444 1.206 ...
 $ index        : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Price_Smooth : num  1901 1903 1905 1908 1910 ...
 $ Zeros        : num  0 0 0 0 0 0 0 0 0 0 ...
 $ serVolatility: num  0.15 0.388 0.498 0.702 0.235 ...

str(df.FTzero, data frame with added Date, serVolatility, volatility_Smooth
'data.frame':  1230 obs. of  6 variables:
 $ Yrtn             : num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ Y0               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis            : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date             : Date, format: "2013-07-09" "2013-07-10" ...
 $ serVolatility    : num  0.15 0.388 0.498 0.702 0.235 ...
 $ volatility_Smooth: num  0.327 0.323 0.319 0.317 0.315 ...

Structure of dataframe for qqplot, returns.df , using package qqplotr
'data.frame':  1230 obs. of  1 variable:
 $ qqdata: num  -0.15 0.388 0.498 0.702 -0.235 ...

Price: min, max,  intv, breaks chosen
[1] 1527.94
[1] 3489.45
[1] 392.302
[1] 1527.940 1920.242 2312.544 2704.846 3097.148 3489.450

Return: min,max, intv, breaks chosen
[1] -11.68255
[1] 13.50979
[1] 5.038467
[1] -11.682548  -6.644080  -1.605613   3.432855   8.471322  13.509789

cumReturn: min,max, intv, breaka chosen
[1] -18.88925
[1] 85.23758
[1] 20.82537
[1] -18.889249   1.936117  22.761484  43.586850  64.412216  85.237582
[1]   -5  230  465  700  935 1170

Volatility: max, intv, breaks chosen
[1] 13.50979
[1] 2.701958
[1]  0.000000  2.701958  5.403916  8.105874 10.807832 13.509789
[1]  0  3  6  9 12 15

Structure of df.FT needed for plots
'data.frame':  1230 obs. of  12 variables:
 $ Date             : Date, format: "2013-07-09" "2013-07-10" ...
 $ Price            : num  1881 1888 1898 1911 1906 ...
 $ prevPrice        : num  1884 1881 1888 1898 1911 ...
 $ serReturn        : num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ frcReturn        : num  -0.0015 0.00388 0.00498 0.00702 -0.00235 ...
 $ absReturn        : num  0.15 0.388 0.498 0.702 0.235 ...
 $ cumReturn        : num  -0.15 0.238 0.737 1.444 1.206 ...
 $ index            : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Price_Smooth     : num  1901 1903 1905 1908 1910 ...
 $ Zeros            : num  0 0 0 0 0 0 0 0 0 0 ...
 $ serVolatility    : num  0.15 0.388 0.498 0.702 0.235 ...
 $ volatility_Smooth: num  0.327 0.323 0.319 0.317 0.315 ...
```

```
Structure of df.FTzero needed for plots
'data.frame':  1230 obs. of  6 variables:
 $ Yrtn             : num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ Y0               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis            : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date             : Date, format: "2013-07-09" "2013-07-10" ...
 $ serVolatility    : num  0.15 0.388 0.498 0.702 0.235 ...
 $ volatility_Smooth: num  0.327 0.323 0.319 0.317 0.315 ...

Calculations to inform construction of plots

Plotting Time Series in R with ggplot

plt1: Line plot of Price Time Series

plt1sm: Line plot of Price Time Series with loess smooth


Using cowplot package to save and display plt1sm as plt_ts1sm.jpeg

plt2: Line plot of Return Time Series

Using cowplot package to save and display plt2 as plt_ts2.jpeg

Using cowplot package for joint display of Price and Return and save
plt1and plt2 as plt1_2.jpeg

plt2cr: Cumulative returns plot here

Using cowplot package to save and display plt2cr as plt2cr_ts3.jpeg

plt3: Needle Plot of Return Time Series
 num [1:7] 0 200 400 600 800 1000 1200

Using cowplot package to save and display plt3 as plt_ts3.jpeg

plt5: Histogram of Returns

Using cowplot package to display and save  plt5 as plt_ts5.jpeg

plt6: QQ PLOT of Returns, using package qqplotr

Using cowplot package to display and save  plt6 as plt_ts6.jpeg

Calculation of Mean, Stdev, Skewness, Excess3-Kurtosis of serReturn
 num 0.0362
 num 0.736
 num 1.8
 - attr(*, "method")= chr "moment"
 num 143
 - attr(*, "method")= chr "excess"

calculation of Anderson Darling Test for Normality
Anderson Darling Statistic
      A
81.7444
Anderson Darling p.value
[1] 3.7e-24

plt4: Needle plot of Volatility with lowess Smooth

df.FTzero data frame
'data.frame':  1230 obs. of  6 variables:
 $ Yrtn             : num  -0.15 0.388 0.498 0.702 -0.235 ...
 $ Y0               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis            : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date             : Date, format: "2013-07-09" "2013-07-10" ...
 $ serVolatility    : num  0.15 0.388 0.498 0.702 0.235 ...
 $ volatility_Smooth: num  0.327 0.323 0.319 0.317 0.315 ...

