initial-stock-analysis.R

ziyad

2020-07-05

library(prophet)

## Loading required package: Rcpp

## Loading required package: rlang

library(quantmod)

## Loading required package: xts

## Loading required package: zoo

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

## Loading required package: TTR

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Version 0.4-0 included new data defaults. See ?getSymbols.

library(forecast)  
library(xts)  
library(tseries)  
library(timeSeries)

## Loading required package: timeDate

##   
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':  
##   
## time<-

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:timeSeries':  
##   
## filter, lag

## The following objects are masked from 'package:xts':  
##   
## first, last

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(fGarch)

## Loading required package: fBasics

##   
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':  
##   
## volatility

start <- as.Date("2010-05-27")  
end <- as.Date("2020-05-27")  
  
getSymbols("AAPL",src = "yahoo",from = start, to = end)

## 'getSymbols' currently uses auto.assign=TRUE by default, but will  
## use auto.assign=FALSE in 0.5-0. You will still be able to use  
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")  
## and getOption("getSymbols.auto.assign") will still be checked for  
## alternate defaults.  
##   
## This message is shown once per session and may be disabled by setting   
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "AAPL"

class(AAPL)

## [1] "xts" "zoo"

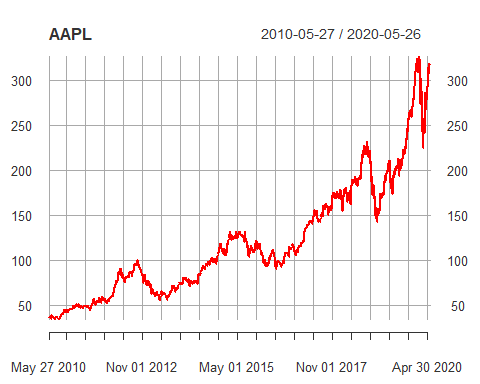
head(AAPL)

## AAPL.Open AAPL.High AAPL.Low AAPL.Close AAPL.Volume AAPL.Adjusted  
## 2010-05-27 35.80000 36.27000 35.58714 36.19286 166570600 31.33206  
## 2010-05-28 37.05571 37.05714 36.19286 36.69714 203903700 31.76862  
## 2010-06-01 37.09857 37.99143 36.99429 37.26143 219118200 32.25713  
## 2010-06-02 37.79143 37.82857 37.19000 37.70714 172137000 32.64297  
## 2010-06-03 37.88286 37.93572 37.20143 37.58857 162526700 32.54033  
## 2010-06-04 36.88714 37.41429 36.37571 36.56572 189576100 31.65483

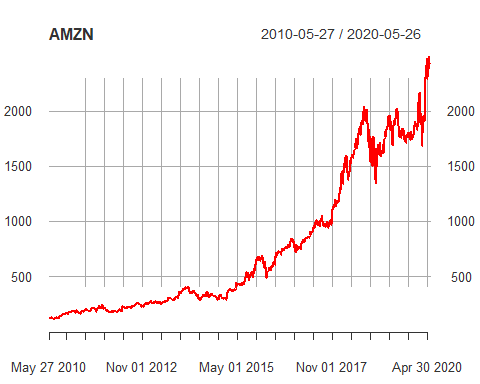
getSymbols(c("MSFT", "GOOG", "AMZN","NDAQ"), src = "yahoo", from = start, to = end)

## [1] "MSFT" "GOOG" "AMZN" "NDAQ"

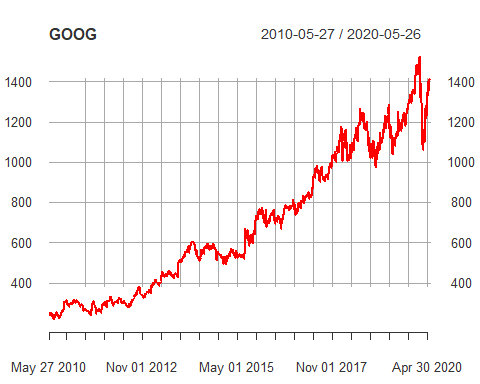
plot(AAPL[, "AAPL.Close"], main = "AAPL", col = "red" )



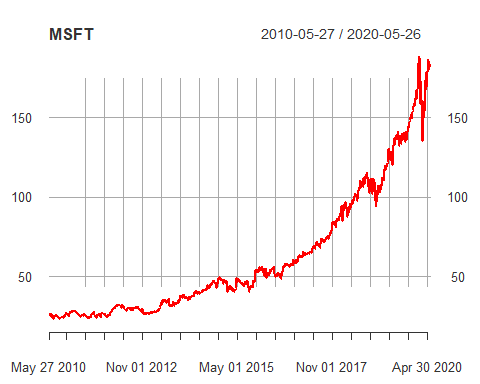
plot(AMZN[, "AMZN.Close"], main = "AMZN", col = "red")



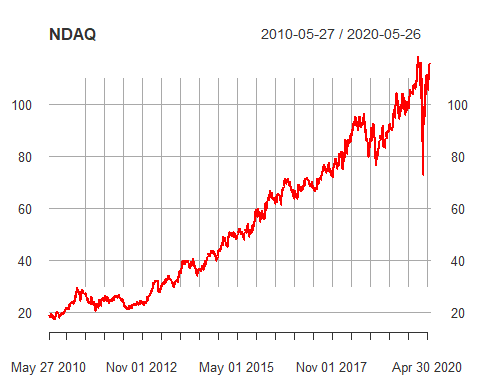
plot(GOOG[, "GOOG.Close"], main = "GOOG", col = "red")



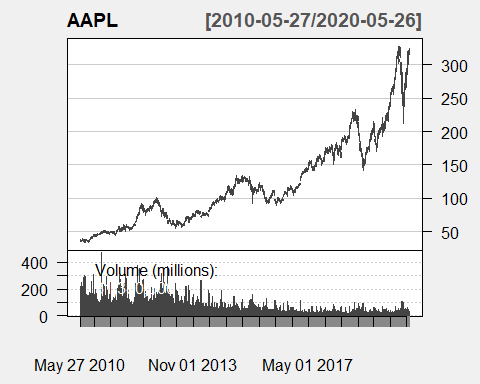
plot(MSFT[, "MSFT.Close"], main = "MSFT", col = "red")



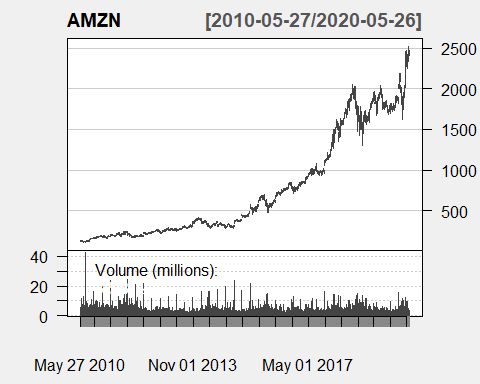
plot(NDAQ[, "NDAQ.Close"], main = "NDAQ", col = "red")



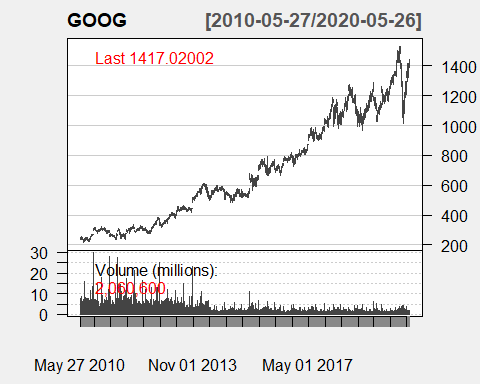
candleChart(AAPL, up.col = "red", dn.col = "white", theme = "white")



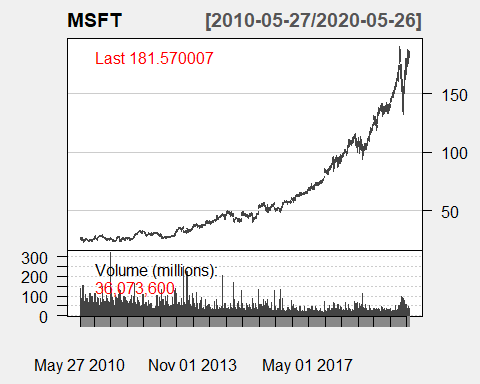
candleChart(AMZN, up.col = "red", dn.col = "white", theme = "white")



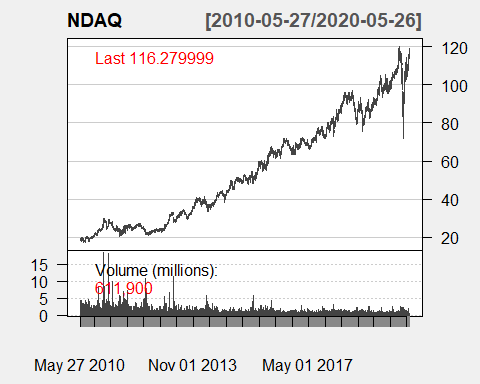
candleChart(GOOG, up.col = "black", dn.col = "red", theme = "white")



candleChart(MSFT, up.col = "black", dn.col = "red", theme = "white")



candleChart(NDAQ, up.col = "black", dn.col = "red", theme = "white")



summary(AAPL)

