Assignment.1

Lorenzo Ausiello

2023-10-01

```
vector_1<-runif(10)
vector_2<-runif(10)
vector_1
```

```
##  [1] 0.40863480 0.92052586 0.57906587 0.05738780 0.75653116 0.74042967
##  [7] 0.10916470 0.89102178 0.75830435 0.08941567
```

```
vector_2
```

```
##  [1] 0.3172487 0.5941793 0.6857719 0.9048191 0.6157585 0.6708453 0.2615384
##  [8] 0.9589790 0.9894368 0.8609089
```

```
vector_tot<-append(vector_1,vector_2)
vector_tot
```

```
##  [1] 0.40863480 0.92052586 0.57906587 0.05738780 0.75653116 0.74042967
##  [7] 0.10916470 0.89102178 0.75830435 0.08941567 0.31724867 0.59417933
## [13] 0.68577193 0.90481911 0.61575849 0.67084533 0.26153839 0.95897900
## [19] 0.98943680 0.86090895
```

```
mean_value<-mean(vector_tot)
mean_value
```

```
## [1] 0.6084984
```

```
for (i in vector_tot){
  if (i>mean_value) {
    print('True')}
  else {print('False')
  }
}
```

```
## [1] "False"
## [1] "True"
## [1] "False"
## [1] "False"
## [1] "True"
## [1] "True"
## [1] "False"
## [1] "True"
## [1] "True"
## [1] "False"
## [1] "False"
## [1] "False"
## [1] "True"
## [1] "True"
## [1] "True"
## [1] "True"
## [1] "False"
## [1] "True"
## [1] "True"
## [1] "True"
```

```
vector_3<-runif(100)
vector_3
```

```
##   [1] 0.26427007 0.35358484 0.73481730 0.70266009 0.91220273 0.90866647
##   [7] 0.47760899 0.18565914 0.22110917 0.10883320 0.21572361 0.15971888
##  [13] 0.02274730 0.13690038 0.83726762 0.77172566 0.09589150 0.72058619
##  [19] 0.92698603 0.37064816 0.96041826 0.48985500 0.91481084 0.45532819
##  [25] 0.91792306 0.80237919 0.49227862 0.67175617 0.75051819 0.75494617
##  [31] 0.69954350 0.69093314 0.01543568 0.99002859 0.04556116 0.61381252
##  [37] 0.81511418 0.42615236 0.16709044 0.56924164 0.62319252 0.05472515
##  [43] 0.10858662 0.68871750 0.81846323 0.51260561 0.98536291 0.04624998
##  [49] 0.77662376 0.79980649 0.95931641 0.10248498 0.68312596 0.92646487
##  [55] 0.76522788 0.03290377 0.85318896 0.20393849 0.87126622 0.77629313
##  [61] 0.76149812 0.96645125 0.40515105 0.28375107 0.69072888 0.48313133
##  [67] 0.57300287 0.70639545 0.54371817 0.34158068 0.74954893 0.85179050
##  [73] 0.71966110 0.22041796 0.06168301 0.89288285 0.96738800 0.31227773
##  [79] 0.33560842 0.64686423 0.58421676 0.09100004 0.89337286 0.60337026
##  [85] 0.45746303 0.49446632 0.85431872 0.66779378 0.43669395 0.88792732
##  [91] 0.40529093 0.75975666 0.22717469 0.46349853 0.98132791 0.17927749
##  [97] 0.57219535 0.57868788 0.98602043 0.53659473
```

```
matrix_1<-matrix(vector_3,nrow = 10)
matrix_1
```

```
##            [,1]      [,2]      [,3]       [,4]       [,5]       [,6]      [,7]
##  [1,] 0.2642701 0.2157236 0.9604183 0.69954350 0.62319252 0.95931641 0.7614981
##  [2,] 0.3535848 0.1597189 0.4898550 0.69093314 0.05472515 0.10248498 0.9664513
##  [3,] 0.7348173 0.0227473 0.9148108 0.01543568 0.10858662 0.68312596 0.4051511
##  [4,] 0.7026601 0.1369004 0.4553282 0.99002859 0.68871750 0.92646487 0.2837511
##  [5,] 0.9122027 0.8372676 0.9179231 0.04556116 0.81846323 0.76522788 0.6907289
##  [6,] 0.9086665 0.7717257 0.8023792 0.61381252 0.51260561 0.03290377 0.4831313
##  [7,] 0.4776090 0.0958915 0.4922786 0.81511418 0.98536291 0.85318896 0.5730029
##  [8,] 0.1856591 0.7205862 0.6717562 0.42615236 0.04624998 0.20393849 0.7063955
##  [9,] 0.2211092 0.9269860 0.7505182 0.16709044 0.77662376 0.87126622 0.5437182
## [10,] 0.1088332 0.3706482 0.7549462 0.56924164 0.79980649 0.77629313 0.3415807
##             [,8]       [,9]     [,10]
##  [1,] 0.74954893 0.58421676 0.4052909
##  [2,] 0.85179050 0.09100004 0.7597567
##  [3,] 0.71966110 0.89337286 0.2271747
##  [4,] 0.22041796 0.60337026 0.4634985
##  [5,] 0.06168301 0.45746303 0.9813279
##  [6,] 0.89288285 0.49446632 0.1792775
##  [7,] 0.96738800 0.85431872 0.5721954
##  [8,] 0.31227773 0.66779378 0.5786879
##  [9,] 0.33560842 0.43669395 0.9860204
## [10,] 0.64686423 0.88792732 0.5365947
```