Using cowplot package to display and save  plt4 as plt_ts4.jpeg

THIS IS END of FTplotSmoothRfunction.R

18.95 sec elapsed
There were 12 warnings (use warnings() to see them)
```

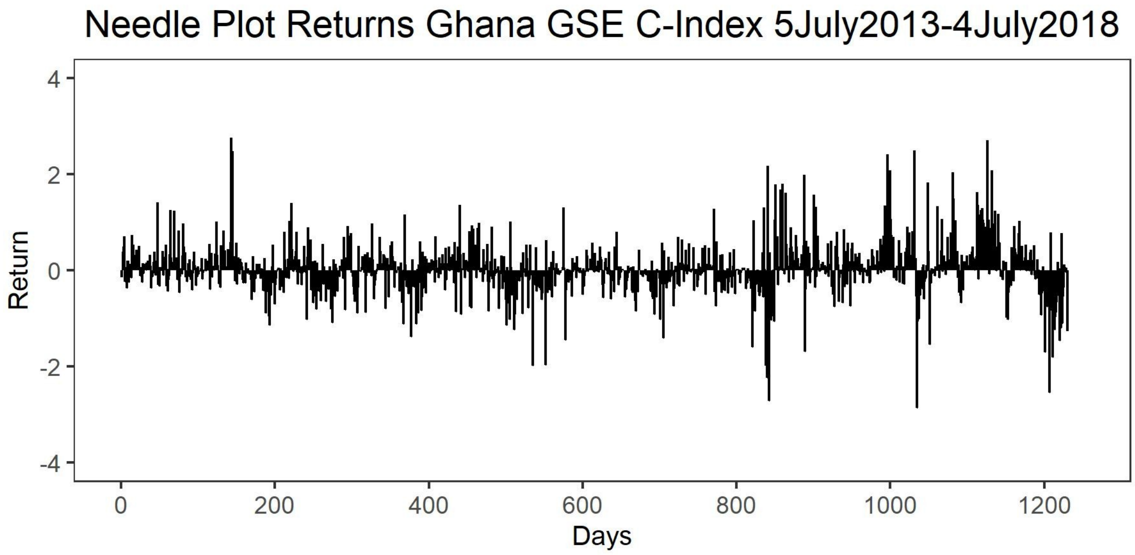
2- Graphics For the Ghana GSE-C-Index Time Series

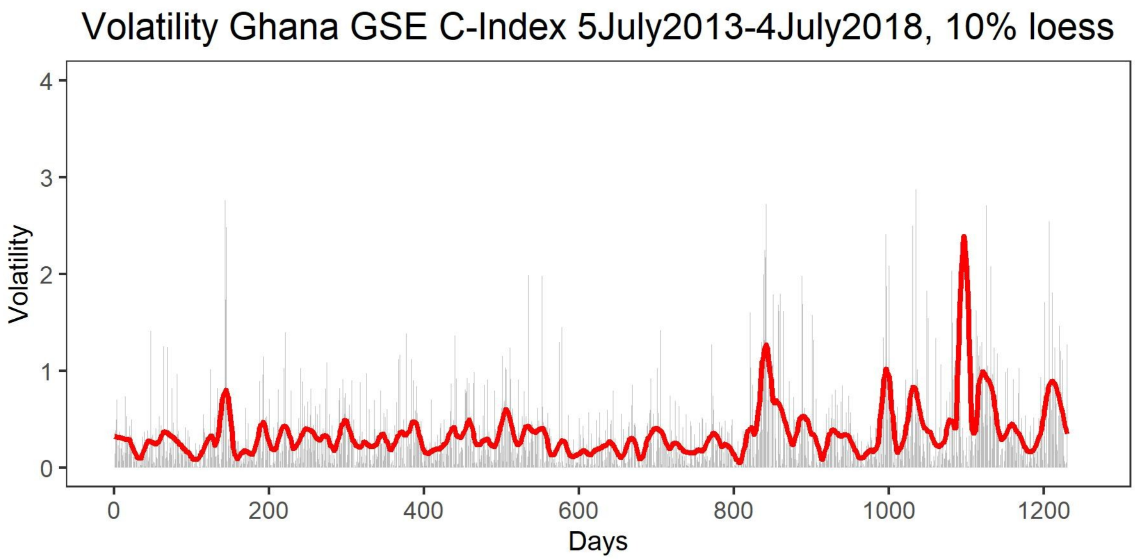


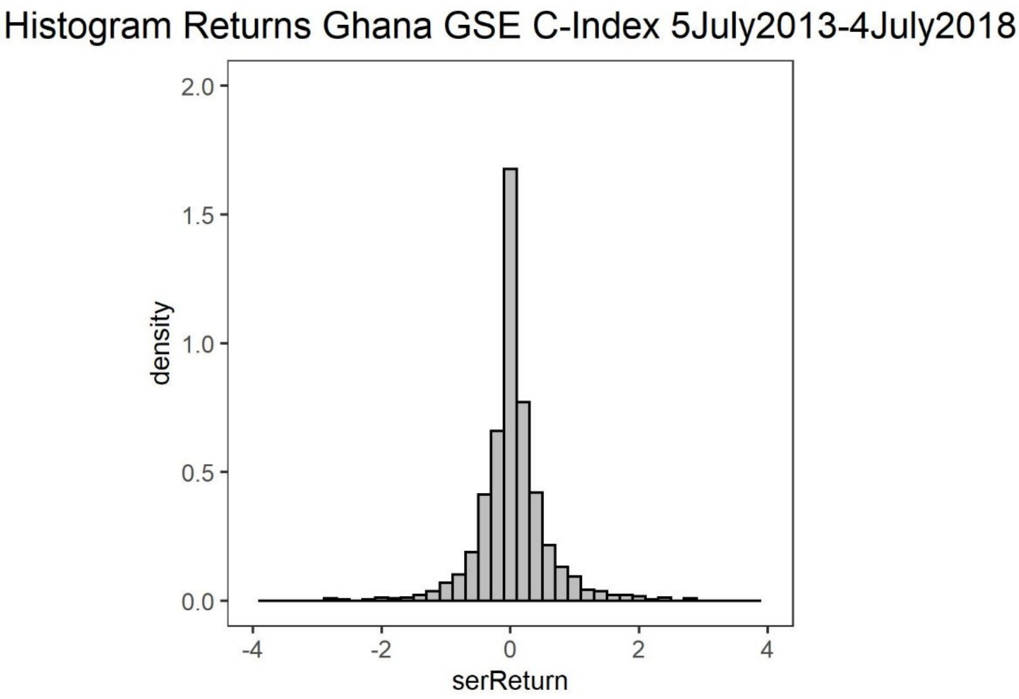
Price Ghana GSE C-Index 5July2013-4July2018, 15% loess

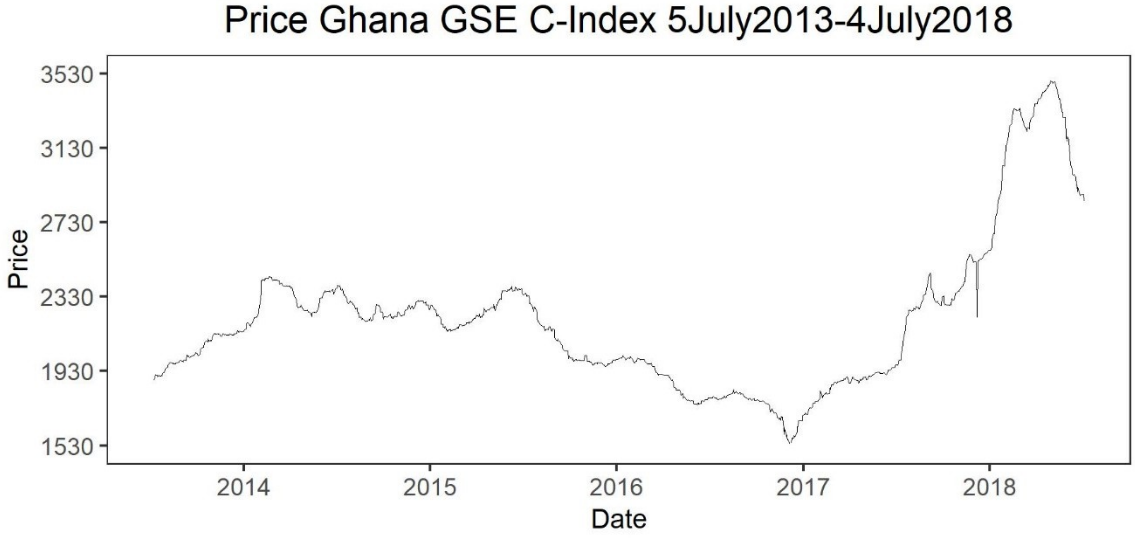


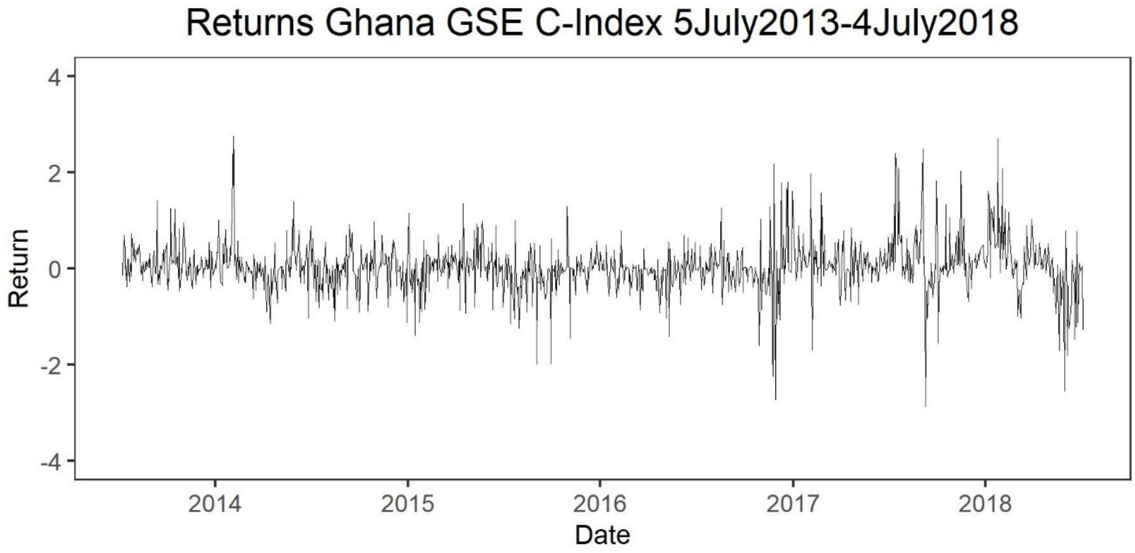
Returns Ghana GSE C-Index 5July2013-4July2018

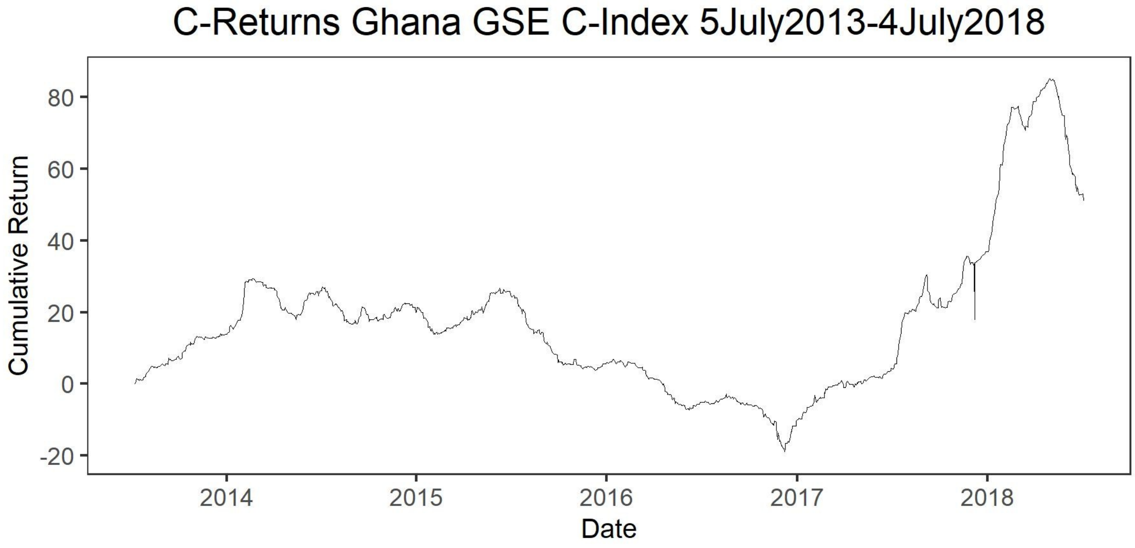