## Index AAPL.Open AAPL.High AAPL.Low   
## Min. :2010-05-27 Min. : 34.01 Min. : 34.66 Min. : 33.65   
## 1st Qu.:2012-11-25 1st Qu.: 72.23 1st Qu.: 72.79 1st Qu.: 71.60   
## Median :2015-05-27 Median :106.64 Median :107.61 Median :105.55   
## Mean :2015-05-26 Mean :120.60 Mean :121.77 Mean :119.47   
## 3rd Qu.:2017-11-21 3rd Qu.:160.41 3rd Qu.:162.13 3rd Qu.:159.07   
## Max. :2020-05-26 Max. :324.74 Max. :327.85 Max. :323.35   
## AAPL.Close AAPL.Volume AAPL.Adjusted   
## Min. : 34.28 Min. : 11362000 Min. : 29.67   
## 1st Qu.: 72.25 1st Qu.: 30370325 1st Qu.: 63.91   
## Median :106.49 Median : 51044200 Median : 99.07   
## Mean :120.67 Mean : 70013248 Mean :113.99   
## 3rd Qu.:160.59 3rd Qu.: 95160450 3rd Qu.:154.79   
## Max. :327.20 Max. :470249500 Max. :326.32

summary(AMZN)

## Index AMZN.Open AMZN.High AMZN.Low   
## Min. :2010-05-27 Min. : 105.9 Min. : 111.3 Min. : 105.8   
## 1st Qu.:2012-11-25 1st Qu.: 253.1 1st Qu.: 255.5 1st Qu.: 250.1   
## Median :2015-05-27 Median : 429.7 Median : 432.8 Median : 426.2   
## Mean :2015-05-26 Mean : 753.5 Mean : 761.1 Mean : 745.1   
## 3rd Qu.:2017-11-21 3rd Qu.:1134.1 3rd Qu.:1144.0 3rd Qu.:1128.0   
## Max. :2020-05-26 Max. :2500.0 Max. :2525.4 Max. :2467.3   
## AMZN.Close AMZN.Volume AMZN.Adjusted   
## Min. : 108.6 Min. : 881300 Min. : 108.6   
## 1st Qu.: 252.5 1st Qu.: 2773975 1st Qu.: 252.5   
## Median : 429.5 Median : 3739050 Median : 429.5   
## Mean : 753.5 Mean : 4382792 Mean : 753.5   
## 3rd Qu.:1137.8 3rd Qu.: 5168100 3rd Qu.:1137.8   
## Max. :2497.9 Max. :42421100 Max. :2497.9

summary(GOOG)

## Index GOOG.Open GOOG.High GOOG.Low   
## Min. :2010-05-27 Min. : 218.3 Min. : 220.3 Min. : 216.0   
## 1st Qu.:2012-11-25 1st Qu.: 355.9 1st Qu.: 358.0 1st Qu.: 352.2   
## Median :2015-05-27 Median : 588.0 Median : 592.5 Median : 582.8   
## Mean :2015-05-26 Mean : 685.9 Mean : 692.0 Mean : 679.8   
## 3rd Qu.:2017-11-21 3rd Qu.:1016.6 3rd Qu.:1029.9 3rd Qu.:1001.4   
## Max. :2020-05-26 Max. :1525.1 Max. :1532.1 Max. :1521.4   
## GOOG.Close GOOG.Volume GOOG.Adjusted   
## Min. : 217.2 Min. : 7900 Min. : 217.2   
## 1st Qu.: 354.2 1st Qu.: 1403400 1st Qu.: 354.2   
## Median : 587.8 Median : 2167350 Median : 587.8   
## Mean : 686.1 Mean : 3087344 Mean : 686.1   
## 3rd Qu.:1017.6 3rd Qu.: 4181050 3rd Qu.:1017.6   
## Max. :1526.7 Max. :29760600 Max. :1526.7

summary(MSFT)

## Index MSFT.Open MSFT.High MSFT.Low   
## Min. :2010-05-27 Min. : 23.09 Min. : 23.32 Min. : 22.73   
## 1st Qu.:2012-11-25 1st Qu.: 30.33 1st Qu.: 30.64 1st Qu.: 30.15   
## Median :2015-05-27 Median : 46.66 Median : 47.05 Median : 46.27   
## Mean :2015-05-26 Mean : 61.83 Mean : 62.39 Mean : 61.24   
## 3rd Qu.:2017-11-21 3rd Qu.: 83.72 3rd Qu.: 84.38 3rd Qu.: 83.19   
## Max. :2020-05-26 Max. :190.65 Max. :190.70 Max. :186.47   
## MSFT.Close MSFT.Volume MSFT.Adjusted   
## Min. : 23.01 Min. : 7425600 Min. : 18.18   
## 1st Qu.: 30.39 1st Qu.: 24135600 1st Qu.: 25.26   
## Median : 46.70 Median : 33598250 Median : 41.96   
## Mean : 61.85 Mean : 39225748 Mean : 57.95   
## 3rd Qu.: 83.90 3rd Qu.: 49103425 3rd Qu.: 80.56   
## Max. :188.70 Max. :319317900 Max. :187.66

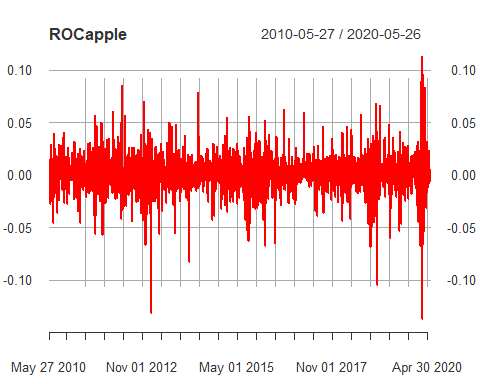
summary(NDAQ)

## Index NDAQ.Open NDAQ.High NDAQ.Low   
## Min. :2010-05-27 Min. : 17.46 Min. : 17.75 Min. : 17.18   
## 1st Qu.:2012-11-25 1st Qu.: 27.25 1st Qu.: 27.58 1st Qu.: 27.01   
## Median :2015-05-27 Median : 50.59 Median : 50.94 Median : 50.22   
## Mean :2015-05-26 Mean : 54.80 Mean : 55.32 Mean : 54.29   
## 3rd Qu.:2017-11-21 3rd Qu.: 77.05 3rd Qu.: 77.39 3rd Qu.: 76.22   
## Max. :2020-05-26 Max. :119.24 Max. :120.23 Max. :118.46   
## NDAQ.Close NDAQ.Volume NDAQ.Adjusted   
## Min. : 17.30 Min. : 217900 Min. : 14.90   
## 1st Qu.: 27.31 1st Qu.: 749375 1st Qu.: 23.60   
## Median : 50.59 Median : 1053400 Median : 45.84   
## Mean : 54.83 Mean : 1342232 Mean : 51.28   
## 3rd Qu.: 76.81 3rd Qu.: 1595925 3rd Qu.: 73.02   
## Max. :118.67 Max. :18263200 Max. :117.63

#computing rates of change  
  
#Apple  
  
ROCapple <- ROC(AAPL[, "AAPL.Close"], n=1,)  
head(ROCapple)

## AAPL.Close  
## 2010-05-27 NA  
## 2010-05-28 0.013837181  
## 2010-06-01 0.015259785  
## 2010-06-02 0.011890803  
## 2010-06-03 -0.003149505  
## 2010-06-04 -0.027588935

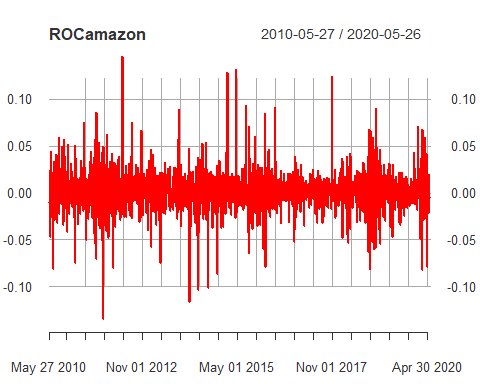
plot(ROCapple, col="red")



#amazon  
  
ROCamazon <- ROC(AMZN[, "AMZN.Close"], n=1,)  
head(ROCamazon)

## AMZN.Close  
## 2010-05-27 NA  
## 2010-05-28 -0.009835089  
## 2010-06-01 -0.017853317  
## 2010-06-02 0.024605529  
## 2010-06-03 0.019210981  
## 2010-06-04 -0.047637506

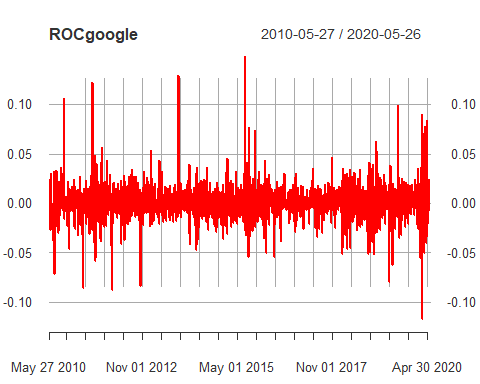
plot(ROCamazon, col="red")



#google  
  
ROCgoogle <- ROC(GOOG[, "GOOG.Close"], n=1,)  
head(ROCgoogle)

## GOOG.Close  
## 2010-05-27 NA  
## 2010-05-28 -0.009896747  
## 2010-06-01 -0.006735512  
## 2010-06-02 0.022547942  
## 2010-06-03 0.024486438  
## 2010-06-04 -0.013701030

