Configuration Manual

MSc Research Project
Masters in Fintech

Oluwagbenga Abayomi Seyingbo
x18126421@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Noel Cosgrave

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Oluwagbenga Abayomi Seyingbo
Student ID:	X18126421@student.ncirl.ie
Programme:	Masters in Fintech **Year:** 2019
Module:	Research Project
Supervisor:	Noel Cosgrave
Submission Due Date:	August 12th, 2019
Project Title:	**A Comparative Analysis of Stock Market Volatility**
Word Count:	**1694** **Page Count** **12**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………

Date: ………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Oluwagbenga Abayomi Seyingbo
x18126421@student.ncirl.ie

1 Introduction

This configuration manual consists of the computer system, **R** programming software, **R** programming packages, **R** programming codes and graphical illustration of datasets which was used to compare the stock market volatility, using the developed stock market index such as S & P 500, Nasdaq Composites and Dow Jones Industrial Average as a case study.

2 Computer System

The specification of the computer system used to carry out the research studies are listed below:

(i). Windows edition: Windows 10
(ii). Processor: intel (R) core (TM) i5 3320m CPU @ 2.60GHz 2.60GHz
(iii). Installed memory (RAM): 8.00 GB (7.70 GB usable)
(iv). System type: 64-bit Operating System, x64 based processor
(v). pen and touch: Touch Support with 2 touch points
(vi). Computer name: DESKTOP-RKOJC3P
(vii). Windows Product ID: 00326- 10000- 00000- AA242

3 R programming Software

The r programming software comprises of RStudio and **R** software, the RStudio is a set of integrated tools designed to help **R** be more productive for analytics. The R software version is 3.5.1: 62 megabytes and 64 bits