Needle Plot Returns Ghana GSE C-Index 5July2013-4July2018
Return
4
2
0
-2
-4
0
200
400
600
800
1000
1200
Days

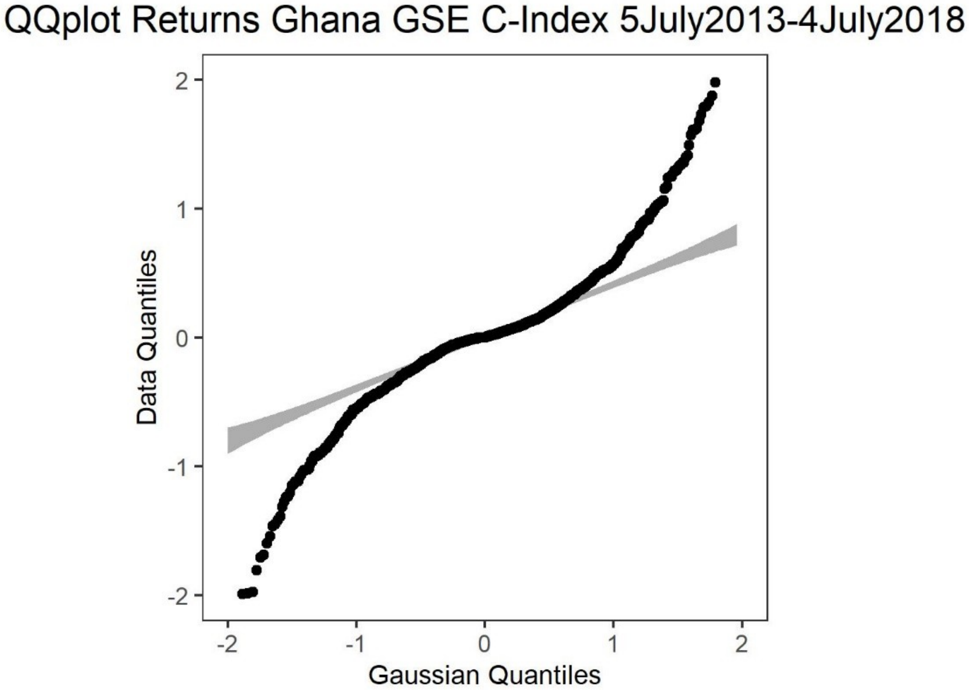

Volatility Ghana GSE C-Index 5July2013-4July2018, 10% loess
Volatility
4
3
2
1
0
0
200
400
600
800
1000
1200
Days


Histogram Returns Ghana GSE C-Index 5July2013-4July2018
density
2.0
1.5
1.0
0.5
0.0
-4
-2
0
2
4
serReturn


Price Ghana GSE C-Index 5July2013-4July2018
Price
3530
3130
2730
2330
1930
1530
2014
2015
2016
2017
2018
Date


Returns Ghana GSE C-Index 5July2013-4July2018
Return
4
2
0
-2
-4
2014
2015
2016
2017
2018
Date





3- GGplot Explanation for Pl1 graph