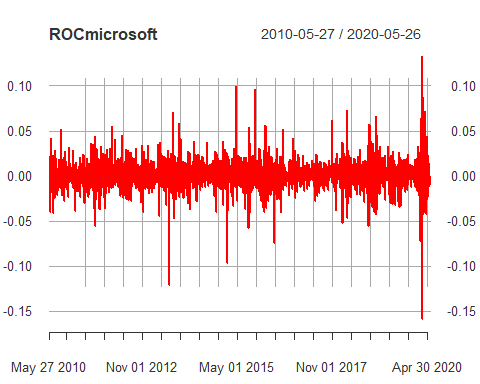
plot(ROCgoogle, col="red")



#micosoft  
  
ROCmicrosoft <- ROC(MSFT[, "MSFT.Close"], n=1,)  
head(ROCmicrosoft)

## MSFT.Close  
## 2010-05-27 NA  
## 2010-05-28 -0.007722085  
## 2010-06-01 0.003482302  
## 2010-06-02 0.021777366  
## 2010-06-03 0.015004107  
## 2010-06-04 -0.040651370

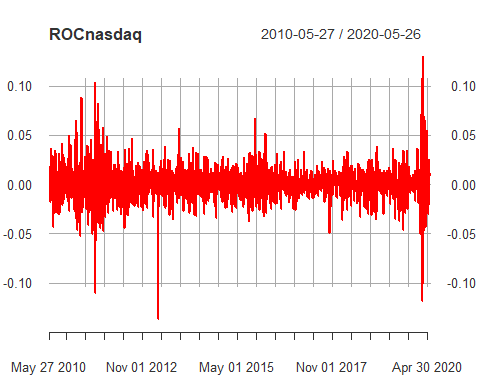
plot(ROCmicrosoft, col="red")



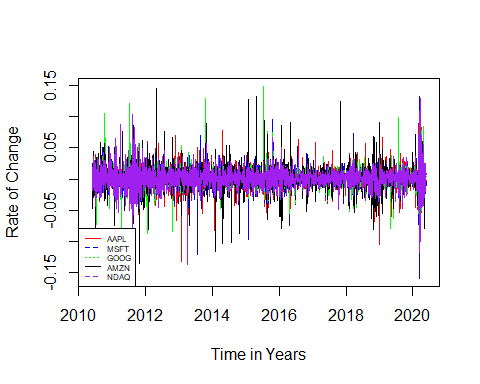
#NASDAQ  
  
ROCnasdaq <- ROC(NDAQ[, "NDAQ.Close"], n=1,)  
head(ROCnasdaq)

## NDAQ.Close  
## 2010-05-27 NA  
## 2010-05-28 -0.018124163  
## 2010-06-01 -0.010816817  
## 2010-06-02 0.018319532  
## 2010-06-03 0.002133334  
## 2010-06-04 -0.018280079

plot(ROCnasdaq, col="red")



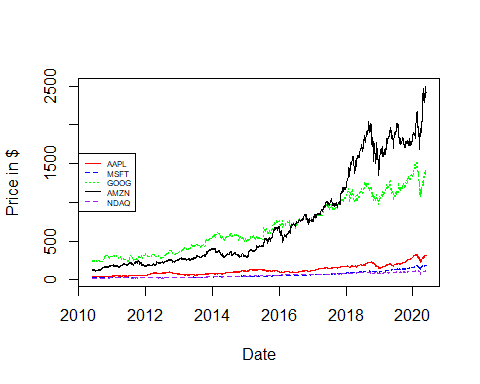
#combining all roc plots  
  
roc <- as.xts(data.frame(AAPL = ROCapple, MSFT = ROCmicrosoft, GOOG = ROCgoogle, AMZN = ROCamazon, NDAQ = ROCnasdaq))  
plot(as.zoo(roc), screens = 1,col = c("red","blue","green","black","Purple"), ylab = "Rate of Change", xlab = "Time in Years")  
legend("bottomleft", c("AAPL", "MSFT", "GOOG", "AMZN", "NDAQ"), col=c("red","blue","green","black","Purple"), lty = 1:3, cex = 0.5)



# Create an xts object (xts is loaded with quantmod) that contains closing  
# prices for AAPL, MSFT, and GOOG  
stocks <- as.xts(data.frame(AAPL = AAPL[, "AAPL.Close"], MSFT = MSFT[, "MSFT.Close"],   
 GOOG = GOOG[, "GOOG.Close"], AMZN = AMZN[, "AMZN.Close"], NDAQ = NDAQ[, "NDAQ.Close"]))  
head(stocks)

## AAPL.Close MSFT.Close GOOG.Close AMZN.Close NDAQ.Close  
## 2010-05-27 36.19286 26.00 244.3143 126.70 18.93  
## 2010-05-28 36.69714 25.80 241.9083 125.46 18.59  
## 2010-06-01 37.26143 25.89 240.2844 123.24 18.39  
## 2010-06-02 37.70714 26.46 245.7638 126.31 18.73  
## 2010-06-03 37.58857 26.86 251.8560 128.76 18.77  
## 2010-06-04 36.56572 25.79 248.4288 122.77 18.43

#Saving into CSV  
write.csv(stocks, file = "stocks.csv", row.names = FALSE)  
setwd("C:/Users/ziyad/Desktop/Data Analytics capstone")  
  
# Create a plot showing all series as lines; must use as.zoo to use the zoo  
# method for plot, which allows for multiple series to be plotted on same  
# plot  
  
plot(as.zoo(stocks), screens = 1, lty = 1:3,col = c("red","blue","green","black","Purple"), xlab = "Date", ylab = "Price in $")  
legend("left", c("AAPL", "MSFT", "GOOG", "AMZN", "NDAQ"), col=c("red","blue","green","black","Purple"), lty = 1:3, cex = 0.5)



#get pipe operator  
  
if (!require("magrittr")) {  
 install.packages("magrittr")  
 library(magrittr)  
}

## Loading required package: magrittr

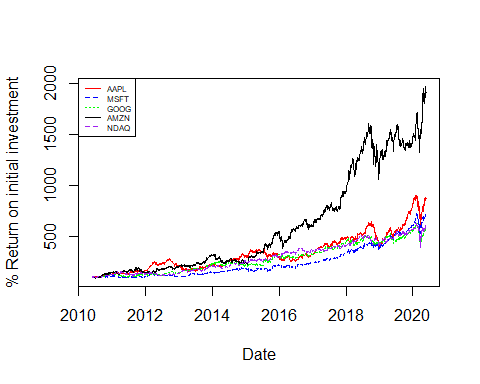
##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:rlang':  
##   
## set\_names

nominal\_return = apply(stocks, 1, function(x) {x / stocks[1,]\*100}) %>%   
 t %>% as.xts  
  
head(nominal\_return)

## AAPL.Close MSFT.Close GOOG.Close AMZN.Close NDAQ.Close  
## 2010-05-27 100.0000 100.00000 100.00000 100.00000 100.00000  
## 2010-05-28 101.3933 99.23077 99.01521 99.02131 98.20391  
## 2010-06-01 102.9524 99.57692 98.35053 97.26914 97.14738  
## 2010-06-02 104.1839 101.76923 100.59332 99.69219 98.94348  
## 2010-06-03 103.8563 103.30770 103.08690 101.62589 99.15478  
## 2010-06-04 101.0302 99.19231 101.68413 96.89818 97.35869

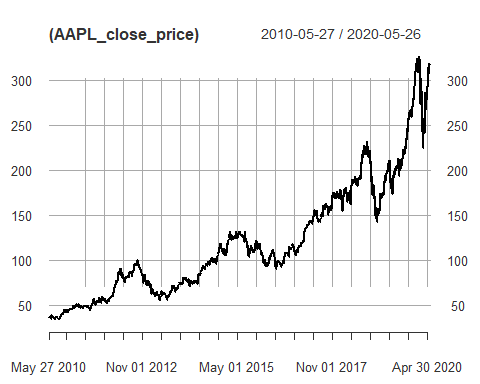
plot(as.zoo(nominal\_return), screens = 1, lty = 1:3, xlab = "Date", ylab = "% Return on initial investment", col = c("red","blue","green","black","purple"))  
legend("topleft", c("AAPL", "MSFT", "GOOG","AMZN","NDAQ"),col = c("red","blue","green","black","purple"), lty = 1:3, cex = 0.5)



#Testing out ARIMA models with closing value on Y axis and Date on x axis  
  
#1) Apple  
  
class(AAPL)

## [1] "xts" "zoo"

AAPL\_close\_price <- (AAPL[,4])  
plot((AAPL\_close\_price))



class(AAPL\_close\_price)

## [1] "xts" "zoo"

par(mfrow = c(1,1))  
print(adf.test(AAPL\_close\_price)) #P-value > 0.05 indicates non stationary data

##   
## Augmented Dickey-Fuller Test  
##   
## data: AAPL\_close\_price  
## Dickey-Fuller = -1.4497, Lag order = 13, p-value = 0.8113  
## alternative hypothesis: stationary

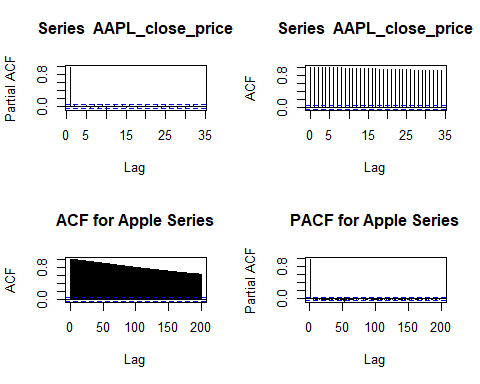
auto.arima(AAPL\_close\_price, seasonal = FALSE)