4 R code

```
##Loading Data from yahoo Finance:
## Deveoped Stock Market Index
## GSPC = S & P 500
## IXIC = NASDAQ
## DJI = Dow Jones Industrial Average
getSymbols("^GSPC", from="2015-01-01", to="2019-06-30")
getSymbols("^IXIC", from="2015-01-01", to="2019-06-30")
getSymbols("^DJI", from="2015-01-01", to="2019-06-30")
##DATASET ONE S&P 500 (GSPC)
##FEATURE ENGINEERING OF S&P 500 (GSPC)
## PART ONE: Finding mising values
z <- c(1:3,NA)
is.na(z)
```

```
sum(is.na(z))
sapply(GSPC, FUN = function(x) {sum(is.na(x))})
unique(z)
length(unique(z))
##PART Two: Showing the Trending analysis and descriptive statistics of the Close price of the stock
market index
## The Data is the Daily close price
barplot(GSPC$GSPC.Close)
hist(GSPC$GSPC.Close,
   main="Distribution of GSPC.Close",
   xlab="GSPC.Close")
qqnorm(GSPC$GSPC.Close,
     main="Distribution of GSPC.Close",
     xlab="GSPC.Close")
sd(GSPC$GSPC.Close)
skewness(GSPC$GSPC.Close)
kurtosis(GSPC$GSPC.Close)
##Chat Series of the Dataset(GSPC)
chartSeries(GSPC)
## When we are estimating volatility, we work with returns either daily or monthly
## Estimating Volatility Models
##Chart Series of Daily Return (GSPC)
S.P500 <- dailyReturn(GSPC)
chartSeries(S.P500)
acf(S.P500)
pacf(S.P500)
###Descriptive Statistics of Daily Returns of S & P 500
min(S.P500)
max(S.P500)
mean(S.P500)
median(S.P500)
var(S.P500)
sd(S.P500)
skewness(S.P500)
kurtosis(S.P500)
##Stationarity test: Augment Dickey-Fuller test
adf.test(S.P500)
##BOX-LJUNG TEST
Box.test(S.P500, lag=10, type="Ljung")
##ARCH TEST
ArchTest(S.P500)
##Model Specification
Modelspec <- auto.arima(S.P500, trace = TRUE, test = "kpss", ic= "bic")
S.P500ug_spec = ugarchspec()
S.P500ug_spec
## Model Estimation
S.P500ugfit = ugarchfit(spec = S.P500ug_spec, data = S.P500)
S.P500ugfit
names(S.P500ugfit@model)
names(S.P500ugfit@fit)
summary(S.P500ugfit)
plot(S.P500ugfit)
```

```
##Measurement of Value at Risk
ctrl = list(tol = 1e-7, delta = 1e-9)
S.P500ugfit_roll <- ugarchroll(S.P500ug_spec, S.P500, n.start = 250, refit.every = 1,
                        refit.window = "moving", solver = "hybrid", calculate.VaR = TRUE,
                        VaR.alpha = 0.01, keep.coef = TRUE, solver.control = ctrl,
                        fit.control = list(scale = 1))
report(S.P500ugfit_roll, type = "VaR", VaR.alpha = 0.01, conf.level =0.99)
plot(S.P500ugfit)
## Extraction of Estimated Coefficients
S.P500ugfit@fit$coef
## Extraction of Estimated Conditional Variances
S.P500ug_var <- S.P500ugfit@fit$var
## Extraction of Estimated Squared residuals
S.P500ug_res2 <- (S.P500ugfit@fit$residuals)^2
##Plotting the squared residuals and the estimated conditional variances
plot(S.P500ug_res2, type ="c")
lines(S.P500ug_var, col = "green")
## Model Forecasting
S.P500ugfore <- ugarchforecast(S.P500ugfit, n.ahead = 10, data = S.P500)