```
plt1 <- ggplot(aes(x=Date, y=Price), data=df.FT)
```

- Here we create a variable named pl1 that will plot price for Ghana GSE index across 2013-2018 includes the data_frame (***GSEC5July13-4July18.xlsx***) and associated x (date) and y (price)

```
plt1 <- plt1 + geom_line(size=0.01, color='black')
```

- Here we add data's graphical representation (geo_line) and the colour black

```
plt1 <- plt1 + scale_x_date(labels = date_format("%Y") , breaks =
date_breaks("1 year"))
```

- Here we adjust the date display format and the number of major and minor ticks for x axis date values using **scale_x_date.** We also create a tick for every year by **using breaks = date_breaks("1 year").**

```
plt1 <- plt1 + scale_y_continuous(limits=c(200,1200),
breaks=c(200,400,600,800,1000,1200))
```

- Here is for the format for the y axis

```
plt1 <- plt1 + ggtitle(Title_plt1)
```

- We add the title who will be in our case "Price Ghana GSE C-Index 5july2013-4july2018"

```
plt1 <- plt1 + coord_fixed(ratio=1/3)
plt1 <- plt1 + theme_bw()
```

- We adjust figure elements by using **theme_bw()**

```
plt1 <- plt1 + theme(plot.title=element_text(size=14,  hjust=0.5),
axis.title=element_text(size=10))
```

- theme(plot.title) allows to format the title separately from other text.

```
plt1 <- plt1 + theme(panel.grid.major=element_blank(),
panel.grid.minor=element_blank())
```

- We modify the plot panel and background

```
plt1 <- plt1 + theme(aspect.ratio=2/5)
```

- We fix the aspect ratio of the plot

```
plot_grid(plt1, nrow=1, ncol=1)
```

- We arrange our plot into a grid.

```
save_plot("plt_ts1.jpeg", plt1, nrow=1, ncol=1)
```

- We save the plot

```
print(plt1)
```

- We visualize the plot

II- Analysis on another daily financial: S&P Index

1- output from running *S&P_FTplotSmoothRfunction.R*

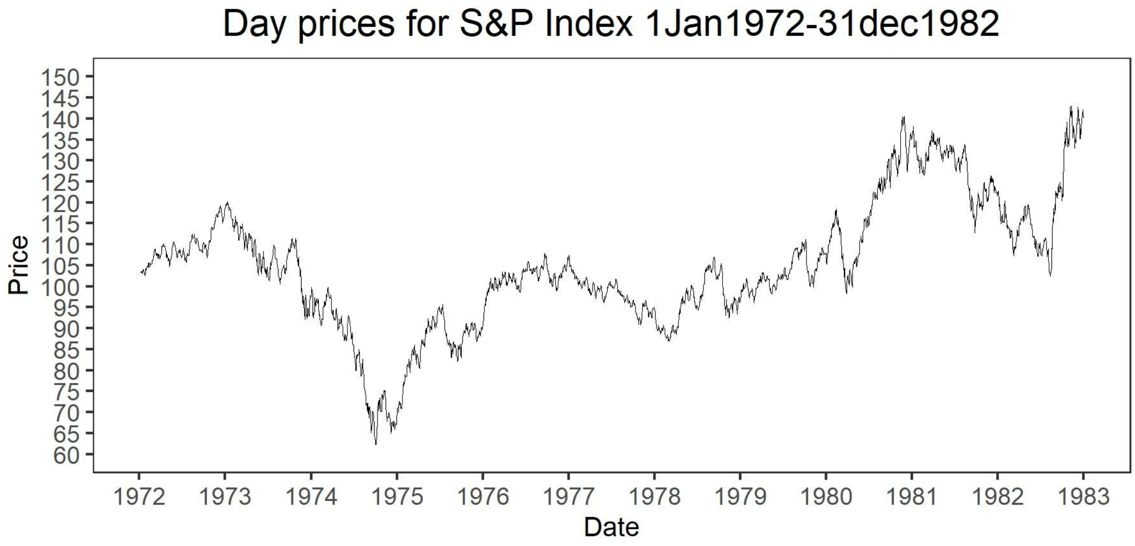
```
Read dateindexpricereturndata file into index-ordered dataframe  df.FT xlsx
file using openxls package

df.FT data frame from original xlsx data file - not usually with date as
date class - review file
'data.frame':  2777 obs. of  4 variables:
 $ index    : num  2 3 4 5 6 7 8 9 10 11 ...
 $ Date     : num  26302 26303 26304 26305 26308 ...
 $ Price    : num  102 103 104 103 103 ...
 $ serReturn: num  0.4123 0.9457 0.4357 -0.0387 -0.1451 ...

1. For Dates of the xlsx excel numerical form, conversion code with
origin=1899-12-30 available here

All dates have been created to be of standard date class form

df.FT data frame   -   Dates of date class
'data.frame':  2777 obs. of  4 variables:
 $ index    : num  2 3 4 5 6 7 8 9 10 11 ...
 $ Date     : Date, format: "1972-01-04" "1972-01-05" ...
 $ Price    : num  102 103 104 103 103 ...
 $ serReturn: num  0.4123 0.9457 0.4357 -0.0387 -0.1451 ...

[1] 2777
Calulate %returns and name serReturn
```

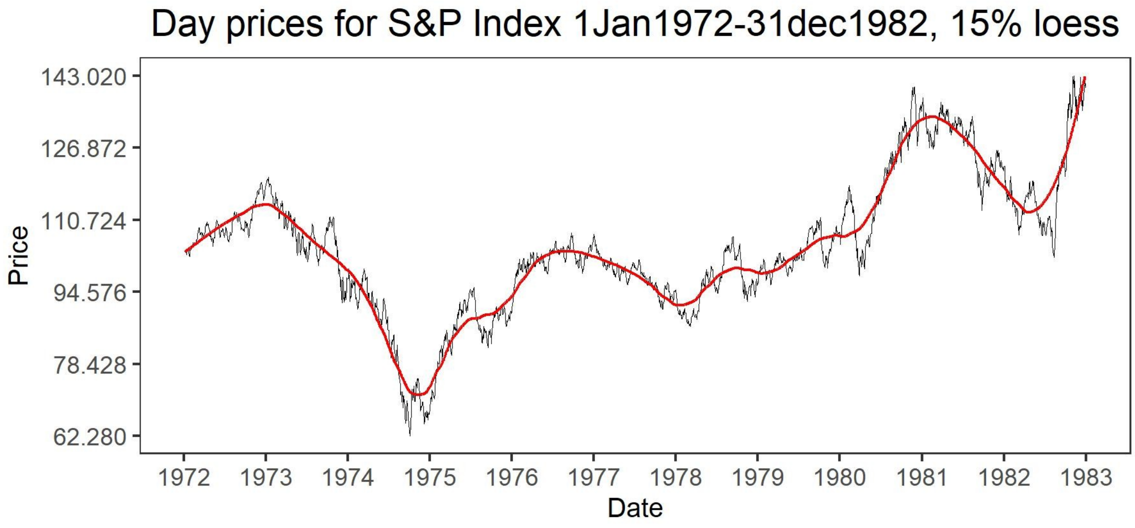
```
 num [1:2777] NA 102 103 104 103 ...
 num [1:2777] NA 0.9501 0.4366 -0.0386 -0.145 ...
 num [1:2777] NA 0.009501 0.004366 -0.000386 -0.00145 ...
Data frame df.FT with % returns as serReturn

