Installing Packages

```
install.packages("quantmod")
library(quantmod)
install.packages("devtools")
devtools::install_github("nipfpmf/eventstudies", ref="master")
install.packages("dplyr")
library(dplyr)
install.packages("rugarch")
library(rugarch)
install.packages("forecast")
library(forecast)
install.packages("xlsx")
library("xlsx")
install.packages("tseries")
library(tseries)
install.packages("timeSeries")
library(timeSeries)
install.packages("fGarch")
library(fGarch)
install.packages("prophet")
library(prophet)
```

EVENT STUDY

Step 1: Retrieving data using Quantmod for Japanese stocks

```
startJP <- as.Date("2018-05-28")
endJP <- as.Date("2018-07-28")

symbolBasketJP <- c("6113.T", "3865.T", "7211.T", "9735.T", "3289.T")

getSymbols(symbolBasketJP , src='yahoo', from = startJP, to = endJP)

Amada <- as.xts(`6113.T`)
names(Amada) <- c("Amada.Open"   ,  "Amada.High"   ,  "Amada.Low"   ,   "Amada.Close"
 , "Amada.Volume",  "Amada.Adjusted")

Hokuetsu <- as.xts(`3865.T`)
names(Hokuetsu) <- c("Hokuetsu.Open"   ,  "Hokuetsu.High"   ,  "Hokuetsu.Low"   ,   "Ho
kuetsu.Close"  ,  "Hokuetsu.Volume",  "Hokuetsu.Adjusted")

Mitsubishi <- as.xts(`7211.T`)
names(Mitsubishi) <- c("Mitsubishi.Open"   ,  "Mitsubishi.High"   ,  "Mitsubishi.Low"
,   "Mitsubishi.Close"  ,  "Mitsubishi.Volume",  "Mitsubishi.Adjusted")

Secom <- as.xts(`9735.T`)
names(Secom) <- c("Secom.Open"   ,  "Secom.High"   ,  "Secom.Low"   ,   "Secom.Close"
 ,  "Secom.Volume",  "Secom.Adjusted")

Tokyu <- as.xts(`3289.T`)
names(Tokyu) <- c("Tokyu.Open"   ,  "Tokyu.High"   ,  "Tokyu.Low"   ,   "Tokyu.Close"
 ,  "Tokyu.Volume",  "Tokyu.Adjusted")
```

Step 2: Calculate the daily stock return for the Japanese stocks

```
AmadaClose <- Amada[, "Amada.Close", drop = FALSE]
Amada_Return <- (diff(AmadaClose)/AmadaClose[-length(AmadaClose)])*100
names(Amada_Return) <- c("Amada")

HokuetsuClose <- Hokuetsu[, "Hokuetsu.Close", drop = FALSE]
Hokuetsu_Return <- (diff(HokuetsuClose)/HokuetsuClose[-length(HokuetsuClose)])*100
names(Hokuetsu_Return) <- c("Hokuetsun")

MitsubishiClose <- Mitsubishi[, "Mitsubishi.Close", drop = FALSE]
Mitsubishi_Return <- (diff(MitsubishiClose)/MitsubishiClose[-length(MitsubishiClose)])*
100
names(Mitsubishi_Return) <- c("Mitsubishi")

SecomClose <- Secom[, "Secom.Close", drop = FALSE]
Secom_Return <- (diff(SecomClose)/SecomClose[-length(SecomClose)])*100
names(Secom_Return) <- c("Secom")

TokyuClose <- Tokyu[, "Tokyu.Close", drop = FALSE]
Tokyu_Return <- (diff(TokyuClose)/TokyuClose[-length(TokyuClose)])*100
names(Tokyu_Return) <- c("Tokyu")
```

Step 3: Merging all the stock return datasets

```
stockPriceReturnJP<-merge(Amada_Return, Hokuetsu_Return, Mitsubishi_Return, Secom_Retur
n, Tokyu_Return)

View(stockPriceReturnJP)
str(stockPriceReturnJP)
```

Step 4: List the assets and the date of the flood

```
eventDateJP<-data.frame("name"=c("Amada","Hokuetsu","Mitsubishi", "Secom", "Tokyu"), "w
hen"=c("2018-06-28","2018-06-28","2018-06-28","2018-06-28","2018-06-28"))
```

Step 5: Convert the text

```
eventDateJP$name<-as.character(eventDateJP$name)
eventDateJP$when<-as.Date(eventDateJP$when)
```

Step 6: Event Study analysis for the Japanese stocks

```
esJP <- eventstudies::eventstudy(firm.returns = stockPriceReturnJP,
                                 event.list = eventDateJP,
                                 event.window = 5,
                                 type = "None",
                                 to.remap = TRUE,
                                 remap = "cumsum",
                                 inference = TRUE,
                                 inference.strategy = "bootstrap",
                                 )
```

Step 7: Plot event study

```
plot(esJP)
title(main="Japan Stock Cumulative Abnormal Return")
```