plot(S.P500ugfore)
S.P500ugfore
S.P500ug_f <- S.P500ugfore@forecast$sigmaFor
plot(S.P500ug_f, type = "c")
##Note that volatility is the square root of the conditional variance
S.P500ug_var_t <- c(tail(S.P500ug_var,20), rep(NA,10)) #get the last 20 observations
S.P500ug_res2_t <- c(tail(S.P500ug_res2,20), rep(NA,10)) #get the last 20 observations
S.P500ug_f <- c(rep(NA,20),(S.P500ug_f)^2)
plot(S.P500ug_res2_t, type ="c")
lines(S.P500ug_f, col ="orange")
lines(S.P500ug_var_t, col ="green")
plot(S.P500ugfit)
##GJRGARCH
S.P500.gjrgarch.spec <- ugarchspec(variance.model = list(model="gjrGARCH",
garchOrder=c(1,1)),distribution.model = "norm")
S.P500.gjrgarch.fit = ugarchfit(S.P500.gjrgarch.spec, S.P500)
S.P500.gjrgarch.fit
coef(S.P500.gjrgarch.fit)
##GJRGARCH FORECASTING
S.P500.gjrgarch_Pred <- ugarchforecast(S.P500.gjrgarch.fit, n.ahead = 20, plot = TRUE)
plot(S.P500.gjrgarch_Pred)
##EGARCH
S.P500.egarch.spec = ugarchspec(variance.model=list(model="eGARCH",garchOrder=c(1,1)),
mean.model=list(armaOrder=c(0,0)))
S.P500.egarch.fit = ugarchfit(S.P500.egarch.spec, S.P500)
S.P500.egarch.fit
coef(S.P500.egarch.fit)
##EGARCH FORECASTING
S.P500.egarch_Pred <- ugarchforecast(S.P500.egarch.fit, n.ahead = 20, plot = TRUE)
plot(S.P500.egarch_Pred)
##DATASET TWO NASDAQ (IXIC)
##FEATURE ENGINEERING OF NASDAQ (IXIC)
## PART ONE: Finding mising values
```

```
z <- c(1:3,NA)
is.na(z)
sum(is.na(z))
sapply(IXIC, FUN = function(x) {sum(is.na(x))})
unique(z)
length(unique(z))
##PART Two: Showing the Trending analysis and descriptive statistics of the Close price of the stock market index
## The Data is the Daily close price
barplot(IXIC$IXIC.Close)
hist(IXIC$IXIC.Close,
    main="Distribution of IXIC.Close",
    xlab="IXIC.Close")
qqnorm(IXIC$IXIC.Close,
      main="Distribution of IXIC.Close",
      xlab="IXIC.Close")
sd(IXIC$IXIC.Close)
skewness(IXIC$IXIC.Close)
kurtosis(IXIC$IXIC.Close)
##Chat Series of the Dataset(IXIC)
chartSeries(IXIC)
## When we are estimating volatility, we work with returns either daily or monthly
## Estimating Volatility Models
##Chart Series of Daily Return (IXIC)
NASDAQ <- dailyReturn(IXIC)
chartSeries(NASDAQ)
acf(NASDAQ)
pacf(NASDAQ)
###Descriptive Statistics of Daily Returns of NASDAQ
min(NASDAQ)
max(NASDAQ)
mean(NASDAQ)
median(NASDAQ)
skewness(NASDAQ)
kurtosis(NASDAQ)
var(NASDAQ)
sd(NASDAQ)
##Stationarity test: Augment Dickey-Fuller test
adf.test(NASDAQ)
##BOX-LJUNG TEST
Box.test(NASDAQ, lag=10, type="Ljung")
##ARCH TEST
ArchTest(NASDAQ)
Modelspec <- auto.arima(NASDAQ, trace = TRUE, test = "kpss", ic= "bic")
##Model Specification
NASDAQug_spec = ugarchspec()
NASDAQug_spec
##Measurement of Value at Risk
ctrl = list(tol = 1e-7, delta = 1e-9)
NASDAQugfit_roll <- ugarchroll(NASDAQug_spec, S.P500, n.start = 250, refit.every = 1,
                        refit.window = "moving", solver = "hybrid", calculate.VaR = TRUE,
