DSc Assignment 3

Vaibhav Mittal

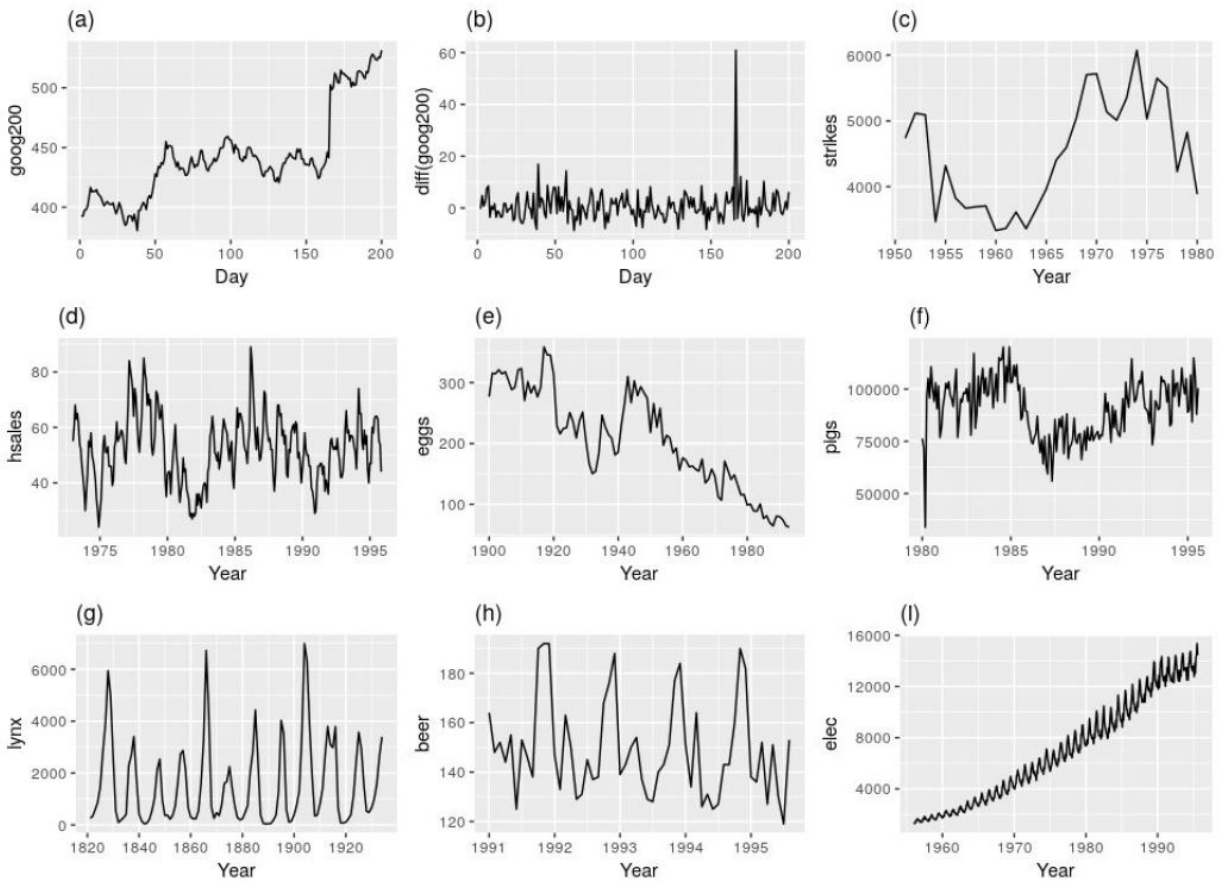
24 October 2019

Time Series Analysis

Submitted by Vaibhav Mittal, MT18242.

Question 1

```r
setwd("E:/College/Sem3/DSc/Assign3/")
```



Question:

a cannot be considered stationary since the mean is changing. b can be considered stationary iff outliers aren't considered. Because of some instances of changing covariances, b isn't stationary. c isn't stationary.

d isn't stationary. It has some oscillatory behaviour too. e isn't stationary due to changing mean. f isn't stationary too due to the same reasons.

g seems stationary albeit it has some outliers. h is a bit less stationary since it has more outliers. There are some covariance changes too. f is not at all stationary.

Question 2

```
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
```

```
## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo
```

```
## 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.
```

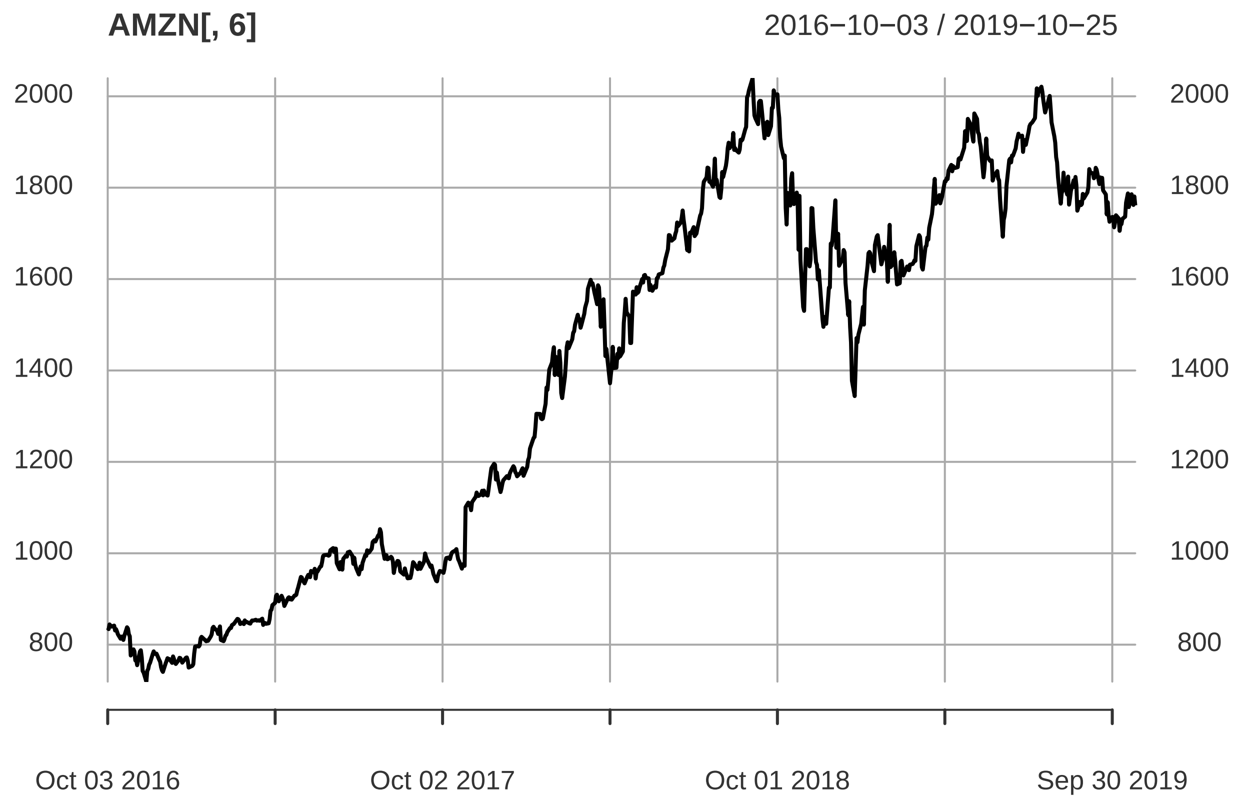
```
getSymbols(c('AMZN', 'MSFT', 'GOOG'), src = 'yahoo', from = '2016-10-01')
```

```
## '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.
```