## Series: AAPL\_close\_price   
## ARIMA(1,1,0) with drift   
##   
## Coefficients:  
## ar1 drift  
## -0.1774 0.1115  
## s.e. 0.0196 0.0456  
##   
## sigma^2 estimated as 7.252: log likelihood=-6059.04  
## AIC=12124.08 AICc=12124.09 BIC=12141.57

par(mfrow = c(2,2))  
pacf(AAPL\_close\_price)  
acf(AAPL\_close\_price)  
  
#Translating Raw price into logarithmic form to obtain returns  
  
lnapple = log(AAPL\_close\_price)  
print(lnapple[1:10])

## AAPL.Close  
## 2010-05-27 3.588862  
## 2010-05-28 3.602699  
## 2010-06-01 3.617959  
## 2010-06-02 3.629850  
## 2010-06-03 3.626700  
## 2010-06-04 3.599111  
## 2010-06-07 3.579304  
## 2010-06-08 3.572867  
## 2010-06-09 3.547974  
## 2010-06-10 3.577589

#Checking for stationarity  
  
acf(lnapple, lag.max = 200, main = "ACF for Apple Series")  
pacf(lnapple, lag.max = 200, main = "PACF for Apple Series")



print(adf.test(lnapple)) #P-value > 0.05 indicates non stationary data

##   
## Augmented Dickey-Fuller Test  
##   
## data: lnapple  
## Dickey-Fuller = -2.6024, Lag order = 13, p-value = 0.3233  
## alternative hypothesis: stationary

#Differencing the data to turn into stationary data  
  
difflnapple = diff(lnapple,)  
head(difflnapple)

## AAPL.Close  
## 2010-05-27 NA  
## 2010-05-28 0.013837181  
## 2010-06-01 0.015259785  
## 2010-06-02 0.011890803  
## 2010-06-03 -0.003149505  
## 2010-06-04 -0.027588935

adf.test(lnapple)

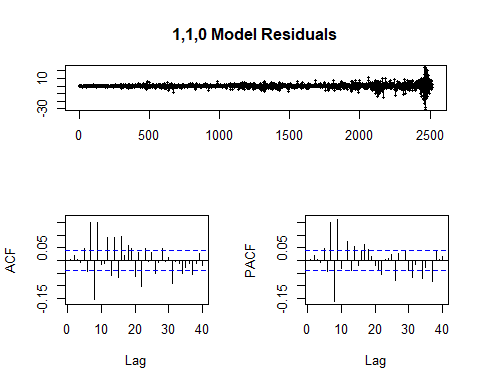
##   
## Augmented Dickey-Fuller Test  
##   
## data: lnapple  
## Dickey-Fuller = -2.6024, Lag order = 13, p-value = 0.3233  
## alternative hypothesis: stationary

adf.test(difflnapple[2:1259,]) # p-value < 0.05 indication stationary data. Model fitting to continue

## Warning in adf.test(difflnapple[2:1259, ]): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: difflnapple[2:1259, ]  
## Dickey-Fuller = -12.088, Lag order = 10, p-value = 0.01  
## alternative hypothesis: stationary

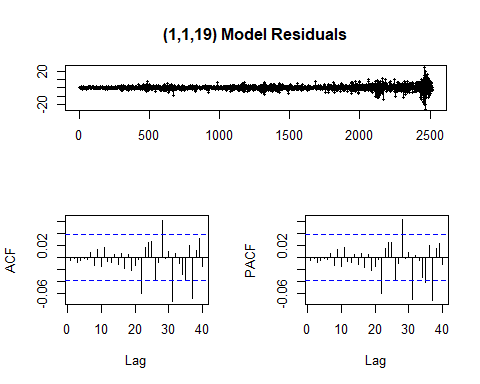
#Trying different fits  
  
fitA = auto.arima(AAPL\_close\_price, seasonal = FALSE)  
tsdisplay(residuals(fitA), lag.max = 40, main = '1,1,0 Model Residuals')



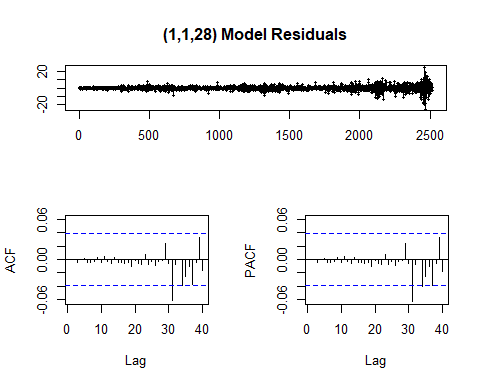
auto.arima(AAPL\_close\_price, seasonal = FALSE)

## Series: AAPL\_close\_price   
## ARIMA(1,1,0) with drift   
##   
## Coefficients:  
## ar1 drift  
## -0.1774 0.1115  
## s.e. 0.0196 0.0456  
##   
## sigma^2 estimated as 7.252: log likelihood=-6059.04  
## AIC=12124.08 AICc=12124.09 BIC=12141.57

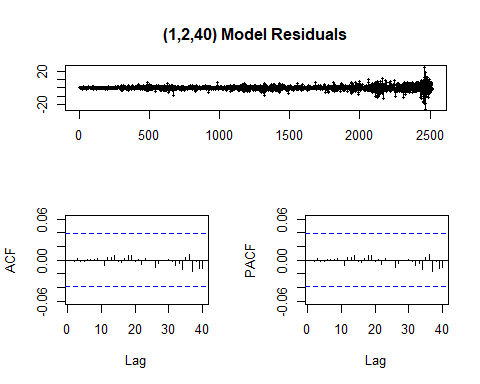
fitB = arima(AAPL\_close\_price, order = c(1,1,19))  
tsdisplay(residuals(fitB), lag.max = 40, main = '(1,1,19) Model Residuals')



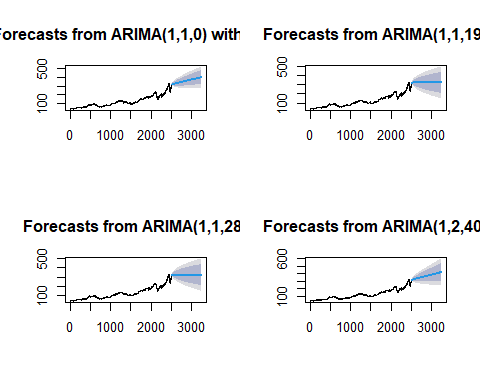
fitC = arima(AAPL\_close\_price, order = c(1,1,28))  
tsdisplay(residuals(fitC), lag.max = 40, main = '(1,1,28) Model Residuals')



fitD = arima(AAPL\_close\_price, order = c(1,2,40))  
tsdisplay(residuals(fitD), lag.max = 40, main = '(1,2,40) Model Residuals')



par(mfrow = c(2,2))  
term <- 730  
fcast1 <- forecast(fitA, h = term)  
plot(fcast1)  
fcast2 <- forecast(fitB, h = term)  
plot(fcast2)  
fcast3 <- forecast(fitC, h = term)  
plot(fcast3)  
fcast4 <- forecast(fitD, h = term)  
plot(fcast4)



print(fcast4)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2517 314.8170 311.5904 318.0437 309.8824 319.7517  
## 2518 316.1997 311.8598 320.5397 309.5624 322.8371  
## 2519 312.1398 306.9324 317.3472 304.1758 320.1038  
## 2520 315.6240 309.6557 321.5924 306.4962 324.7518  
## 2521 313.7070 307.0645 320.3495 303.5481 323.8658  
## 2522 317.1991 309.9095 324.4887 306.0507 328.3475  
## 2523 318.1448 310.2975 325.9920 306.1434 330.1461  
## 2524 317.2591 308.7282 325.7900 304.2123 330.3059  
## 2525 317.8926 308.9084 326.8769 304.1524 331.6328  
## 2526 317.5074 307.9299 327.0848 302.8599 332.1548  
## 2527 320.0622 309.9451 330.1794 304.5894 335.5350  
## 2528 319.6926 309.0822 330.3029 303.4654 335.9197  
## 2529 320.4477 309.2747 331.6206 303.3601 337.5352  
## 2530 319.2804 307.6432 330.9176 301.4828 337.0780  
## 2531 319.5633 307.4029 331.7237 300.9656 338.1610  
## 2532 319.7945 307.1930 332.3961 300.5221 339.0670  
## 2533 319.1800 306.0656 332.2944 299.1232 339.2368  
## 2534 320.6022 306.9642 334.2402 299.7447 341.4597  
## 2535 318.8652 304.6985 333.0318 297.1992 340.5311  
## 2536 319.1417 304.4132 333.8703 296.6164 341.6671  
## 2537 318.6959 303.4921 333.8997 295.4437 341.9481  
## 2538 319.3175 303.6094 335.0255 295.2940 343.3409  
## 2539 319.0925 302.9729 335.2121 294.4397 343.7453

#2) Amazon  
  
class(AMZN)

## [1] "xts" "zoo"

AMZN\_close\_price <- (AMZN[,4])  
plot((AMZN\_close\_price))  
class(AMZN\_close\_price)