Trim df.FT dataframe of initial NA in prevPrice and serReturn
Final form of data frame df.FT
'data.frame':  2776 obs. of  6 variables:
 $ index    : num  3 4 5 6 7 8 9 10 11 12 ...
 $ Date     : Date, format: "1972-01-05" "1972-01-06" ...
 $ Price    : num  103 104 103 103 104 ...
 $ serReturn: num  0.9501 0.4366 -0.0386 -0.145 0.3194 ...
 $ prevPrice: num  102 103 104 103 103 ...
 $ frcReturn: num  0.009501 0.004366 -0.000386 -0.00145 0.003194 ...

Calculate absReturn and cumulative return in date order and add to df.FT

Structure of df.FT
'data.frame':  2776 obs. of  8 variables:
 $ index    : num  3 4 5 6 7 8 9 10 11 12 ...
 $ Date     : Date, format: "1972-01-05" "1972-01-06" ...
 $ Price    : num  103 104 103 103 104 ...
 $ serReturn: num  0.9501 0.4366 -0.0386 -0.145 0.3194 ...
 $ prevPrice: num  102 103 104 103 103 ...
 $ frcReturn: num  0.009501 0.004366 -0.000386 -0.00145 0.003194 ...
 $ absReturn: num  0.9501 0.4366 0.0386 0.145 0.3194 ...
 $ cumReturn: num  0 0.437 0.398 0.252 0.572 ...

Final trimmed form of df.FT with index, data lengths FTlengths = 2775
'data.frame':  2775 obs. of  8 variables:
 $ index    : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date     : Date, format: "1972-01-06" "1972-01-07" ...
 $ Price    : num  104 103 103 104 104 ...
 $ serReturn: num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ prevPrice: num  103 104 103 103 104 ...
 $ frcReturn: num  0.004366 -0.000386 -0.00145 0.003194 -0.000579 ...
 $ absReturn: num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ cumReturn: num  0.437 0.398 0.252 0.572 0.514 ...

Write from df.FT to FTfullData.xlsx

Add index to df.FT)
Current form of df.FT)
'data.frame':  2775 obs. of  8 variables:
 $ index    : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date     : Date, format: "1972-01-06" "1972-01-07" ...
 $ Price    : num  104 103 103 104 104 ...
 $ serReturn: num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ prevPrice: num  103 104 103 103 104 ...
 $ frcReturn: num  0.004366 -0.000386 -0.00145 0.003194 -0.000579 ...
 $ absReturn: num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ cumReturn: num  0.437 0.398 0.252 0.572 0.514 ...

write from df.FT with selected time period to xlsx file FTdataplotfile.xlsx

Add smooth Prices to df.FT

str(df.FT) with loess smooth prices added to df.FT
'data.frame':  2775 obs. of  9 variables:
 $ index       : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date        : Date, format: "1972-01-06" "1972-01-07" ...
 $ Price       : num  104 103 103 104 104 ...
 $ serReturn   : num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ prevPrice   : num  103 104 103 103 104 ...
 $ frcReturn   : num  0.004366 -0.000386 -0.00145 0.003194 -0.000579 ...
 $ absReturn   : num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ cumReturn   : num  0.437 0.398 0.252 0.572 0.514 ...
 $ Price_Smooth: num  104 104 104 104 104 ...

str(df.FTzero, data frame for needle plot, and df.FT$Zeros
'data.frame':  2775 obs. of  3 variables:
 $ Yrtn : num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ Y0   : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis: num  1 2 3 4 5 6 7 8 9 10 ...

Time Series of Volatiliy and its Smooth

Structure of df.FT with serVolatility
'data.frame':  2775 obs. of  11 variables:
 $ index        : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date         : Date, format: "1972-01-06" "1972-01-07" ...
 $ Price        : num  104 103 103 104 104 ...
 $ serReturn    : num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ prevPrice    : num  103 104 103 103 104 ...
 $ frcReturn    : num  0.004366 -0.000386 -0.00145 0.003194 -0.000579 ...
 $ absReturn    : num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ cumReturn    : num  0.437 0.398 0.252 0.572 0.514 ...
 $ Price_Smooth : num  104 104 104 104 104 ...
 $ Zeros        : num  0 0 0 0 0 0 0 0 0 0 ...
 $ serVolatility: num  0.4366 0.0386 0.145 0.3194 0.0579 ...

str(df.FTzero, data frame with added Date, serVolatility, volatility_Smooth
'data.frame':  2775 obs. of  6 variables:
 $ Yrtn             : num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ Y0               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis            : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date             : Date, format: "1972-01-06" "1972-01-07" ...
 $ serVolatility    : num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ volatility_Smooth: num  0.295 0.297 0.299 0.301 0.303 ...

Structure of dataframe for qqplot, returns.df , using package qqplotr
'data.frame':  2775 obs. of  1 variable:
 $ qqdata: num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...

Price: min, max,  intv, breaks chosen
[1] 62.28
[1] 143.02
[1] 16.148
[1]  62.280  78.428  94.576 110.724 126.872 143.020

Return: min,max, intv, breaks chosen
[1] -3.968883
[1] 4.7555
[1] 1.744877
[1] -3.9688828 -2.2240062 -0.4791297  1.2657469  3.0106235  4.7555000

cumReturn: min,max, intv, breaka chosen
[1] -39.56918
[1] 38.77353
[1] 15.66854
[1] -39.569183 -23.900640  -8.232098   7.436445  23.104987  38.773530
[1]   -5  230  465  700  935 1170