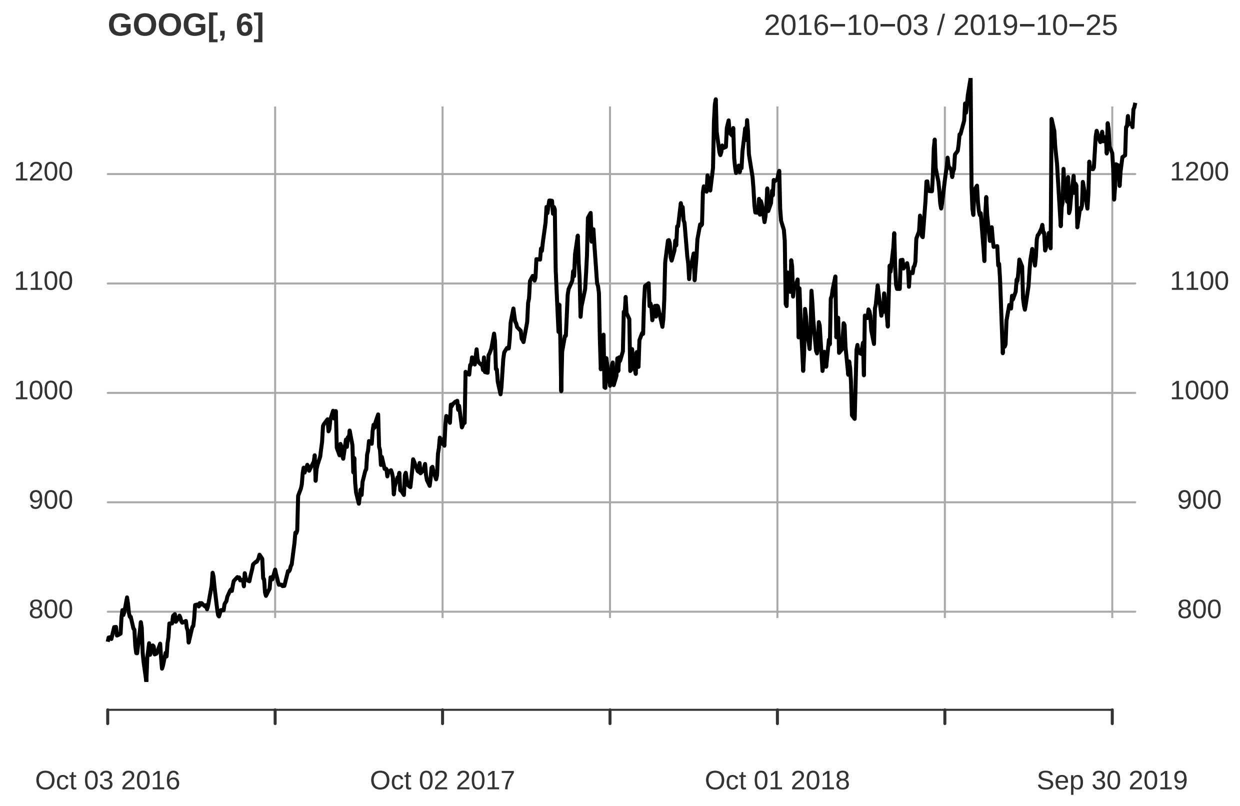
```
## [1] "AMZN" "MSFT" "GOOG"
```

We won't go into Bollinger Bang charts and will only plot their adjusted time series.

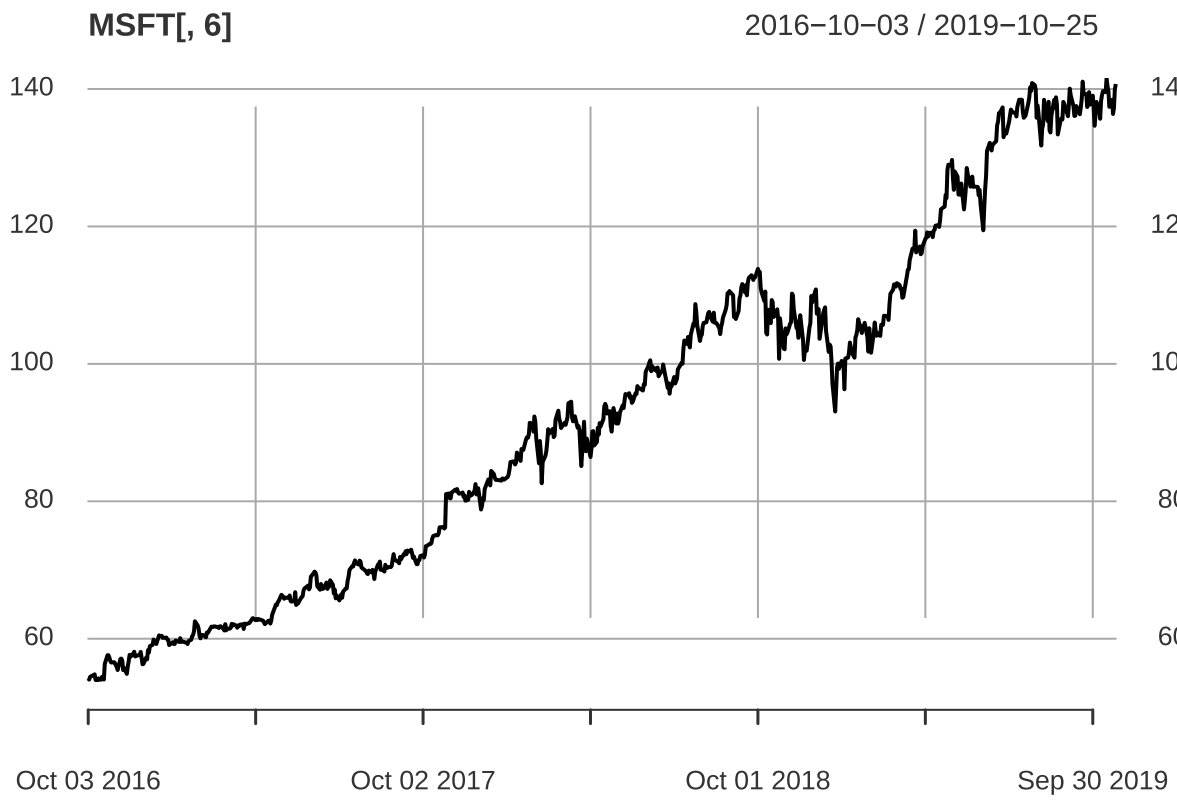
```
plot(AMZN[,6])
```


AMZN[, 6]
2016−10−03 / 2019−10−25
2000
1800
1600
1400
1200
1000
800
Oct 03 2016
Oct 02 2017
Oct 01 2018
Sep 30 2019

```r
plot(GOOG[,6])
```


GOOG[, 6]
2016−10−03 / 2019−10−25
1200
1100
1000
900
800
Oct 03 2016
Oct 02 2017
Oct 01 2018
Sep 30 2019

```r
plot(MSFT[,6])
```