## [1] "xts" "zoo"

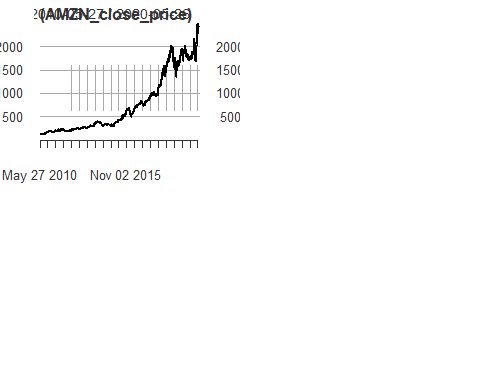
print(adf.test(AMZN\_close\_price)) #P-value > 0.05 indicates non stationary data

##   
## Augmented Dickey-Fuller Test  
##   
## data: AMZN\_close\_price  
## Dickey-Fuller = -1.004, Lag order = 13, p-value = 0.9381  
## alternative hypothesis: stationary

auto.arima(AMZN\_close\_price, seasonal = FALSE)

## Series: AMZN\_close\_price   
## ARIMA(0,1,1) with drift   
##   
## Coefficients:  
## ma1 drift  
## -0.0739 0.9127  
## s.e. 0.0193 0.3541  
##   
## sigma^2 estimated as 368: log likelihood=-10996.96  
## AIC=21999.93 AICc=21999.94 BIC=22017.42

par(mfrow = c(2,2))

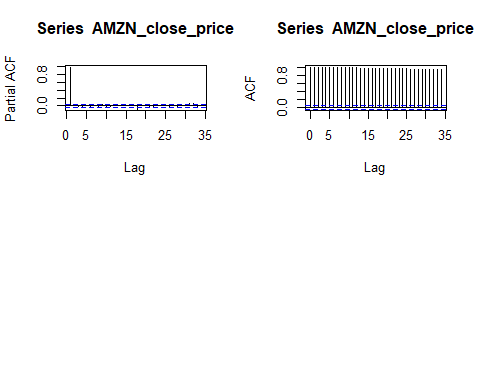


pacf(AMZN\_close\_price)  
acf(AMZN\_close\_price)

#Translating Raw price into logarithmic form to obtain returns  
  
lnamzn = log(AMZN\_close\_price)  
head(lnamzn)

## AMZN.Close  
## 2010-05-27 4.841822  
## 2010-05-28 4.831987  
## 2010-06-01 4.814134  
## 2010-06-02 4.838739  
## 2010-06-03 4.857950  
## 2010-06-04 4.810313

#Checking for stationarity  
  
par(mfrow = c(2,2))



acf(lnamzn, lag.max = 200, main = "ACF for Amazon Series")  
pacf(lnamzn, lag.max = 200, main = "PACF for Amazon Series")  
print(adf.test(lnamzn)) #P-value > 0.05 indicates non stationary data

##   
## Augmented Dickey-Fuller Test  
##   
## data: lnamzn  
## Dickey-Fuller = -2.7387, Lag order = 13, p-value = 0.2656  
## alternative hypothesis: stationary

#Differencing the data to turn into stationary data  
  
difflnamzn = diff(lnamzn,)  
head(difflnamzn)

## AMZN.Close  
## 2010-05-27 NA  
## 2010-05-28 -0.009835089  
## 2010-06-01 -0.017853317  
## 2010-06-02 0.024605529  
## 2010-06-03 0.019210981  
## 2010-06-04 -0.047637506

adf.test(lnamzn)

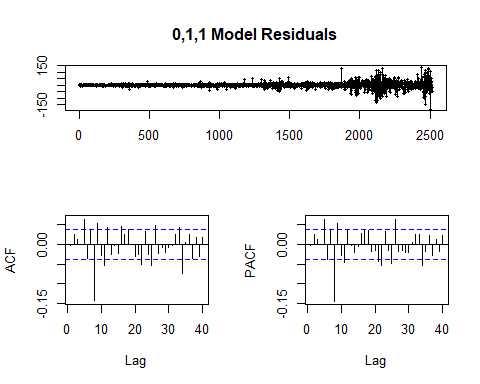
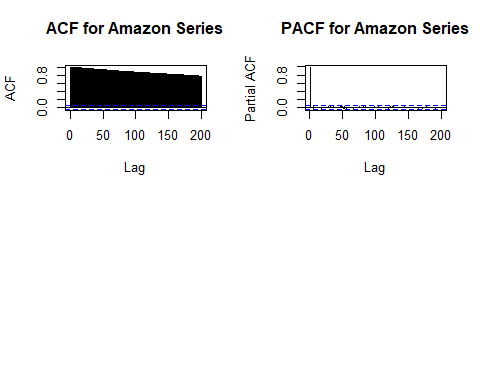
##   
## Augmented Dickey-Fuller Test  
##   
## data: lnamzn  
## Dickey-Fuller = -2.7387, Lag order = 13, p-value = 0.2656  
## alternative hypothesis: stationary

adf.test(difflnamzn[2:1259,]) # p-value < 0.05 indication stationary data. Model fitting to continue

## Warning in adf.test(difflnamzn[2:1259, ]): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: difflnamzn[2:1259, ]  
## Dickey-Fuller = -11.195, Lag order = 10, p-value = 0.01  
## alternative hypothesis: stationary

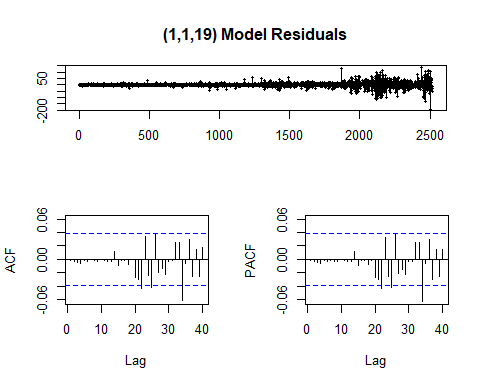
#Trying different fits  
  
fitA = auto.arima(AMZN\_close\_price, seasonal = FALSE)  
tsdisplay(residuals(fitA), lag.max = 40, main = '0,1,1 Model Residuals')



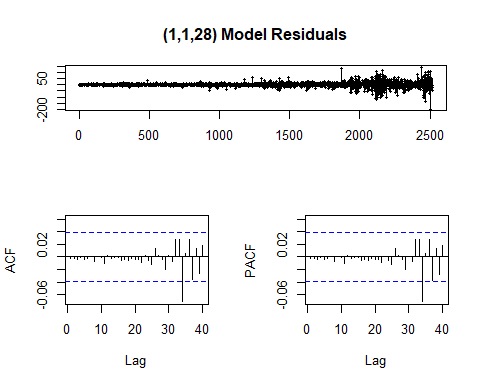
auto.arima(AMZN\_close\_price, seasonal = FALSE)

## Series: AMZN\_close\_price   
## ARIMA(0,1,1) with drift   
##   
## Coefficients:  
## ma1 drift  
## -0.0739 0.9127  
## s.e. 0.0193 0.3541  
##   
## sigma^2 estimated as 368: log likelihood=-10996.96  
## AIC=21999.93 AICc=21999.94 BIC=22017.42

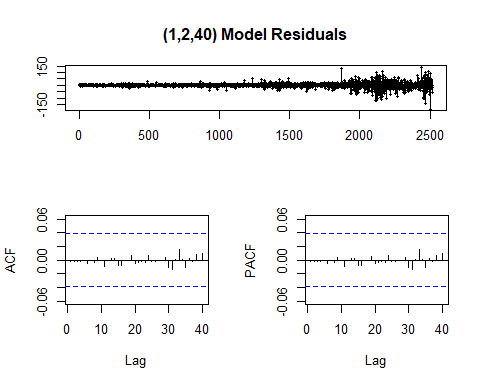
fitB = arima(AMZN\_close\_price, order = c(1,1,19))  
tsdisplay(residuals(fitB), lag.max = 40, main = '(1,1,19) Model Residuals')



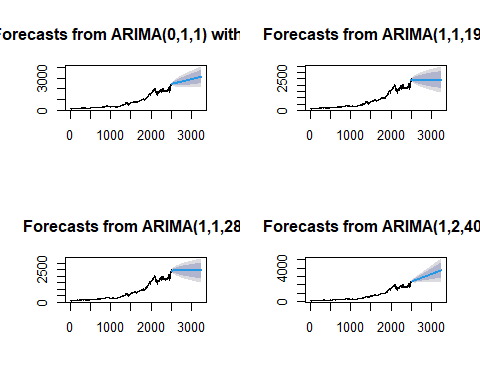
fitC = arima(AMZN\_close\_price, order = c(1,1,28))  
tsdisplay(residuals(fitC), lag.max = 40, main = '(1,1,28) Model Residuals')



fitD = arima(AMZN\_close\_price, order = c(1,2,40))  
tsdisplay(residuals(fitD), lag.max = 40, main = '(1,2,40) Model Residuals')



par(mfrow = c(2,2))  
term <- 730  
fcast1 <- forecast(fitA, h = term)  
plot(fcast1)  
fcast2 <- forecast(fitB, h = term)  
plot(fcast2)  
fcast3 <- forecast(fitC, h = term)  
plot(fcast3)  
fcast4 <- forecast(fitD, h = term)  
plot(fcast4)



print(fcast4)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2517 2424.683 2400.863 2448.502 2388.253 2461.112  
## 2518 2422.232 2389.448 2455.016 2372.093 2472.371  
## 2519 2422.756 2382.685 2462.827 2361.473 2484.040  
## 2520 2413.059 2366.683 2459.435 2342.133 2483.985  
## 2521 2413.796 2361.760 2465.833 2334.213 2493.380  
## 2522 2432.720 2374.957 2490.483 2344.379 2521.061  
## 2523 2405.989 2343.274 2468.704 2310.075 2501.904  
## 2524 2412.001 2344.379 2479.624 2308.582 2515.421  
## 2525 2423.832 2352.708 2494.955 2315.058 2532.605  
## 2526 2414.770 2339.907 2489.633 2300.278 2529.263  
## 2527 2415.986 2337.785 2494.187 2296.388 2535.584  
## 2528 2412.952 2331.775 2494.128 2288.803 2537.100