Volatility: max, intv, breaks chosen
[1] 4.7555
```

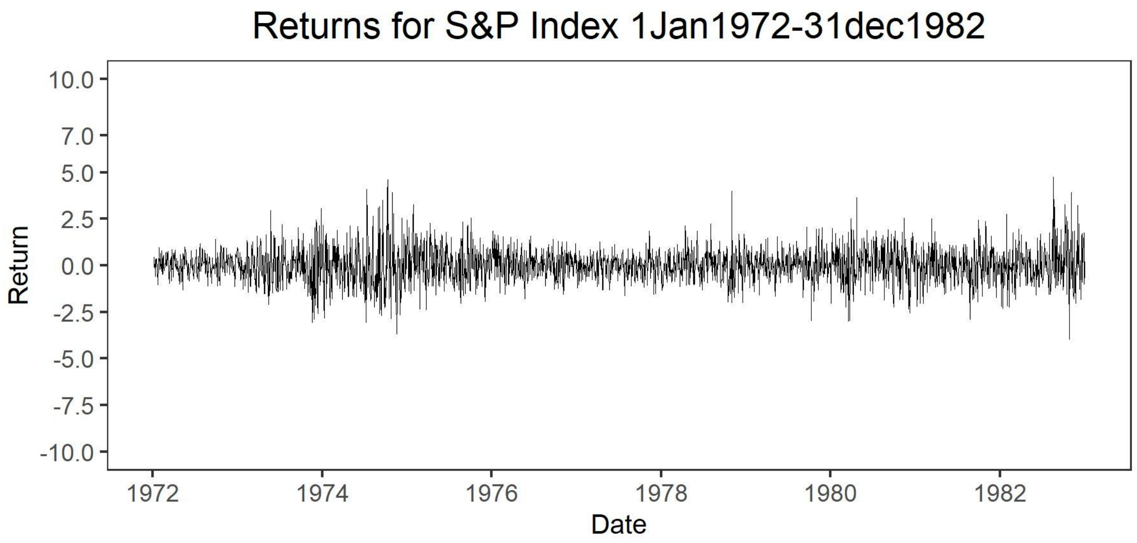
```
[1] 0.9511
[1] 0.0000 0.9511 1.9022 2.8533 3.8044 4.7555
[1]  0  3  6  9 12 15

Structure of df.FT needed for plots
'data.frame':	2775 obs. of  12 variables:
 $ index           : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date            : Date, format: "1972-01-06" "1972-01-07" ...
 $ Price           : num  104 103 103 104 104 ...
 $ serReturn       : num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ prevPrice       : num  103 104 103 103 104 ...
 $ frcReturn       : num  0.004366 -0.000386 -0.00145 0.003194 -
0.000579 ...
 $ absReturn       : num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ cumReturn       : num  0.437 0.398 0.252 0.572 0.514 ...
 $ Price_Smooth    : num  104 104 104 104 104 ...
 $ Zeros           : num  0 0 0 0 0 0 0 0 0 0 ...
 $ serVolatility   : num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ volatility_Smooth: num  0.295 0.297 0.299 0.301 0.303 ...

Structure of df.FTzero needed for plots
'data.frame':	2775 obs. of  6 variables:
 $ Yrtn            : num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ Y0              : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis           : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date            : Date, format: "1972-01-06" "1972-01-07" ...
 $ serVolatility   : num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ volatility_Smooth: num  0.295 0.297 0.299 0.301 0.303 ...

Calculations to inform construction of plots

Plotting Time Series in R with ggplot

plt1: Line plot of Price Time Series

plt1sm: Line plot of Price Time Series with loess smooth


Using cowplot package to save and display plt1sm as plt_ts1sm.jpeg

plt2: Line plot of Return Time Series

Using cowplot package to save and display plt2 as plt_ts2.jpeg

Using cowplot package for joint display of Price and Return and save
plt1and plt2 as plt1_2.jpeg

plt2cr: Cumulative returns plot here

Using cowplot package to save and display plt2cr as plt2cr_ts3.jpeg

plt3: Needle Plot of Return Time Series
 num [1:7] 0 200 400 600 800 1000 1200

Using cowplot package to save and display plt3 as plt_ts3.jpeg

plt5: Histogram of Returns

Using cowplot package to display and save  plt5 as plt_ts5.jpeg

plt6: QQ PLOT of Returns, using package qqplotr

Using cowplot package to display and save  plt6 as plt_ts6.jpeg
```

```
Calculation of Mean, Stdev, Skewness, Excess3-Kurtosis of serReturn
 num 0.0154
 num 0.912
 num 0.267
 - attr(*, "method")= chr "moment"
 num 1.81
 - attr(*, "method")= chr "excess"

calculation of Anderson Darling Test for Normality
Anderson Darling Statistic
       A 
8.216824 
Anderson Darling p.value
[1] 5.362021e-20

plt4: Needle plot of Volatility with lowess Smooth

df.FTzero data frame
'data.frame':	2775 obs. of  6 variables:
 $ Yrtn             : num  0.4366 -0.0386 -0.145 0.3194 -0.0579 ...
 $ Y0               : num  0 0 0 0 0 0 0 0 0 0 ...
 $ Xaxis            : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Date             : Date, format: "1972-01-06" "1972-01-07" ...
 $ serVolatility    : num  0.4366 0.0386 0.145 0.3194 0.0579 ...
 $ volatility_Smooth: num  0.295 0.297 0.299 0.301 0.303 ...