                        VaR.alpha = 0.01, keep.coef = TRUE, solver.control = ctrl,
```

```
                    fit.control = list(scale = 1))
report(NASDAQugfit_roll, type = "VaR", VaR.alpha = 0.01, conf.level =0.99)
## Model Estimation
NASDAQugfit = ugarchfit(spec = NASDAQug_spec, data = NASDAQ)
NASDAQugfit
names(NASDAQugfit@model)
names(NASDAQugfit@fit)
summary(NASDAQugfit)
plot(NASDAQugfit)
## Extraction of Estimated Coefficients
NASDAQugfit@fit$coef
## Extraction of Estimated Conditional Variances
NASDAQug_var <- NASDAQugfit@fit$var
## Extraction of Estimated Squared residuals
NASDAQug_res2 <- (NASDAQugfit@fit$residuals)^2
##Plotting the squared residuals and the estimated conditional variances
plot(NASDAQug_res2, type ="c")
lines(NASDAQug_var, col = "green")
## Model Forecasting
NASDAQugfore <- ugarchforecast(NASDAQugfit, n.ahead = 10)
NASDAQugfore
NASDAQug_f <- NASDAQugfore@forecast$sigmaFor
plot(NASDAQug_f, type = "c")
plot(NASDAQugfore)
##GJRGARCH
NASDAQgjrgarch.spec <- ugarchspec(variance.model = list(model="gjrGARCH",
garchOrder=c(1,1)),distribution.model = "norm")
NASDAQgjrgarch.fit = ugarchfit(NASDAQgjrgarch.spec, NASDAQ)
NASDAQgjrgarch.fit
coef(NASDAQgjrgarch.fit)
##GJRGARCH FORECASTING
NASDAQgjrgarch_Pred <- ugarchforecast(NASDAQgjrgarch.fit, n.ahead = 10, plot = TRUE)
plot(NASDAQgjrgarch_Pred)
##EGARCH
NASDAQ.egarch.spec = ugarchspec(variance.model=list(model="eGARCH",garchOrder=c(1,1)),
mean.model=list(armaOrder=c(0,0)))
NASDAQ.egarch.fit = ugarchfit(NASDAQ.egarch.spec, NASDAQ)
NASDAQ.egarch.fit
coef(NASDAQ.egarch.fit)
##EGARECH FORECASTING
NASDAQ.egarch_Pred <- ugarchforecast(NASDAQ.egarch.fit, n.ahead = 10, plot = TRUE)
plot(NASDAQ.egarch_Pred)
##DATASET THREE DOW JONE INDUSTRIAL AVERAGE (DJI)
##FEATURE ENGINEERING OF DOW JONE INDUSTRIAL AVERAGE (DJI)
## PART ONE: Finding mising values
z <- c(1:3,NA)
is.na(z)
sum(is.na(z))
sapply(DJI, FUN = function(x) {sum(is.na(x))})
unique(z)
length(unique(z))
```

```
##PART Two: Showing the Trending analysis and descriptive statistics of the Close price of the stock
market index
## The Data is the Daily close price
barplot(DJI$DJI.Close)
hist(DJI$DJI.Close,
   main="Distribution of DJI.Close",
   xlab="DJI.Close")
qqnorm(DJI$DJI.Close,
     main="Distribution of DJI.Close",
     xlab="DJI.Close")
sd(DJI$DJI.Close)
skewness(DJI$DJI.Close)
kurtosis(DJI$DJI.Close)
##Chat Series of the Dataset(DJI)
chartSeries(DJI)
## When we are estimating volatility, we work with returns either daily or monthly
## Estimating Volatility Models
##Chart Series of Daily Return (DJI)
DOWJONES <- dailyReturn(DJI)
chartSeries(DOWJONES)
acf(DOWJONES)
pacf(DOWJONES)
###Descriptive Statistics of Daily Returns of DOWJONES
min(DOWJONES)
max(DOWJONES)
mean(DOWJONES)
median(DOWJONES)
skewness(DOWJONES)
kurtosis(DOWJONES)
var(DOWJONES)