All the stocks seem healthy in the fact that their means show an increasing trend.

```
amazon = c(AMZN[,6])
google = c(GOOG[,6])
microsoft = c(MSFT[,6])

amazon_1 = rep(0, length(amazon))
amazon_1[1:(length(amazon)-1)] = c(amazon[2:length(amazon)])
amazon_2 = rep(0, length(amazon))
amazon_2[1:(length(amazon)-2)] = c(amazon[3:length(amazon)])
amazon_3 = rep(0, length(amazon))
amazon_3[1:(length(amazon)-3)] = c(amazon[4:length(amazon)])
#plot(x_t, x_t_1)
cor(amazon, amazon_1)
```

```
##                    [,1]
## AMZN.Adjusted 0.9863523
```

```
cor(amazon, amazon_2)
```

```
##                    [,1]
## AMZN.Adjusted 0.9727468
```

```
cor(amazon, amazon_3)
```

```
##                    [,1]
## AMZN.Adjusted 0.9597422
```

```r
google_1 = rep(0, (length(google)-1))
google_1[1:(length(google)-1)] = c(google[1:(length(google)-1)])
cor(x = as.vector(google[2:772]), y = google_1)
```

```
## [1] 0.9937932
```

```r
cor(google[3:772], google[1:770])
```

```
##               GOOG.Adjusted
## GOOG.Adjusted      0.987505
```

```r
cor(google[4:772], google[1:769])
```

```
##               GOOG.Adjusted
## GOOG.Adjusted     0.9823539
```

```r
cor(microsoft[2:772], microsoft[1:771])
```

```
##               MSFT.Adjusted
## MSFT.Adjusted     0.9985696
```

```r
cor(microsoft[3:772], microsoft[1:770])
```

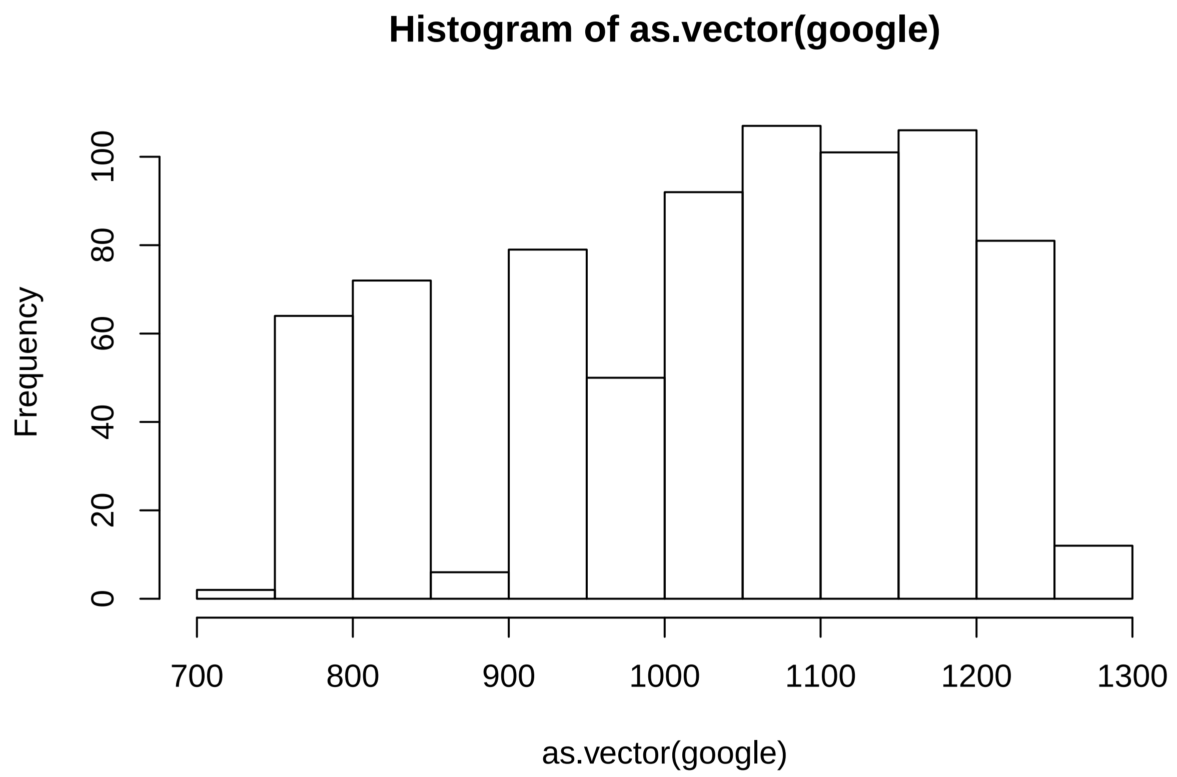
```
##               MSFT.Adjusted
## MSFT.Adjusted      0.997512
```

```r
cor(microsoft[4:772], microsoft[1:769])
```

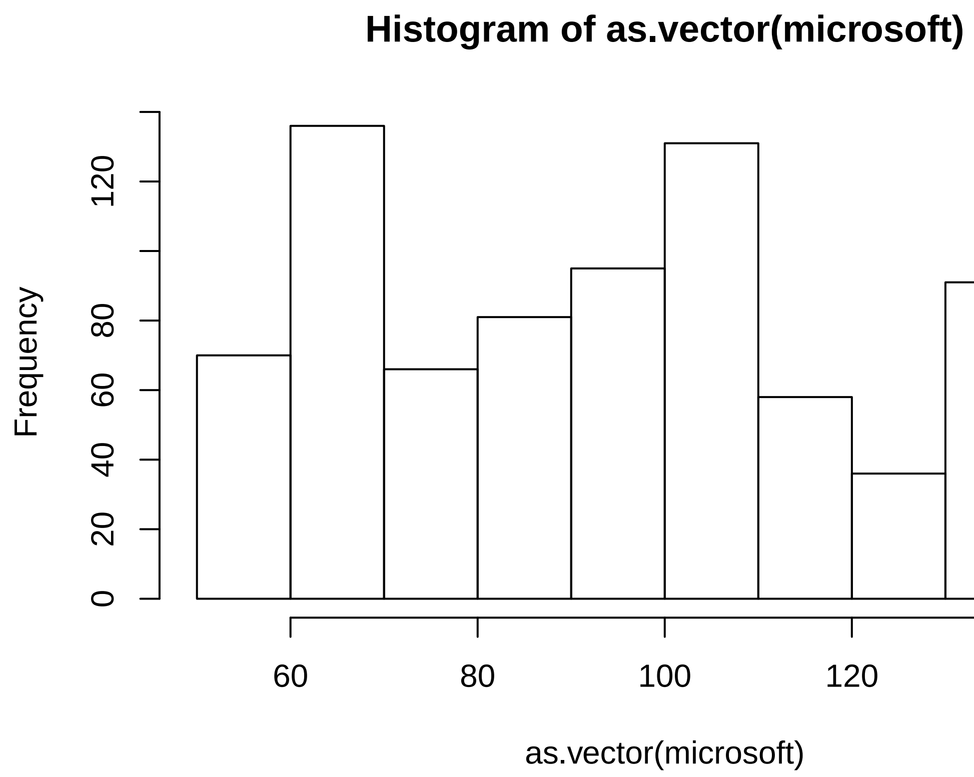
```
##               MSFT.Adjusted
## MSFT.Adjusted     0.9968224
```

All of them seem very positively autocorrelated. This was checked with 1-lag, 2-lag, and 3-lag.