#3) Google  
  
class(GOOG)

## [1] "xts" "zoo"

GOOG\_close\_price <- (GOOG[,4])  
plot((GOOG\_close\_price))  
class(GOOG\_close\_price)

## [1] "xts" "zoo"

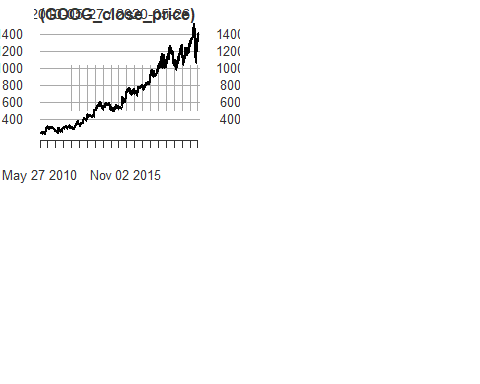
print(adf.test(GOOG\_close\_price)) #P-value < 0.05 indicates stationary data

##   
## Augmented Dickey-Fuller Test  
##   
## data: GOOG\_close\_price  
## Dickey-Fuller = -3.6475, Lag order = 13, p-value = 0.02809  
## alternative hypothesis: stationary

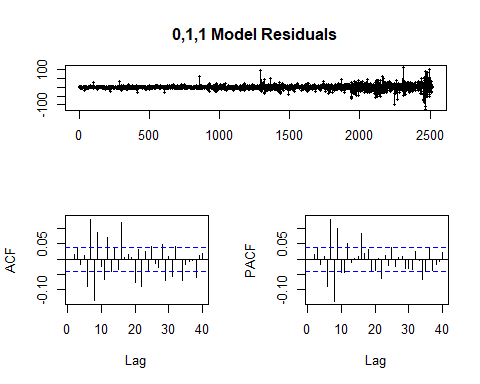
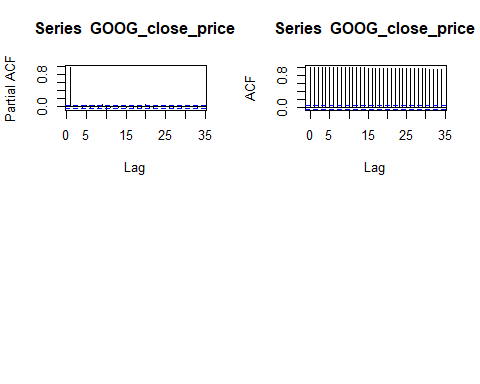
auto.arima(GOOG\_close\_price, seasonal = FALSE)

## Series: GOOG\_close\_price   
## ARIMA(0,1,1) with drift   
##   
## Coefficients:  
## ma1 drift  
## -0.1210 0.4663  
## s.e. 0.0194 0.2285  
##   
## sigma^2 estimated as 170: log likelihood=-10025.93  
## AIC=20057.87 AICc=20057.88 BIC=20075.36

par(mfrow = c(2,2))



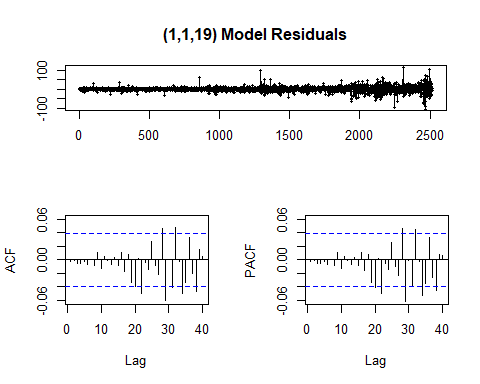
pacf(GOOG\_close\_price)  
acf(GOOG\_close\_price)  
  
  
#Trying different fits  
  
fitA = auto.arima(GOOG\_close\_price, seasonal = FALSE)  
tsdisplay(residuals(fitA), lag.max = 40, main = '0,1,1 Model Residuals')



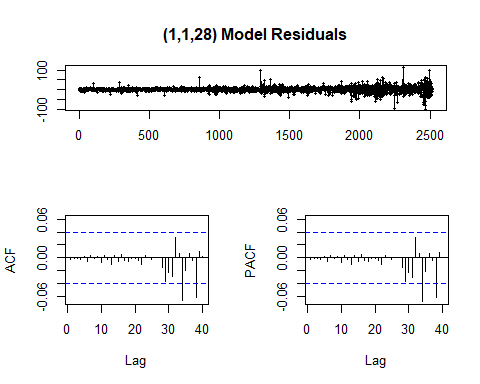
auto.arima(GOOG\_close\_price, seasonal = FALSE)

## Series: GOOG\_close\_price   
## ARIMA(0,1,1) with drift   
##   
## Coefficients:  
## ma1 drift  
## -0.1210 0.4663  
## s.e. 0.0194 0.2285  
##   
## sigma^2 estimated as 170: log likelihood=-10025.93  
## AIC=20057.87 AICc=20057.88 BIC=20075.36

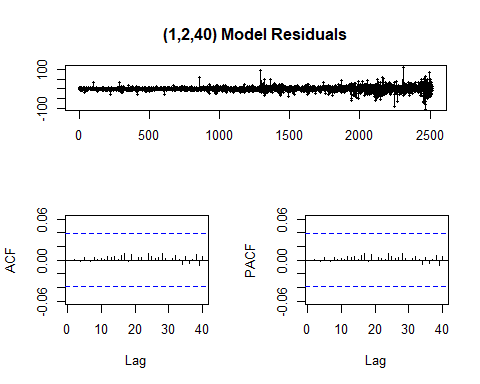
fitB = arima(GOOG\_close\_price, order = c(1,1,19))  
tsdisplay(residuals(fitB), lag.max = 40, main = '(1,1,19) Model Residuals')



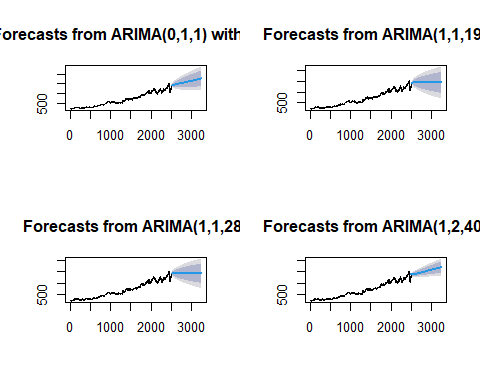
fitC = arima(GOOG\_close\_price, order = c(1,1,28))  
tsdisplay(residuals(fitC), lag.max = 40, main = '(1,1,28) Model Residuals')



fitD = arima(GOOG\_close\_price, order = c(1,2,40))  
tsdisplay(residuals(fitD), lag.max = 40, main = '(1,2,40) Model Residuals')



par(mfrow = c(2,2))  
term <- 730  
fcast1 <- forecast(fitA, h = term)  
plot(fcast1)  
fcast2 <- forecast(fitB, h = term)  
plot(fcast2)  
fcast3 <- forecast(fitC, h = term)  
plot(fcast3)  
fcast4 <- forecast(fitD, h = term)  
plot(fcast4)



print(fcast4)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2517 1405.057 1389.173 1420.941 1380.765 1429.349  
## 2518 1410.976 1389.308 1432.644 1377.837 1444.115  
## 2519 1407.811 1381.806 1433.816 1368.040 1447.582  
## 2520 1409.917 1379.843 1439.991 1363.923 1455.910  
## 2521 1401.559 1368.076 1435.043 1350.350 1452.768  
## 2522 1406.975 1370.413 1443.536 1351.059 1462.891  
## 2523 1410.135 1371.159 1449.110 1350.527 1469.743  
## 2524 1396.256 1354.327 1438.185 1332.132 1460.381  
## 2525 1397.105 1353.129 1441.082 1329.850 1464.361  
## 2526 1399.631 1353.407 1445.855 1328.937 1470.325  
## 2527 1403.348 1355.017 1451.679 1329.432 1477.264  
## 2528 1392.548 1342.583 1442.514 1316.132 1468.965  
## 2529 1401.252 1349.433 1453.071 1322.002 1480.503

#4) Microsoft  
  
class(MSFT)

## [1] "xts" "zoo"

MSFT\_close\_price <- (MSFT[,4])  
plot((MSFT\_close\_price))  
class(MSFT\_close\_price)

## [1] "xts" "zoo"

print(adf.test(MSFT\_close\_price)) #P-value > 0.05 indicates non stationary data