sd(DOWJONES)
##Stationarity test: Augment Dickey-Fuller test
adf.test(DOWJONES)
##BOX-LJUNG TEST
Box.test(DOWJONES, lag=10, type="Ljung")
##ARCH TEST
ArchTest(DOWJONES)
##Model Specification
Modelspec <- auto.arima(DOWJONES, trace = TRUE, test = "kpss", ic= "bic")
DOWJONESug_spec = ugarchspec()
DOWJONESug_spec
##Measurement of Value at Risk
ctrl = list(tol = 1e-7, delta = 1e-9)
DOWJONESugfit_roll <- ugarchroll(DOWJONESug_spec, S.P500, n.start = 250, refit.every = 1,
                        refit.window = "moving", solver = "hybrid", calculate.VaR = TRUE,
                        VaR.alpha = 0.01, keep.coef = TRUE, solver.control = ctrl,
                        fit.control = list(scale = 1))
report(DOWJONESugfit_roll, type = "VaR", VaR.alpha = 0.01, conf.level =0.99)
## Model Estimation
DOWJONESugfit = ugarchfit(spec = DOWJONESug_spec, data = DOWJONES)
DOWJONESugfit
names(DOWJONESugfit@model)
```

```
names(DOWJONESugfit@fit)
plot(DOWJONESugfit)
## Extraction of Estimated Coefficients
DOWJONESugfit@fit$coef
## Extraction of Estimated Conditional Variances
DOWJONESug_var <- DOWJONESugfit@fit$var
## Extraction of Estimated Squared residuals
DOWJONESug_res2 <- (DOWJONESugfit@fit$residuals)^2
##Plotting the squared residuals and the estimated conditional variances
plot(DOWJONESug_res2, type ="c")
lines(DOWJONESug_var, col = "green")
## Model Forecasting
DOWJONESugfore <- ugarchforecast(DOWJONESugfit, n.ahead = 10)
DOWJONESugfore
DOWJONESug_f <- DOWJONESugfore@forecast$sigmaFor
plot(DOWJONESug_f, type = "c")
plot(DOWJONESugfore)
##GJRGARCH
DOWJONESgjrgarch.spec <- ugarchspec(variance.model = list(model="gjrGARCH",
garchOrder=c(1,1)),distribution.model = "norm")
DOWJONESgjrgarch.fit = ugarchfit(DOWJONESgjrgarch.spec, DOWJONES)
DOWJONESgjrgarch.fit
coef(DOWJONESgjrgarch.fit)
##GJRGARCH FORECASTING
NASDAQgjrgarch_Pred <- ugarchforecast(NASDAQgjrgarch.fit, n.ahead = 10, plot = TRUE)
plot(NASDAQgjrgarch_Pred)
##EGARCH
DOWJONES.egarch.spec = ugarchspec(variance.model=list(model="eGARCH",garchOrder=c(1,1)),
mean.model=list(armaOrder=c(0,0)))
DOWJONES.egarch.fit = ugarchfit(DOWJONES.egarch.spec, DOWJONES)
DOWJONES.egarch.fit
coef(DOWJONES.egarch.fit)
##EGARECH FORECASTING
DOWJONES.egarch_Pred <- ugarchforecast(DOWJONES.egarch.fit, n.ahead = 20, plot = TRUE)
plot(DOWJONES.egarch_Pred)
##Volatility Estimators for Datasetone: S & P 500
NGSPC <- GSPC
NGSPC
##Yang and Zhang volatility estimators
GSPCtest5 <- volatility(NGSPC, n = 30, calc = "yang.zhang", N = 256)
GSPCtest5
plot(GSPCtest5,
   main="Yang and Zhang Estimator: S & P 500",
   xlab="GSPCtest5")
##Volatility Estimators for Dataset two; NASDAQ
NIXIC <- IXIC
NIXIC
##Yang and Zhang volatility estimators
IXICtest5 <- volatility(NIXIC, n = 30, calc = "yang.zhang", N = 365)
IXICtest5
plot(IXICtest5,
   main="Yang and Zhang Estimator: NASDAQ",
```