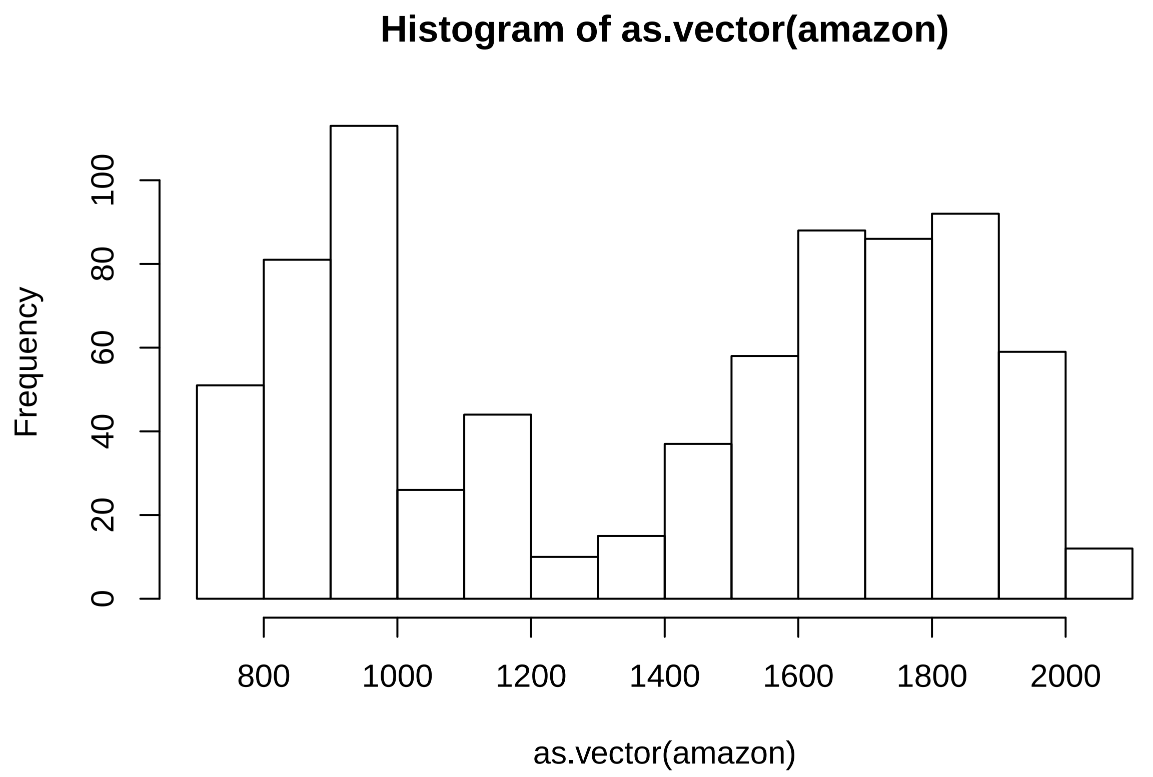
```r
hist(as.vector(google))
```



```r
hist(as.vector(microsoft))
```



```r
hist(as.vector(amazon))
```

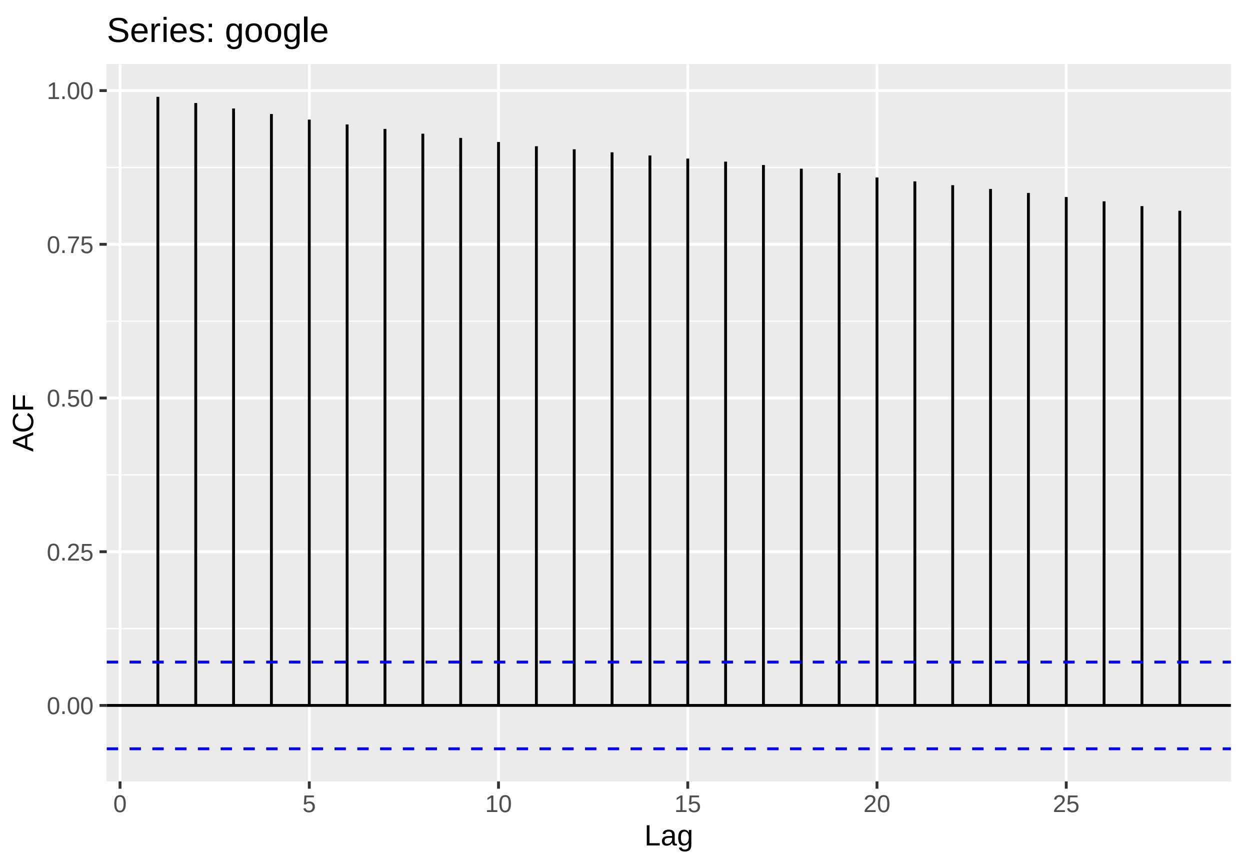


If the data had been stationary, we'd have seen consistent frequency histograms. Looking at the time series plot itself, we can easily infer that the mean is changing and they aren't stationary. Making a 50-50 split of the data will return different mean and variances.

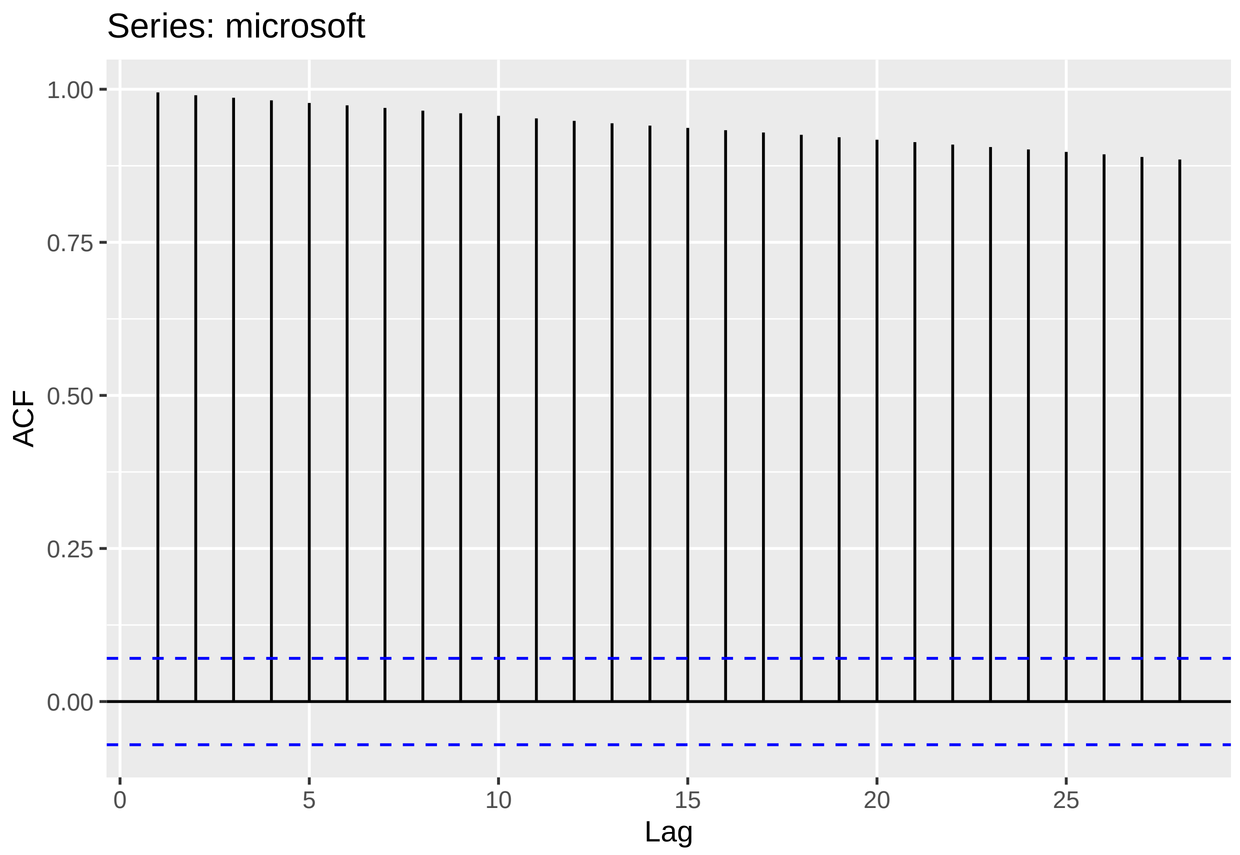
```
library(forecast)
```

```
## Registered S3 methods overwritten by 'forecast':
##   method             from
##   fitted.fracdiff    fracdiff
##   residuals.fracdiff fracdiff
```