Using cowplot package to display and save  plt4 as plt_ts4.jpeg

THIS IS END of FTplotSmoothRfunction.R

18.92 sec elapsed
There were 21 warnings (use warnings() to see them)
```

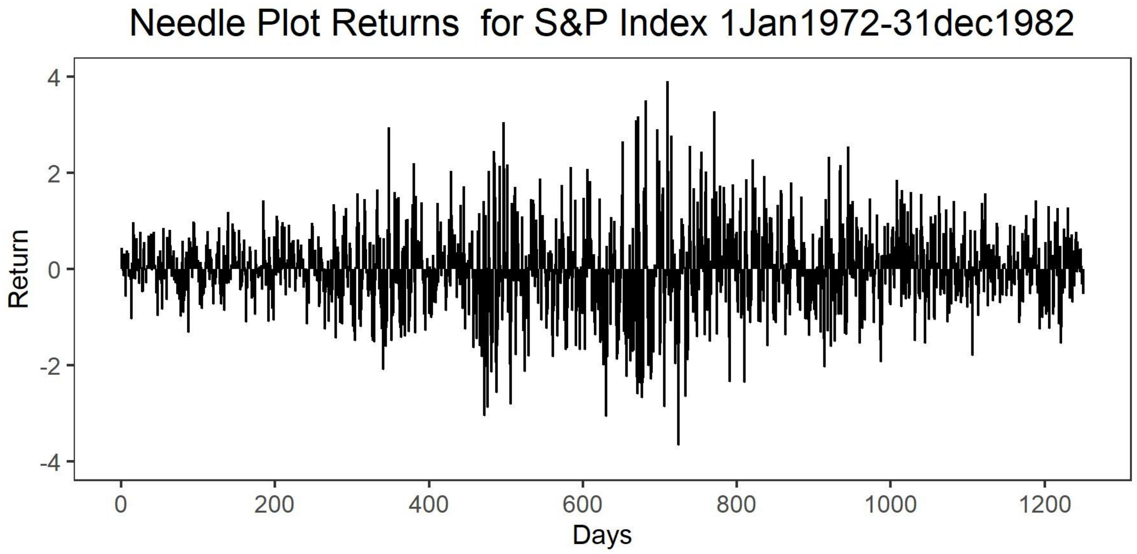
2- Visualization of the different graphs


Day prices for S&P Index 1Jan1972-31dec1982
Price
150
145
140
135
130
125
120
115
110
105
100
95
90
85
80
75
70
65
60
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
Date

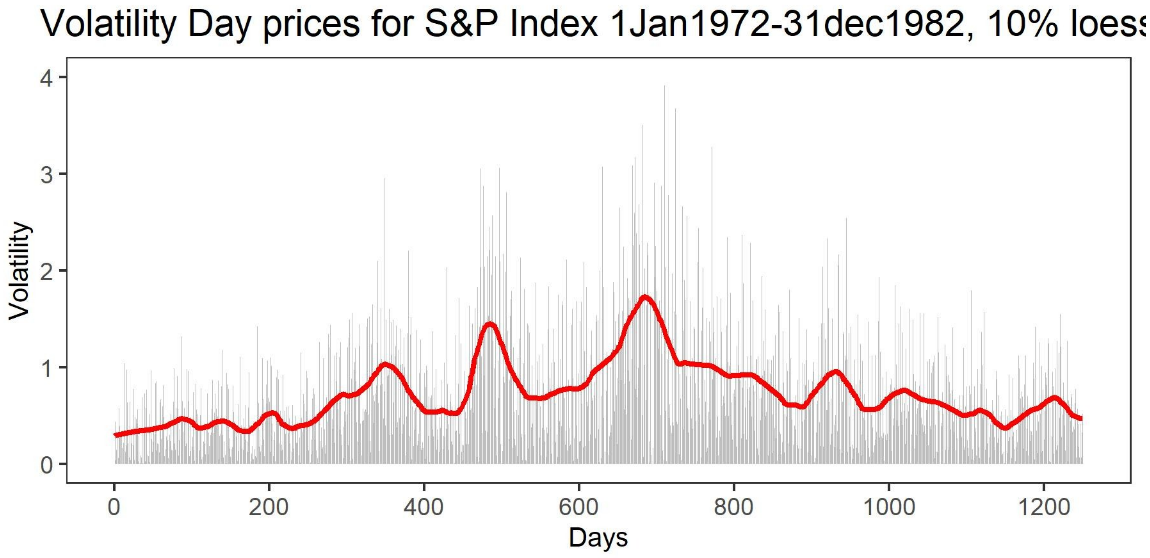

Day prices for S&P Index 1Jan1972-31dec1982, 15% loess
Price
143.020
126.872
110.724
94.576
78.428
62.280
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
Date

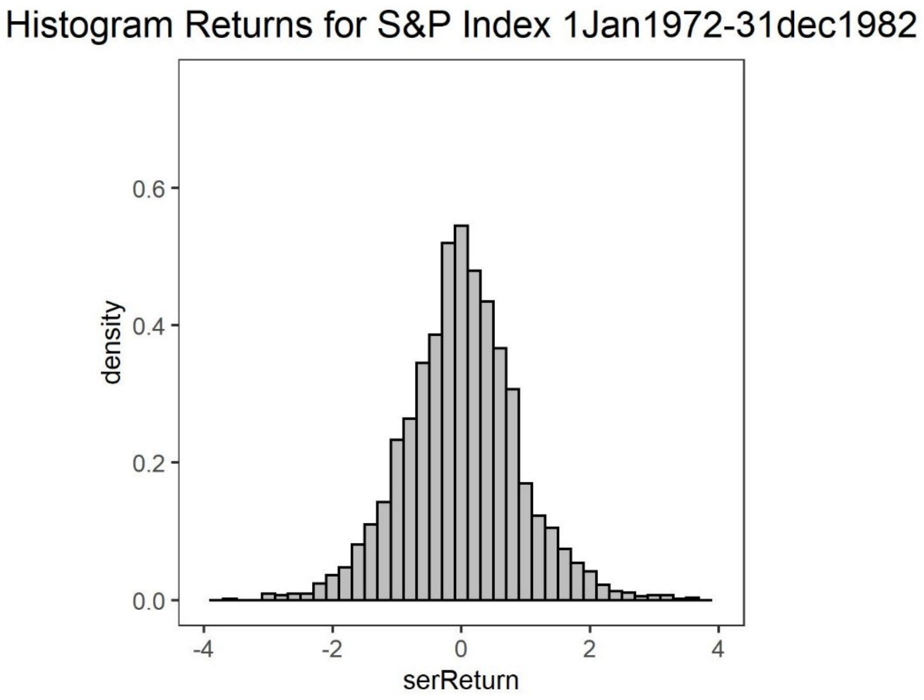


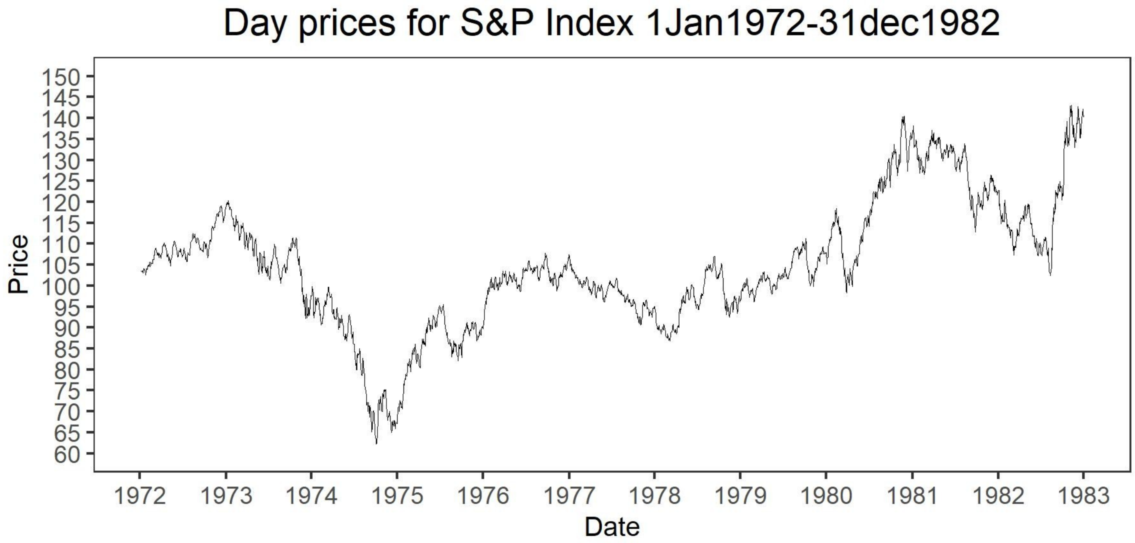
Returns for S&P Index 1Jan1972-31dec1982

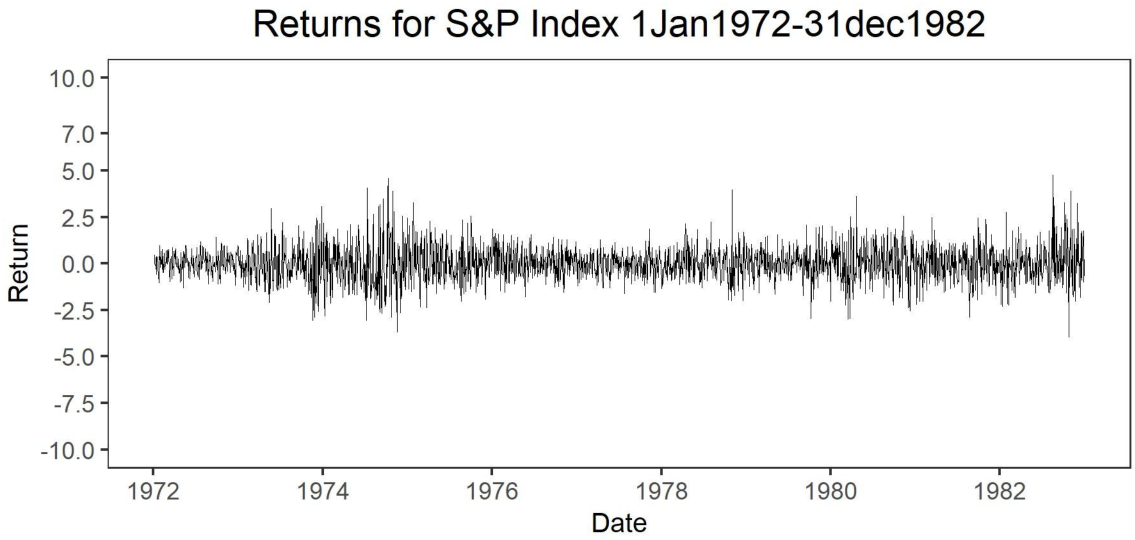


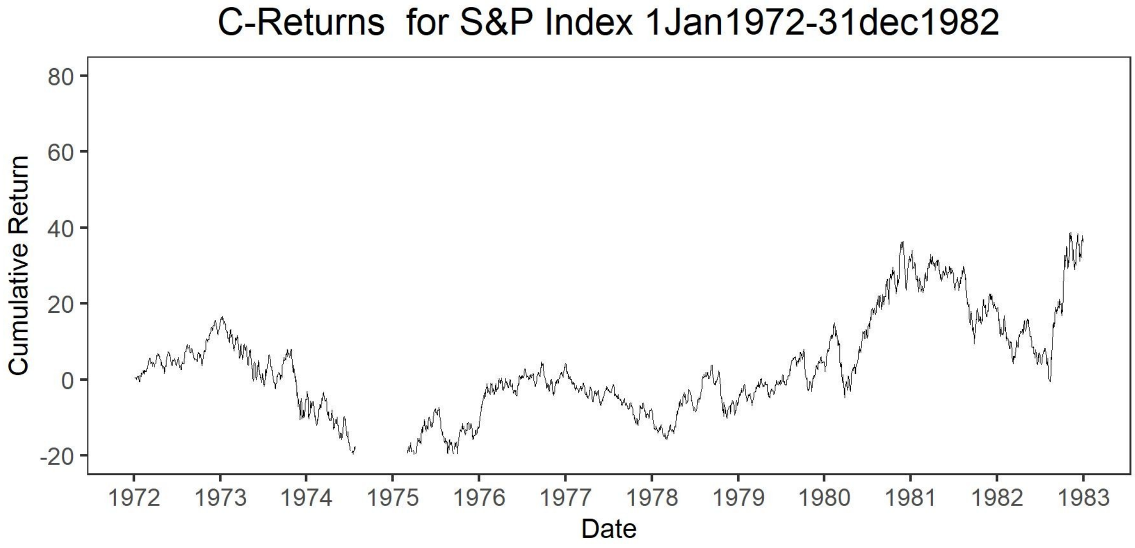