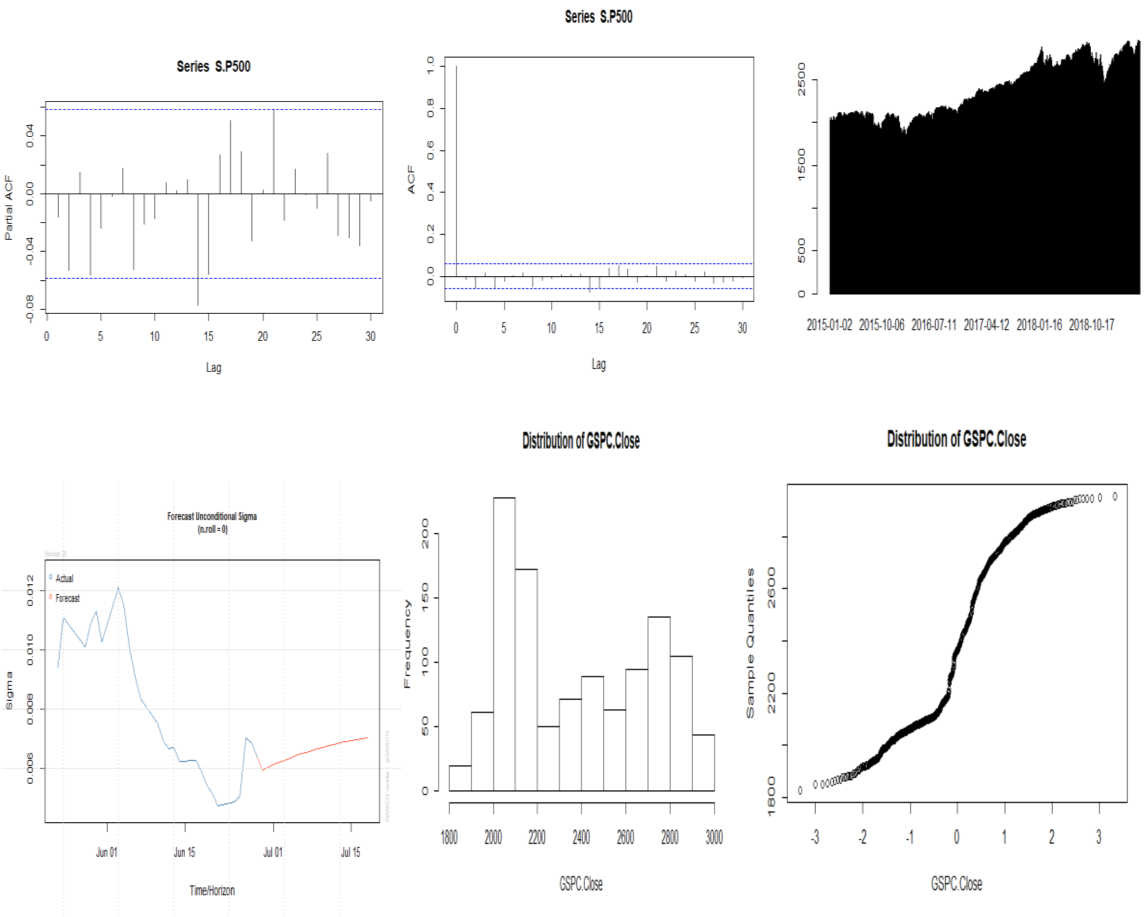
```
    xlab="IXICtest5")
##Volatility Estimators for Dataset three: DOWJONES
NDJI <- DJI
NDJI
##Yang and Zhang volatility estimators
DJItest5 <- volatility(NDJI, n = 30, calc = "yang.zhang", N = 365)
DJItest5
plot(DJItest5,
    main="Yang and Zhang Estimator: DOWJONES",
    xlab="DJItest5")
```