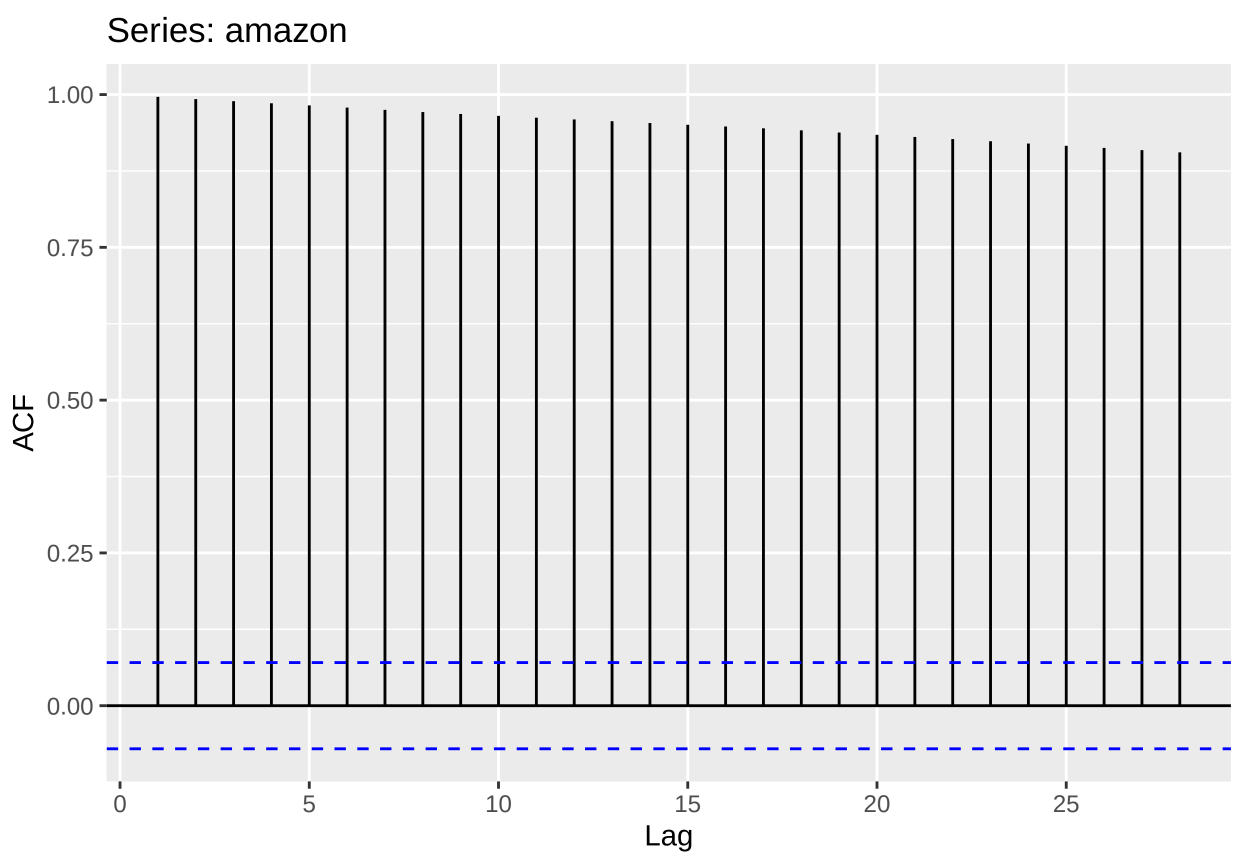
```
library(tseries)

ggAcf(google)
```



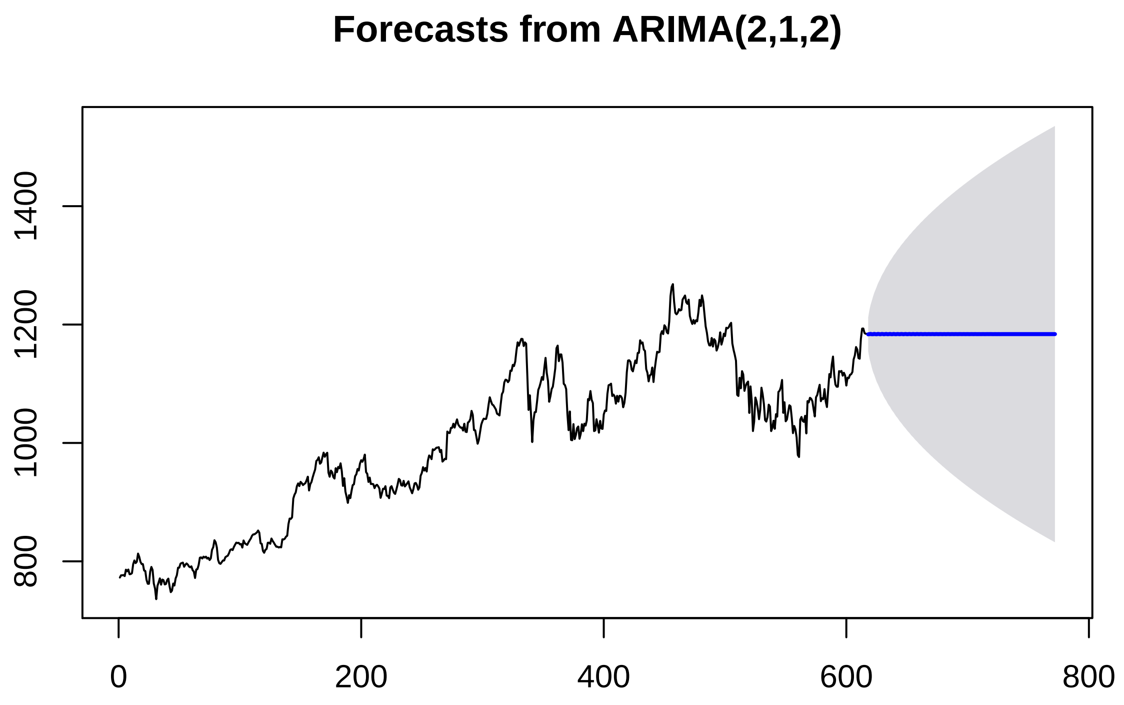
```r
ggAcf(microsoft)
```



```r
ggAcf(amazon)
```