Step 8: Japan Flood Results

```
class(esJP)
str(esJP)
summary(esJP)
```

Step 9: Retrieving data using Quantmod for India stocks

```
startIN <- as.Date("2018-07-16")
endIN <- as.Date("2018-09-16")

symbolBasketIN <- c("TATASTEEL.BO", "TECHM.BO", "BAJAJ-AUTO.BO", "ICICIBANK.BO", "LT.B
O")

getSymbols(symbolBasketIN, src='yahoo', from = startIN, to = endIN)

Tata <- as.xts(`TATASTEEL.BO`)
names(Tata) <- c("Tata.Open"   ,  "Tata.High"   ,  "Tata.Low"   ,   "Tata.Close"  ,  "T
ata.Volume",  "Tata.Adjusted")

Mahindra <- as.xts(`TECHM.BO`)
names(Mahindra) <- c("Mahindra.Open"   ,  "Mahindra.High"   ,  "Mahindra.Low"   ,   "Ma
hindra.Close"  ,  "Mahindra.Volume",  "Mahindra.Adjusted")

Bajaj <- as.xts(`BAJAJ-AUTO.BO`)
names(Bajaj) <- c("Bajaj.Open"   ,  "Bajaj.High"   ,  "Bajaj.Low"   ,   "Bajaj.Close"
 ,  "Bajaj.Volume",  "Bajaj.Adjusted")

ICICI <- as.xts(`ICICIBANK.BO`)
names(ICICI) <- c("ICICI.Open"   ,  "ICICI.High"   ,  "ICICI.Low"   ,   "ICICI.Close"
 ,  "ICICI.Volume",  "ICICI.Adjusted")

LT <- as.xts(`LT.BO`)
names(LT) <- c("LT.Open"   ,  "LT.High"   ,  "LT.Low"   ,   "LT.Close"  ,  "LT.Volume",
"LT.Adjusted")
```

Step 10: Calculate the daily stock return for the Indian stocks

```
TataClose <- Tata[, "Tata.Close", drop = FALSE]
Tata_Return <- (diff(TataClose)/TataClose[-length(TataClose)])*100
names(Tata_Return) <- c("Tata")

MahindraClose <- Mahindra[, "Mahindra.Close", drop = FALSE]
Mahindra_Return <- (diff(MahindraClose)/MahindraClose[-length(MahindraClose)])*100
names(Mahindra_Return) <- c("Mahindran")

BajajClose <- Bajaj[, "Bajaj.Close", drop = FALSE]
Bajaj_Return <- (diff(BajajClose)/BajajClose[-length(BajajClose)])*100
names(Bajaj_Return) <- c("Bajaj")

ICICIClose <- ICICI[, "ICICI.Close", drop = FALSE]
ICICI_Return <- (diff(ICICIClose)/ICICIClose[-length(ICICIClose)])*100
names(ICICI_Return) <- c("ICICI")

LTClose <- LT[, "LT.Close", drop = FALSE]
LT_Return <- (diff(LTClose)/LTClose[-length(LTClose)])*100
names(LT_Return) <- c("LT")
```

Step 11: Merging all the stock return datasets

```
stockPriceReturnIN<-merge(Tata_Return, Mahindra_Return, Bajaj_Return, ICICI_Return, LT_
Return)

View(stockPriceReturnIN)
str(stockPriceReturnIN)
```

Step 12: List the assets and the date of the flood

```
eventDateIN<-data.frame("name"=c("Tata","Mahindra","Bajaj", "ICICI", "LT"), "when"=c("2
018-08-16","2018-08-16","2018-08-16","2018-08-16","2018-08-16"))
```

Step 13: Convert the text

```
eventDateIN$name<-as.character(eventDateIN$name)
eventDateIN$when<-as.Date(eventDateIN$when)
```

Step 14: Event Study analysis for the Indian stocks

```
esIN <- eventstudies::eventstudy(firm.returns = stockPriceReturnIN,
                                 event.list = eventDateIN,
                                 event.window = 5,
                                 type = "None",
                                 to.remap = TRUE,
                                 remap = "cumsum",
                                 inference = TRUE,
                                 inference.strategy = "bootstrap",
                                 )
```

Step 15: Plot event study

```
plot(esIN)
title(main="India Stock Cumulative Abnormal Return")
```

Step 16: India Flood Results

```
class(esIN)
str(esIN)
summary(esIN)
```

PRICE FORECAST

Step 1: Recollect Closing Price for Japanese Stocks

```
AmadaClose
HokuetsuClose
MitsubishiClose
SecomClose
TokyuClose
```

Step 2: Calculating the weighted average price of the 5 Japanese stocks

```
ForecastAVGJP <- AmadaClose^2*(1/(AmadaClose+HokuetsuClose+MitsubishiClose+SecomClose+T
okyuClose))
+HokuetsuClose^2*(1/(AmadaClose+HokuetsuClose+MitsubishiClose+SecomClose+TokyuClose))
+MitsubishiClose^2*(1/(AmadaClose+HokuetsuClose+MitsubishiClose+SecomClose+TokyuClose))
+SecomClose^2*(1/(AmadaClose+HokuetsuClose+MitsubishiClose+SecomClose+TokyuClose))
+TokyuClose^2*(1/(AmadaClose+HokuetsuClose+MitsubishiClose+SecomClose+TokyuClose))
```