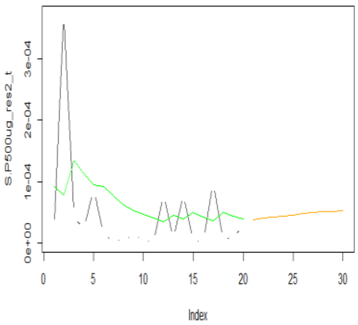
5 R Packages

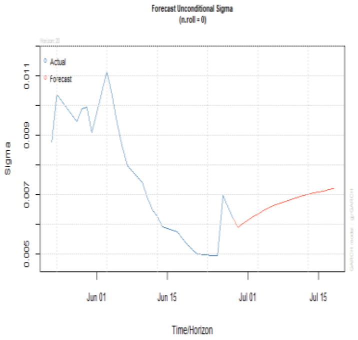
The following are the packages used for to enables data exploration, analysis, interpretation and evaluation: ("zoo"), ("xts"), (curl), (TTR), (quantmod), (lattice), (timeDate), (timeSeries), (parallel), (rugarch), (aTSA), (forecast), (ggplot2), (FinTS), (pdfetch), (rmgarch),(e1071), (MLmetrics), (tseries), (psycho) and (Metrics).

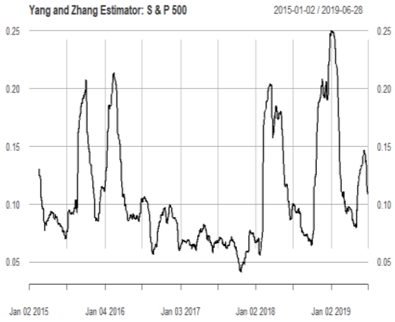
6 Graphical Illustration of Datasets

(a). S & P 500

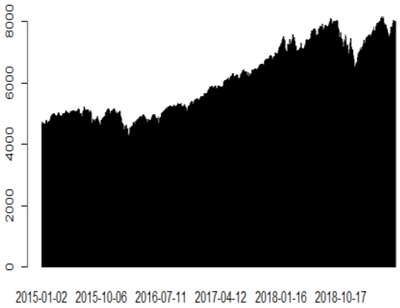


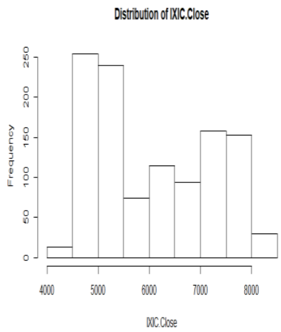


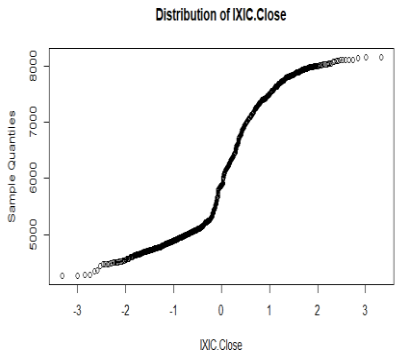


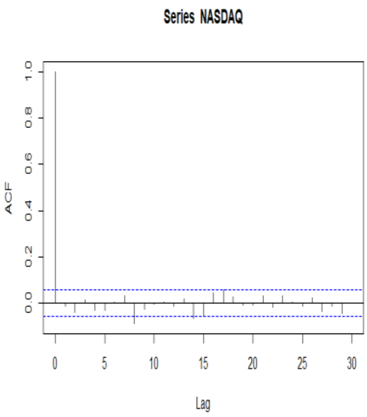


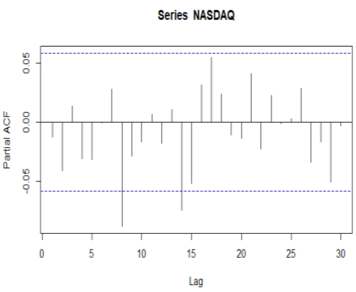
(b). NASDAQ Composites

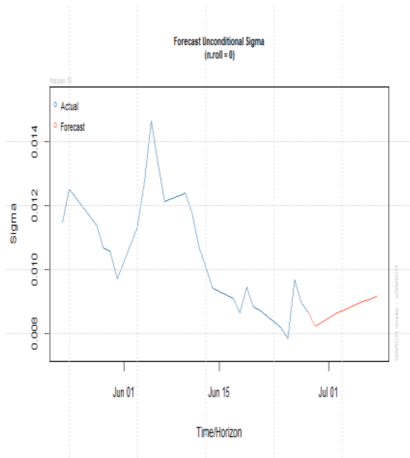


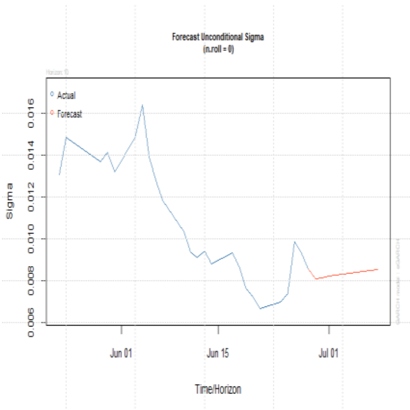


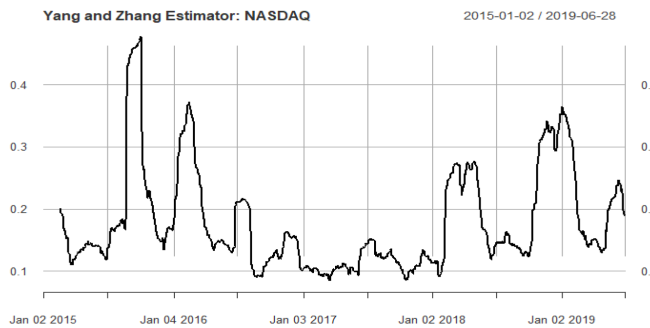












(c). Dow Jones Industrial Average