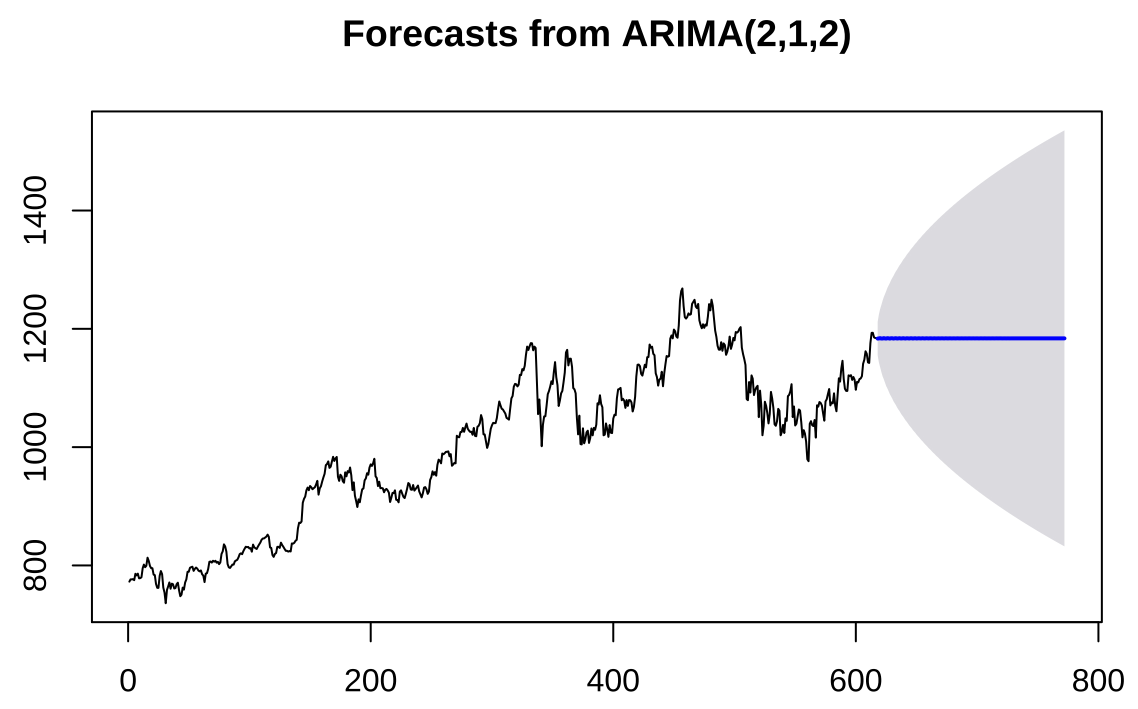


```r
fcast = forecast(auto.arima(as.numeric(google)[1:617]), h = 155, level = c(95))
plot(fcast)
```

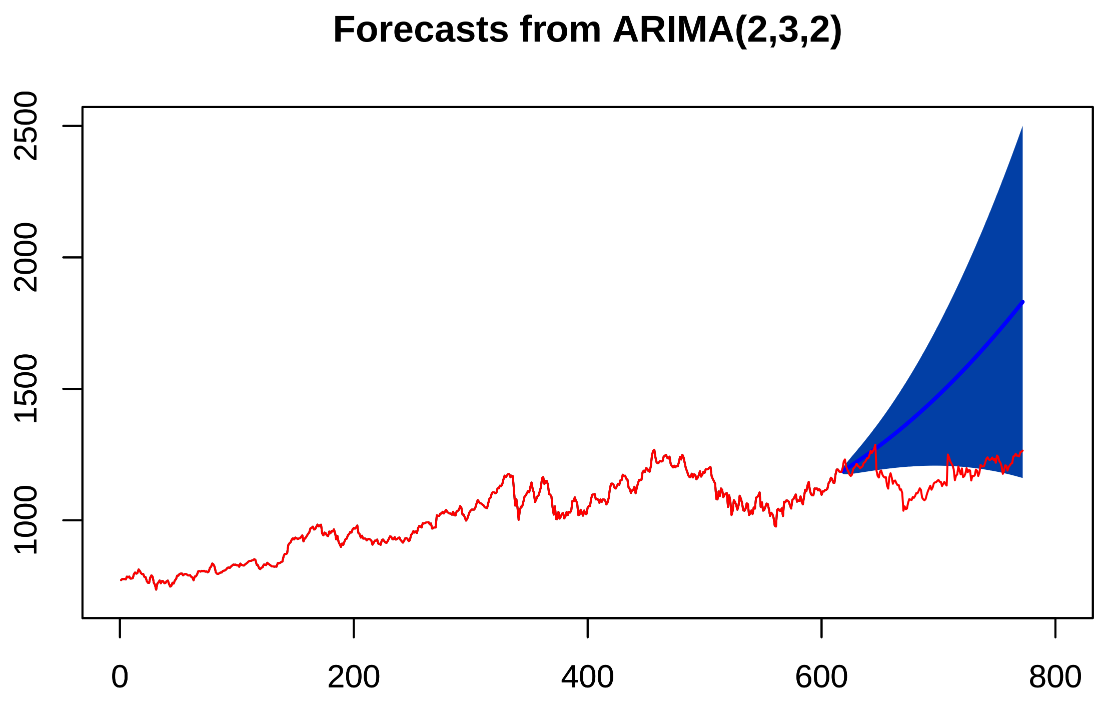


It can be seen that the model isn't very good. We should smoothen the time series by the taking differences.

```
google_diff1 = rep(0, length(google)-1)
google_diff1 = as.vector(google)[2:772] - google[1:771]
fcast = forecast(auto.arima(as.numeric(google)[1:617]), h = 155, level = c(95))
plot(fcast)
```

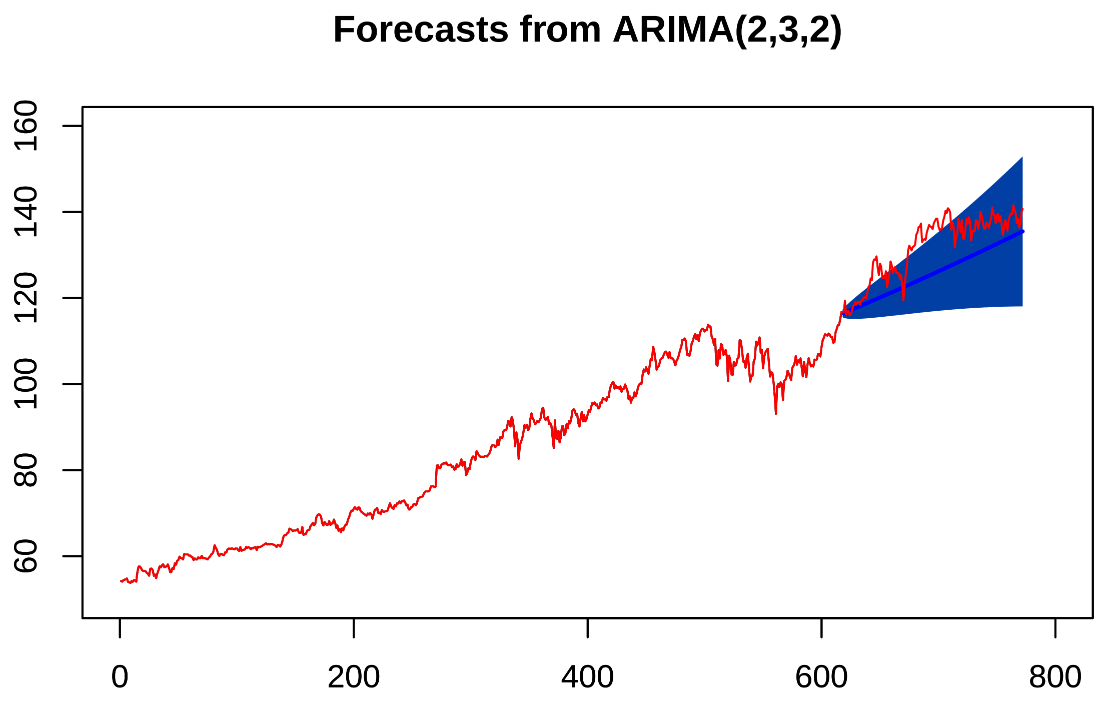


```
fit_google = forecast(Arima(google[1:617], order = c(2,3,2)), h = 155, level = c(50))
plot(fit_google, ylim = c(700,2500), xlim = c(0,800))
par(new = T)
plot(as.vector(google), ylim = c(700,2500), xlim = c(0,800), xlab="", ylab="", main="", col="red", type
```