Needle Plot Returns for S&P Index 1Jan1972-31dec1982

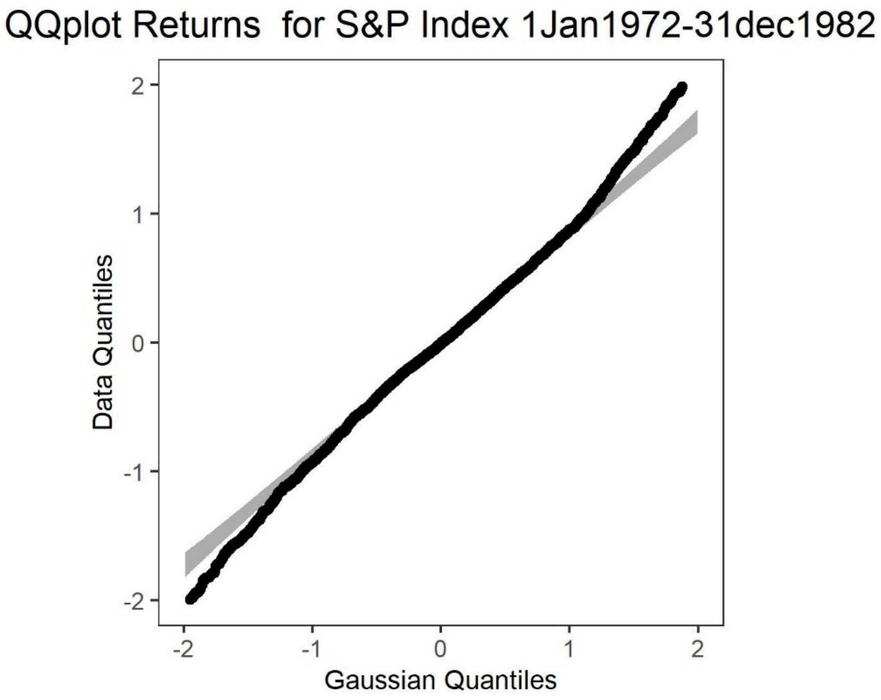

Volatility Day prices for S&P Index 1Jan1972-31dec1982, 10% loess
Volatility
4
3
2
1
0
0
200
400
600
800
1000
1200
Days


Histogram Returns for S&P Index 1Jan1972-31dec1982
density
0.6
0.4
0.2
0.0
-4
-2
0
2
4
serReturn


Day prices for S&P Index 1Jan1972-31dec1982
Price
150
145
140
135
130
125
120
115
110
105
100
95
90
85
80
75
70
65
60
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
Date


Returns for S&P Index 1Jan1972-31dec1982
Return
10.0
7.0
5.0
2.5
0.0
-2.5
-5.0
-7.5
-10.0
1972
1974
1976
1978
1980
1982
Date


C-Returns for S&P Index 1Jan1972-31dec1982
Cumulative Return
80
60
40
20
0
-20
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
Date


QQplot Returns for S&P Index 1Jan1972-31dec1982
Data Quantiles
2
1
0
-1
-2
-2
-1
0
1
2
Gaussian Quantiles