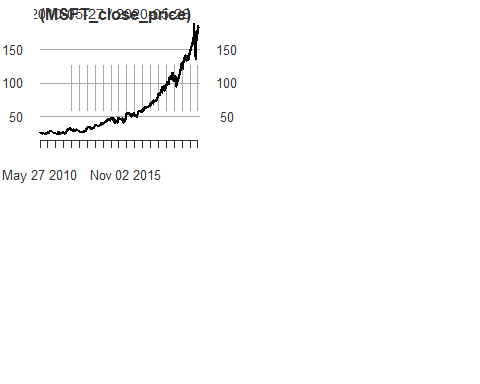
## Warning in adf.test(MSFT\_close\_price): p-value greater than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: MSFT\_close\_price  
## Dickey-Fuller = -0.18412, Lag order = 13, p-value = 0.99  
## alternative hypothesis: stationary

auto.arima(MSFT\_close\_price, seasonal = FALSE)

## Series: MSFT\_close\_price   
## ARIMA(5,2,0)   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5  
## -1.1345 -0.8642 -0.5125 -0.2833 -0.0585  
## s.e. 0.0199 0.0296 0.0327 0.0296 0.0199  
##   
## sigma^2 estimated as 2.291: log likelihood=-4607.53  
## AIC=9227.06 AICc=9227.1 BIC=9262.04

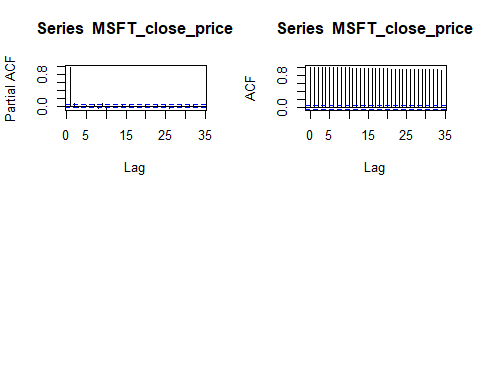
par(mfrow = c(2,2))



pacf(MSFT\_close\_price)  
acf(MSFT\_close\_price)  
  
#Translating Raw price into logarithmic form to obtain returns  
  
lnMSFT = log(MSFT\_close\_price)  
head(lnMSFT)

## MSFT.Close  
## 2010-05-27 3.258097  
## 2010-05-28 3.250374  
## 2010-06-01 3.253857  
## 2010-06-02 3.275634  
## 2010-06-03 3.290638  
## 2010-06-04 3.249987

#Checking for stationarity  
  
par(mfrow = c(2,2))



acf(lnMSFT, lag.max = 200, main = "ACF for Microsoft Series")  
pacf(lnMSFT, lag.max = 200, main = "PACF for Microsoft Series")  
print(adf.test(lnMSFT)) #P-value > 0.05 indicates non stationary data

##   
## Augmented Dickey-Fuller Test  
##   
## data: lnMSFT  
## Dickey-Fuller = -2.4475, Lag order = 13, p-value = 0.3888  
## alternative hypothesis: stationary

#Differencing the data to turn into stationary data  
  
difflnMSFT = diff(lnMSFT,)  
head(difflnMSFT)

## MSFT.Close  
## 2010-05-27 NA  
## 2010-05-28 -0.007722085  
## 2010-06-01 0.003482302  
## 2010-06-02 0.021777366  
## 2010-06-03 0.015004107  
## 2010-06-04 -0.040651370

adf.test(lnMSFT)

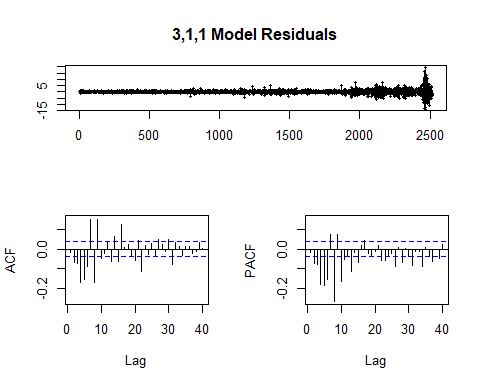
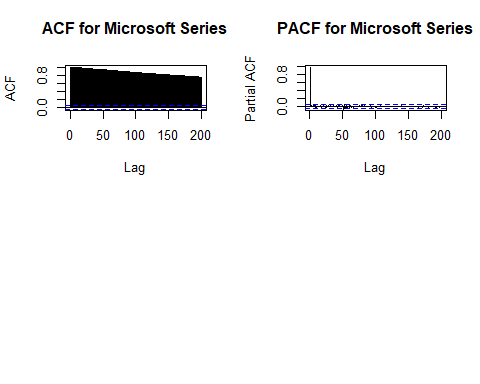
##   
## Augmented Dickey-Fuller Test  
##   
## data: lnMSFT  
## Dickey-Fuller = -2.4475, Lag order = 13, p-value = 0.3888  
## alternative hypothesis: stationary

adf.test(difflnMSFT[2:1259,]) # p-value < 0.05 indication stationary data. Model fitting to continue

## Warning in adf.test(difflnMSFT[2:1259, ]): p-value smaller than printed p-value

##   
## Augmented Dickey-Fuller Test  
##   
## data: difflnMSFT[2:1259, ]  
## Dickey-Fuller = -11.475, Lag order = 10, p-value = 0.01  
## alternative hypothesis: stationary

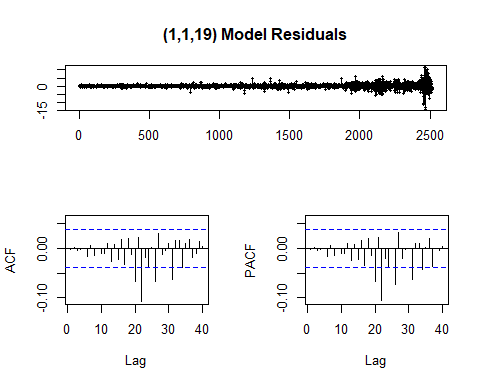
#Trying different fits  
  
fitA = auto.arima(MSFT\_close\_price, seasonal = FALSE)  
tsdisplay(residuals(fitA), lag.max = 40, main = '3,1,1 Model Residuals')



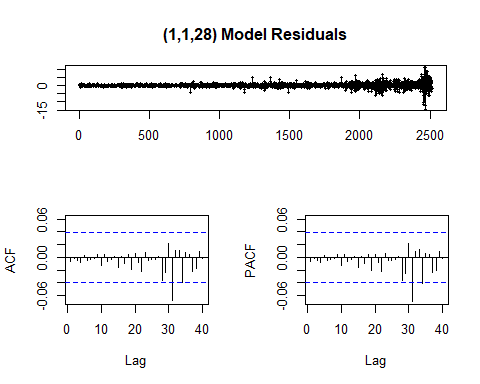
auto.arima(MSFT\_close\_price, seasonal = FALSE)

## Series: MSFT\_close\_price   
## ARIMA(5,2,0)   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ar5  
## -1.1345 -0.8642 -0.5125 -0.2833 -0.0585  
## s.e. 0.0199 0.0296 0.0327 0.0296 0.0199  
##   
## sigma^2 estimated as 2.291: log likelihood=-4607.53  
## AIC=9227.06 AICc=9227.1 BIC=9262.04

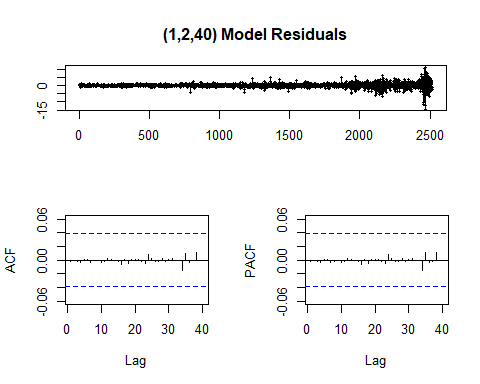
fitB = arima(MSFT\_close\_price, order = c(1,1,19))  
tsdisplay(residuals(fitB), lag.max = 40, main = '(1,1,19) Model Residuals')



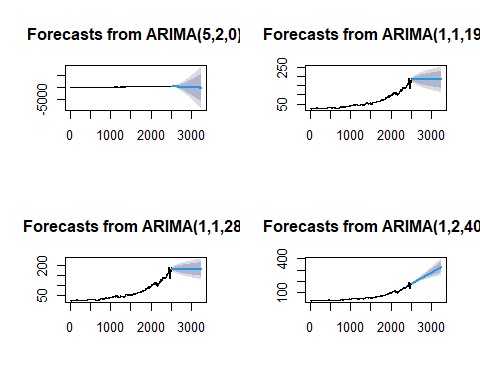
fitC = arima(MSFT\_close\_price, order = c(1,1,28))  
tsdisplay(residuals(fitC), lag.max = 40, main = '(1,1,28) Model Residuals')



fitD = arima(MSFT\_close\_price, order = c(1,2,40))  
tsdisplay(residuals(fitD), lag.max = 40, main = '(1,2,40) Model Residuals')



par(mfrow = c(2,2))  
term <- 730  
fcast1 <- forecast(fitA, h = term)  
plot(fcast1)  
fcast2 <- forecast(fitB, h = term)  
plot(fcast2)  
fcast3 <- forecast(fitC, h = term)  
plot(fcast3)  
fcast4 <- forecast(fitD, h = term)  
plot(fcast4)



print(fcast4)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2517 182.5733 180.9323 184.2143 180.0637 185.0830  
## 2518 182.5177 180.4626 184.5727 179.3748 185.6605  
## 2519 181.4292 179.0064 183.8519 177.7238 185.1345  
## 2520 183.0628 180.2421 185.8835 178.7489 187.3766  
## 2521 180.7688 177.6704 183.8671 176.0302 185.5073  
## 2522 182.8971 179.5412 186.2531 177.7646 188.0296  
## 2523 182.0705 178.5150 185.6260 176.6328 187.5082  
## 2524 180.9297 177.0957 184.7637 175.0662 186.7933  
## 2525 180.4250 176.4374 184.4126 174.3266 186.5234  
## 2526 180.5899 176.3670 184.8129 174.1315 187.0484  
## 2527 180.7540 176.3414 185.1666 174.0055 187.5025  
## 2528 180.6504 176.0687 185.2320 173.6433 187.6574