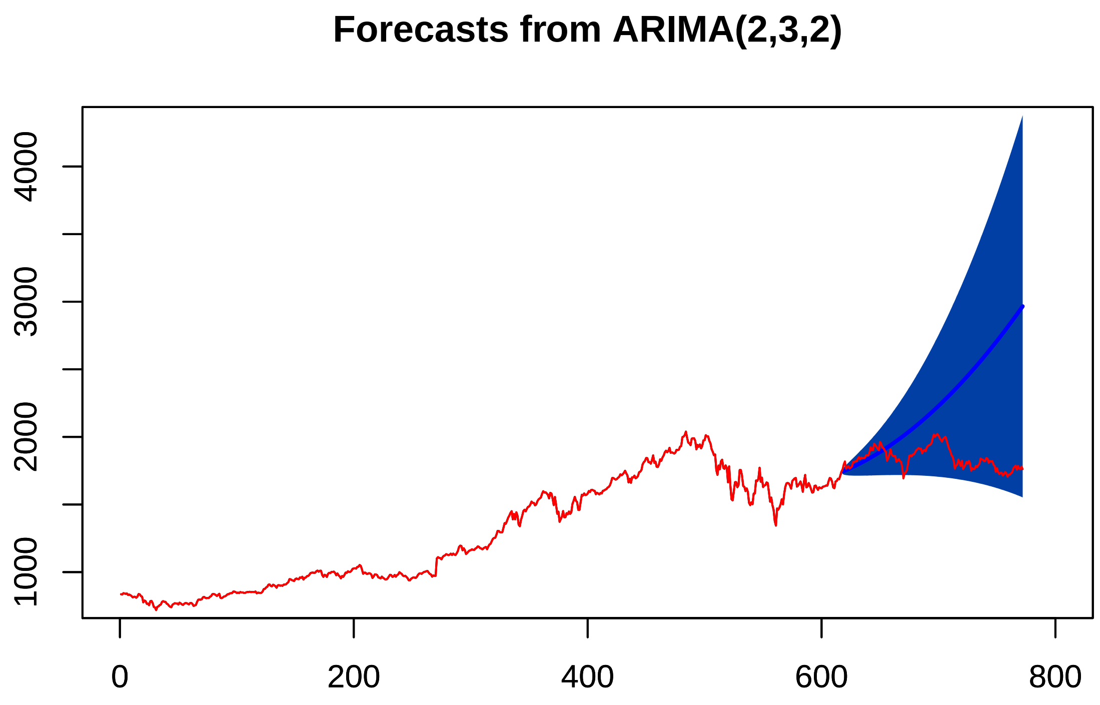
Both predictions on the Google stock have a high variability. The first fit seems better since it relates to the actual data much better. Doing the same analysis on different stocks.

```
fit_msft = forecast(Arima(microsoft[1:617], order = c(2,3,2)), h = 155, level = c(50))
plot(fit_msft, ylim = c(50, 160), xlim = c(0,800))
par(new = T)
plot(as.vector(microsoft), ylim = c(50,160), xlim = c(0,800), xlab="", ylab="", main="", col="red", typ
```



Microsoft's stock is predicted much better! Now for Amazon : –

```
fit_amzn = forecast(Arima(amazon[1:617], order = c(2,3,2)), h = 155, level = c(50))
plot(fit_amzn, xlim = c(0,800), ylim = c(800, 4300))
par(new = T)
plot(as.vector(amazon), ylim = c(800,4300), xlim = c(0,800), xlab="", ylab="", main="", col="red", type
```



Microsoft's stock was predicted the best with the least variance seen due to the confidence intervals. Amazon's was still not predicted very well when we check the actual data.

Question 3

```r
library(mlbench)
data("BreastCancer")
library(superml)
```

```
## Loading required package: R6
```

```r
library(MASS)
```

Viewing the dataframe, we see that the Id column is a character column. All other are categorical variables encoded as numbers.

```r
BreastCancer$Id = as.numeric(BreastCancer$Id)
lbl = superml::LabelEncoder$new()
BreastCancer$Class = lbl$fit_transform(BreastCancer$Class)
```

This encoding converts 'benign' to '0' and 'malignant' to '1' (retains the column as a factor).

Let's check for Spearman correlation with the dependent variables. We may very well discard the column *Id*.

```r
cor(data.matrix(BreastCancer), method = 'spearman')
```

```
##                            Id Cl.thickness   Cell.size  Cell.shape
## Id                1.000000000 -0.004408568 -0.04332721 -0.06009478
## Cl.thickness    -0.004408568  1.000000000  0.66646314  0.66412065
## Cell.size       -0.043327205  0.666463145  1.00000000  0.89190373
## Cell.shape      -0.060094775  0.664120648  0.89190373  1.00000000
## Marg.adhesion   -0.049512616  0.541592582  0.74251672  0.71173014
## Epith.c.size    -0.087263125  0.583978964  0.78715882  0.75923394
## Bare.nuclei               NA           NA          NA          NA
## Bl.cromatin     -0.095749854  0.538092726  0.71932965  0.69231583
## Normal.nucleoli -0.070964365  0.570398358  0.75719871  0.72526781
## Mitoses         -0.075138482  0.418717438  0.50878003  0.47274889
## Class           -0.105498425  0.682451869  0.85548668  0.83639413
##                 Marg.adhesion Epith.c.size Bare.nuclei Bl.cromatin
## Id                -0.04951262  -0.08726313          NA -0.09574985
## Cl.thickness       0.54159258   0.58397896          NA  0.53809273
## Cell.size          0.74251672   0.78715882          NA  0.71932965
## Cell.shape         0.71173014   0.75923394          NA  0.69231583
## Marg.adhesion      1.00000000   0.66780890          NA  0.62451532
## Epith.c.size       0.66780890   1.00000000          NA  0.63950687
## Bare.nuclei                NA           NA           1          NA
## Bl.cromatin        0.62451532   0.63950687          NA  1.00000000
## Normal.nucleoli    0.63430951   0.70599721          NA  0.66230918
## Mitoses            0.44699240   0.48025536          NA  0.38694363
## Class              0.72799520   0.76273087          NA  0.74035037
##                 Normal.nucleoli     Mitoses      Class
## Id                  -0.07096437 -0.07513848 -0.1054984
## Cl.thickness         0.57039836  0.41871744  0.6824519
## Cell.size            0.75719871  0.50878003  0.8554867
## Cell.shape           0.72526781  0.47274889  0.8363941
## Marg.adhesion        0.63430951  0.44699240  0.7279952
## Epith.c.size         0.70599721  0.48025536  0.7627309
## Bare.nuclei                  NA          NA         NA
## Bl.cromatin          0.66230918  0.38694363  0.7403504
## Normal.nucleoli      1.00000000  0.50414036  0.7438226
## Mitoses              0.50414036  1.00000000  0.5267662
## Class                0.74382258  0.52676617  1.0000000
```