Step 3: Plot the weighted average price of the 5 Japanese stocks

```
plot(ForecastAVGJP)
```

ARIMA

Step 4: Model Fit

```
print(adf.test(ForecastAVGJP))
modelfitJP <- auto.arima(ForecastAVGJP, lambda = "auto")
```

Step 5: Box test for lag=2

```
Box.test(modelfitJP$residuals, lag= 2, type="Ljung-Box")
```

Step 6: Dividing the data into train and test, applying the model

```
N = length(ForecastAVGJP)
trainJP =ForecastAVGJP [1:18,]
testJP  =ForecastAVGJP [(19):N,]
trainarimafitJP <- auto.arima(trainJP, lambda = "auto")
predlenJP=length(testJP)
trainarimafitJP <- forecast(trainarimafitJP, h=predlenJP)
```

Step 7: Plotting mean predicted values vs real data

```
meanvaluesJP <- as.vector(trainarimafitJP$mean)
preciosJP <- as.vector(testJP$ForecastAVGJP)
plot(meanvaluesJP, type= "l", col= "red")
lines(preciosJP, type = "l")
```

Step 8: Dataset forecasting for the next 30 days

```
price_forecastJP <- forecast(modelfitJP, h=30)
```

Step 9: Dataset forecast lower first 5 values

```
plot(price_forecastJP)

head(price_forecastJP$mean)

head(price_forecastJP$lower)

head(price_forecastJP$upper)
```

Feed Forward Neural network

Step 10: Hidden layers creation

```
alphaJP <- 1.5^(-10)
hnJP <- length(ForecastAVGJP)/(alphaJP*(length(ForecastAVGJP)+30))
```

Step 11: Fitting nnetar

```
lambdaJP <- BoxCox.lambda(ForecastAVGJP)
dnn_predJP <- nnetar(ForecastAVGJP, size= hnJP, lambda = lambdaJP)
dnn_forecastJP <- forecast(dnn_predJP, h= 30)
```

Step 12: Plot Fitting nnetar

```
plot(dnn_forecastJP)
```

Step 13: Recollect Closing Price for Indian Stocks

```
TataClose
MahindraClose
BajajClose
ICICIClose
LTClose
```

Step 14: Calculating the weighted average price of the 5 Japanese stocks

```
ForecastAVGIN <- TataClose^2*(1/(TataClose+MahindraClose+BajajClose+ICICIClose+LTClos
e))
+MahindraClose^2*(1/(TataClose+MahindraClose+BajajClose+ICICIClose+LTClose))
+BajajClose^2*(1/(TataClose+MahindraClose+BajajClose+ICICIClose+LTClose))
+ICICIClose^2*(1/(TataClose+MahindraClose+BajajClose+ICICIClose+LTClose))
+LTClose^2*(1/(TataClose+MahindraClose+BajajClose+ICICIClose+LTClose))
```

Step 15: Plot the weighted average price of the 5 Indian stocks

```
plot(ForecastAVGIN)
```

ARIMA

Step 16: Model Fit

```
print(adf.test(ForecastAVGIN))
modelfitIN <- auto.arima(ForecastAVGIN, lambda = "auto")
```

Step 17: Box test for lag=2

```
Box.test(modelfitIN$residuals, lag= 2, type="Ljung-Box")
```

Step 18: Dividing the data into train and test, applying the model

```
N = length(ForecastAVGIN)
trainIN =ForecastAVGIN [1:18,]
testIN  =ForecastAVGIN [(19):N,]
trainarimafitIN <- auto.arima(trainIN, lambda = "auto")
predlenIN=length(test)
trainarimafitIN <- forecast(trainarimafitIN, h=predlenIN)
```

Step 19: Plotting mean predicted values vs real data

```
meanvaluesIN <- as.vector(trainarimafitIN$mean)
preciosIN <- as.vector(testIN$ForecastAVGIN)
plot(meanvaluesIN, type= "l", col= "red")
lines(preciosIN, type = "l")
```

Step 20: Dataset forecasting for the next 30 days

```
price_forecastIN <- forecast(modelfitIN, h=30)
```

Step 21: Dataset forecast lower first 5 values

```
plot(price_forecastIN)

head(price_forecastIN$mean)

head(price_forecastIN$lower)

head(price_forecastIN$upper)
```

Feed Forward Neural network

Step 22: Hidden layers creation

```
alphaIN <- 1.5^(-10)
hnIN <- length(ForecastAVGIN)/(alphaIN*(length(ForecastAVGIN)+30))
```

Step 23: Fitting nnetar

```
lambdaIN <- BoxCox.lambda(ForecastAVGJP)
dnn_predIN <- nnetar(ForecastAVGJP, size= hnIN, lambda = lambdaIN)
dnn_forecastIN <- forecast(dnn_predIN, h= 30)
```

Step 24: Plot Fitting nnetar

```
plot(dnn_forecastIN)
```