#5) Nasdaq  
  
class(NDAQ)

## [1] "xts" "zoo"

NDAQ\_close\_price <- (NDAQ[,4])  
plot((NDAQ\_close\_price))  
class(NDAQ\_close\_price)

## [1] "xts" "zoo"

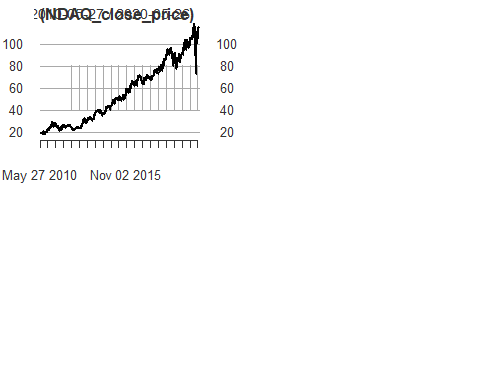
print(adf.test(NDAQ\_close\_price)) #P-value < 0.05 indicates stationary data

##   
## Augmented Dickey-Fuller Test  
##   
## data: NDAQ\_close\_price  
## Dickey-Fuller = -3.4999, Lag order = 13, p-value = 0.04229  
## alternative hypothesis: stationary

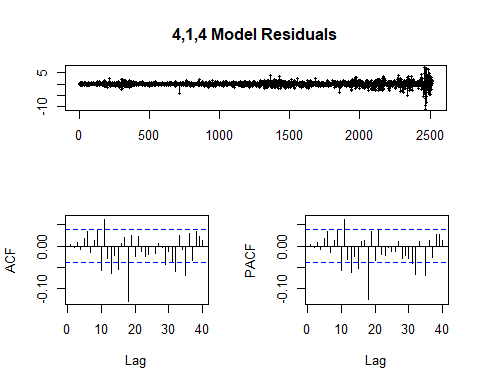
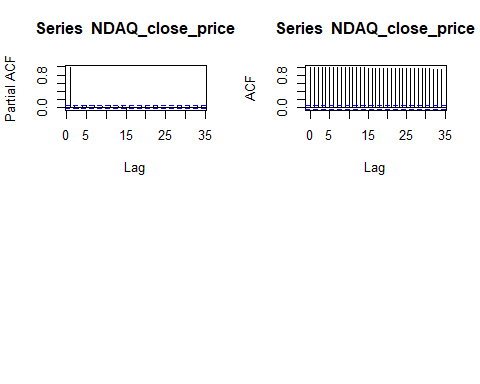
auto.arima(NDAQ\_close\_price, seasonal = FALSE)

## Series: NDAQ\_close\_price   
## ARIMA(4,1,4) with drift   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4  
## -0.4556 0.4208 -0.4570 -0.7842 0.3934 -0.3814 0.4492 0.6070  
## s.e. 0.0362 0.0422 0.0493 0.0393 0.0453 0.0516 0.0610 0.0491  
## drift  
## 0.0388  
## s.e. 0.0174  
##   
## sigma^2 estimated as 0.9233: log likelihood=-3464.02  
## AIC=6948.03 AICc=6948.12 BIC=7006.33

par(mfrow = c(2,2))



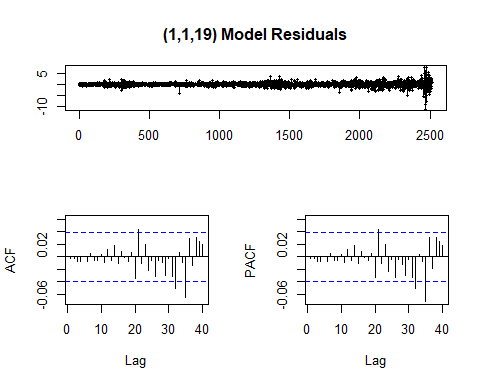
pacf(NDAQ\_close\_price)  
acf(NDAQ\_close\_price)  
  
  
#Trying different fits  
  
fitA = auto.arima(NDAQ\_close\_price, seasonal = FALSE)  
tsdisplay(residuals(fitA), lag.max = 40, main = '4,1,4 Model Residuals')



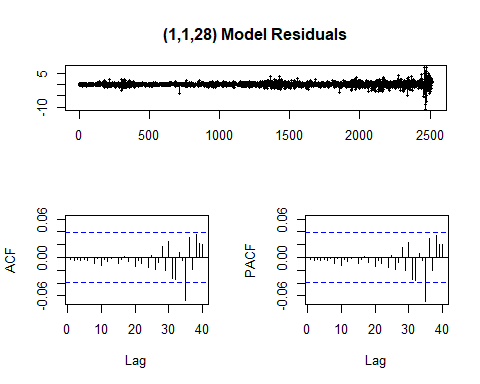
auto.arima(NDAQ\_close\_price, seasonal = FALSE)

## Series: NDAQ\_close\_price   
## ARIMA(4,1,4) with drift   
##   
## Coefficients:  
## ar1 ar2 ar3 ar4 ma1 ma2 ma3 ma4  
## -0.4556 0.4208 -0.4570 -0.7842 0.3934 -0.3814 0.4492 0.6070  
## s.e. 0.0362 0.0422 0.0493 0.0393 0.0453 0.0516 0.0610 0.0491  
## drift  
## 0.0388  
## s.e. 0.0174  
##   
## sigma^2 estimated as 0.9233: log likelihood=-3464.02  
## AIC=6948.03 AICc=6948.12 BIC=7006.33

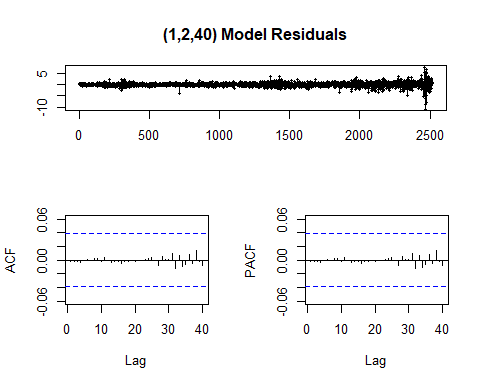
fitB = arima(NDAQ\_close\_price, order = c(1,1,19))  
tsdisplay(residuals(fitB), lag.max = 40, main = '(1,1,19) Model Residuals')



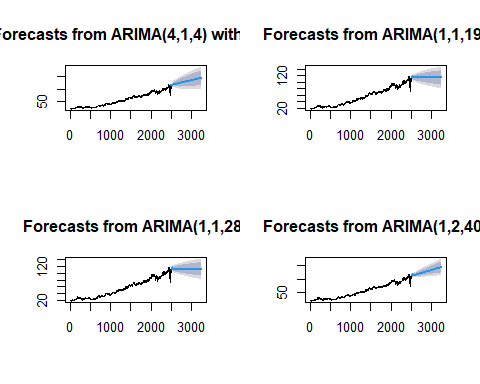
fitC = arima(NDAQ\_close\_price, order = c(1,1,28))  
tsdisplay(residuals(fitC), lag.max = 40, main = '(1,1,28) Model Residuals')



fitD = arima(NDAQ\_close\_price, order = c(1,2,40))  
tsdisplay(residuals(fitD), lag.max = 40, main = '(1,2,40) Model Residuals')



par(mfrow = c(2,2))  
term <- 730  
fcast1 <- forecast(fitA, h = term)  
plot(fcast1)  
fcast2 <- forecast(fitB, h = term)  
plot(fcast2)  
fcast3 <- forecast(fitC, h = term)  
plot(fcast3)  
fcast4 <- forecast(fitD, h = term)  
plot(fcast4)



print(fcast4)

## Point Forecast Lo 80 Hi 80 Lo 95 Hi 95  
## 2517 116.7115 115.5169 117.9060 114.8846 118.5383  
## 2518 117.2365 115.5894 118.8836 114.7175 119.7555  
## 2519 116.3650 114.3283 118.4017 113.2501 119.4799  
## 2520 116.2553 113.9312 118.5795 112.7009 119.8098  
## 2521 115.9227 113.3903 118.4552 112.0497 119.7958  
## 2522 115.9039 113.1646 118.6432 111.7144 120.0933  
## 2523 115.7409 112.8303 118.6516 111.2894 120.1924  
## 2524 115.6430 112.5155 118.7704 110.8599 120.4260  
## 2525 115.7996 112.4844 119.1147 110.7294 120.8697  
## 2526 115.3866 111.8444 118.9289 109.9693 120.8040  
## 2527 115.6976 111.9908 119.4043 110.0286 121.3666  
## 2528 114.6597 110.7861 118.5333 108.7355 120.5838