Cell.size and *Cell.shape* show very good correlations ($\rho > 0.8$). *Bare.nuclei* contains some NAs. Let's remove them and check the correlation.

```r
cor(data.matrix(BreastCancer), method = 'spearman')
```

```
##                           Id Cl.thickness   Cell.size  Cell.shape
## Id               1.000000000 -0.004408568 -0.04332721 -0.06009478
## Cl.thickness    -0.004408568  1.000000000  0.66646314  0.66412065
## Cell.size       -0.043327205  0.666463145  1.00000000  0.89190373
## Cell.shape      -0.060094775  0.664120648  0.89190373  1.00000000
## Marg.adhesion   -0.049512616  0.541592582  0.74251672  0.71173014
## Epith.c.size    -0.087263125  0.583978964  0.78715882  0.75923394
## Bare.nuclei               NA           NA          NA          NA
## Bl.cromatin     -0.095749854  0.538092726  0.71932965  0.69231583
## Normal.nucleoli -0.070964365  0.570398358  0.75719871  0.72526781
## Mitoses         -0.075138482  0.418717438  0.50878003  0.47274889
## Class           -0.105498425  0.682451869  0.85548668  0.83639413
##                 Marg.adhesion Epith.c.size Bare.nuclei Bl.cromatin
## Id                -0.04951262  -0.08726313          NA -0.09574985
## Cl.thickness       0.54159258   0.58397896          NA  0.53809273
## Cell.size          0.74251672   0.78715882          NA  0.71932965
## Cell.shape         0.71173014   0.75923394          NA  0.69231583
## Marg.adhesion      1.00000000   0.66780890          NA  0.62451532
## Epith.c.size       0.66780890   1.00000000          NA  0.63950687
## Bare.nuclei                NA           NA           1          NA
## Bl.cromatin        0.62451532   0.63950687          NA  1.00000000
## Normal.nucleoli    0.63430951   0.70599721          NA  0.66230918
## Mitoses            0.44699240   0.48025536          NA  0.38694363
## Class              0.72799520   0.76273087          NA  0.74035037
##                 Normal.nucleoli     Mitoses      Class
## Id                  -0.07096437 -0.07513848 -0.1054984
## Cl.thickness         0.57039836  0.41871744  0.6824519
## Cell.size            0.75719871  0.50878003  0.8554867
## Cell.shape           0.72526781  0.47274889  0.8363941
## Marg.adhesion        0.63430951  0.44699240  0.7279952
## Epith.c.size         0.70599721  0.48025536  0.7627309
## Bare.nuclei                  NA          NA         NA
## Bl.cromatin          0.66230918  0.38694363  0.7403504
## Normal.nucleoli      1.00000000  0.50414036  0.7438226
## Mitoses              0.50414036  1.00000000  0.5267662
## Class                0.74382258  0.52676617  1.0000000
```

$\rho = 0.74$ for *bare.nuclei*.

Now, since the response variable is a binary variable, we'll use the binomial distribution as the link function.

```
for (i in 1:11){
  BreastCancer[,i] = as.numeric(BreastCancer[,i])
}

BreastCancer = na.omit(BreastCancer)
```

```
model = stats::glm(Class ~ . , data = BreastCancer,  family = binomial(link = "logit"), control = list(
summary(model)
```

```
##
## Call:
## stats::glm(formula = Class ~ ., family = binomial(link = "logit"),
##     data = BreastCancer, control = list(maxit = 100))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4892  -0.1155  -0.0613   0.0222   2.4672
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.016e+01  1.453e+00  -6.989 2.76e-12 ***
## Id               3.997e-08  7.336e-07   0.054 0.956546
## Cl.thickness     5.352e-01  1.419e-01   3.771 0.000163 ***
## Cell.size       -6.481e-03  2.092e-01  -0.031 0.975290
## Cell.shape       3.228e-01  2.308e-01   1.399 0.161836
## Marg.adhesion    3.307e-01  1.234e-01   2.679 0.007385 **
## Epith.c.size     9.640e-02  1.567e-01   0.615 0.538490
## Bare.nuclei      3.839e-01  9.548e-02   4.021 5.79e-05 ***
## Bl.cromatin      4.479e-01  1.716e-01   2.609 0.009070 **
## Normal.nucleoli  2.135e-01  1.131e-01   1.887 0.059221 .
## Mitoses          5.382e-01  3.263e-01   1.650 0.099035 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 884.35  on 682  degrees of freedom
## Residual deviance: 102.90  on 672  degrees of freedom
## AIC: 124.9
##
## Number of Fisher Scoring iterations: 8
```

The statistics displayed by the *stats* :: *glm* function displays : –

1. **Call** - What actual command was given to generate the generalised linear model.
2. **Deviance Residuals** - Gives the summary of the deviance residuals - their minimum value, 25th, 50th, and 75th percentile, and the maximum value.
3. **Coefficients** - These are the coefficients (or slopes) given to the different independent variables. Note the significance codes below. Our data has the intercept, Cl.thickness, Marg.adhesion, Bare.nuclei, Bl.cromatin as signinficant variables. These coefficients are actually in units of logit. These are log of odd ratios.
4. **Null and Residual deviance** - The null model (containing random values) had a deviance of 884.35 on 682 degrees of freedom. The residual deviance could bring it down to 102.90 — a significant drop with a reduction to 672 degrees of freedom. Lesser this deviance, better the fit.
5. ***AIC** - The Akaike Information Criteria or AIC is 124.9. This model can be compared with other such models on the basis of complexity with this measure.
6. **Number of Fisher Scoring Iterations** - The maximum iterations were given to be 100. This number tells us that the model was itself able to converge (using Newton's approximation method) with 8 iterations.

```
exp(cbind(coef(model), confint(model, level = 0.95)))
```

```
## Waiting for profiling to be done...

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

##                                        2.5 %       97.5 %
## (Intercept)     3.883384e-05 2.026971e-06 0.0006751557
## Id              1.000000e+00 9.999981e-01 1.0000006406
## Cl.thickness    1.707765e+00 1.316023e+00 2.3112071985
## Cell.size       9.935402e-01 6.734836e-01 1.5505839408
## Cell.shape      1.381011e+00 8.610222e-01 2.1567226486
## Marg.adhesion   1.391920e+00 1.097572e+00 1.7984950474
## Epith.c.size    1.101202e+00 8.047423e-01 1.4992552531
## Bare.nuclei     1.468056e+00 1.226733e+00 1.7906219676
## Bl.cromatin     1.564976e+00 1.131276e+00 2.2279653149
## Normal.nucleoli 1.237953e+00 9.981035e-01 1.5630323124
## Mitoses         1.712889e+00 9.914547e-01 3.0297864371
```

The warnings displayed is due to the fact of 'perfect distinguishability', i.e., the data is very able to separate the benign and malignant cases. This gives us exact 0 or 1 probabilities in two cases. The odds discussed earlier were in logit units or log odd ratios, so we exponentiate them. They give us 95% confidence intervals (2.5% on both sides). Here we see that all the variables are positively correlated with our target variable - meaning as they increase from the discrete 0 towards 10, the target variable increases from 0 to 1 (benign to malignant). The biggest odd ratios are seen in *Cell.size* and the intercept. This means that *Cell.size* dominates the prediction towards benign or malignancy. All other have a near about coefficient of 1 (although the confidence interval seems too high) so they are being used as additives in the model.