```
matrix_2<-t(matrix_1)
matrix_2
```

```
##            [,1]       [,2]       [,3]      [,4]       [,5]       [,6]      [,7]
##  [1,] 0.2642701 0.35358484 0.73481730 0.7026601 0.91220273 0.90866647 0.4776090
##  [2,] 0.2157236 0.15971888 0.02274730 0.1369004 0.83726762 0.77172566 0.0958915
##  [3,] 0.9604183 0.48985500 0.91481084 0.4553282 0.91792306 0.80237919 0.4922786
##  [4,] 0.6995435 0.69093314 0.01543568 0.9900286 0.04556116 0.61381252 0.8151142
##  [5,] 0.6231925 0.05472515 0.10858662 0.6887175 0.81846323 0.51260561 0.9853629
##  [6,] 0.9593164 0.10248498 0.68312596 0.9264649 0.76522788 0.03290377 0.8531890
##  [7,] 0.7614981 0.96645125 0.40515105 0.2837511 0.69072888 0.48313133 0.5730029
##  [8,] 0.7495489 0.85179050 0.71966110 0.2204180 0.06168301 0.89288285 0.9673880
##  [9,] 0.5842168 0.09100004 0.89337286 0.6033703 0.45746303 0.49446632 0.8543187
## [10,] 0.4052909 0.75975666 0.22717469 0.4634985 0.98132791 0.17927749 0.5721954
##             [,8]      [,9]     [,10]
##  [1,] 0.18565914 0.2211092 0.1088332
##  [2,] 0.72058619 0.9269860 0.3706482
##  [3,] 0.67175617 0.7505182 0.7549462
##  [4,] 0.42615236 0.1670904 0.5692416
##  [5,] 0.04624998 0.7766238 0.7998065
##  [6,] 0.20393849 0.8712662 0.7762931
##  [7,] 0.70639545 0.5437182 0.3415807
##  [8,] 0.31227773 0.3356084 0.6468642
##  [9,] 0.66779378 0.4366940 0.8879273
## [10,] 0.57868788 0.9860204 0.5365947
```

```
print(matrix_2[2,1])
```

```
## [1] 0.2157236
```

```
matrix_3<-matrix(NA,nrow = 10,ncol=10)
for (i in 1:10){
  for (j in 1:10){
    matrix_3[i,j]<-sum(matrix_2[i,]*matrix_1[,j])
  }
}
matrix_3
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
##  [1,] 3.209651 2.116278 3.593546 2.302802 2.698128 2.893145 2.671363 2.715179
##  [2,] 2.116278 2.913006 3.263135 1.660112 2.464979 2.355934 2.512455 1.953645
##  [3,] 3.593546 3.263135 5.519775 3.252237 3.903675 4.544110 4.148853 4.095128
##  [4,] 2.302802 1.660112 3.252237 3.523958 2.917164 3.094560 2.869039 3.238983
##  [5,] 2.698128 2.464979 3.903675 2.917164 4.026047 4.106539 2.872609 2.997540
##  [6,] 2.893145 2.355934 4.544110 3.094560 4.106539 4.973704 3.285512 3.262371
##  [7,] 2.671363 2.512455 4.148853 2.869039 2.872609 3.285512 3.708725 3.400436
##  [8,] 2.715179 1.953645 4.095128 3.238983 2.997540 3.262371 3.400436 4.219335
##  [9,] 2.858807 2.305141 4.381575 2.966418 3.431519 4.040248 3.122851 3.516971
## [10,] 2.583574 2.822164 4.040460 2.608817 3.318642 3.689957 3.486960 2.849463
##           [,9]     [,10]
##  [1,] 2.858807 2.583574
##  [2,] 2.305141 2.822164
##  [3,] 4.381575 4.040460
##  [4,] 2.966418 2.608817
##  [5,] 3.431519 3.318642
##  [6,] 4.040248 3.689957
##  [7,] 3.122851 3.486960
##  [8,] 3.516971 2.849463
##  [9,] 4.120456 3.108424
## [10,] 3.108424 3.925532
```

```
matrix_4<-matrix_2%*%matrix_1
matrix_4
```

```
##           [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
##  [1,] 3.209651 2.116278 3.593546 2.302802 2.698128 2.893145 2.671363 2.715179
##  [2,] 2.116278 2.913006 3.263135 1.660112 2.464979 2.355934 2.512455 1.953645
##  [3,] 3.593546 3.263135 5.519775 3.252237 3.903675 4.544110 4.148853 4.095128
##  [4,] 2.302802 1.660112 3.252237 3.523958 2.917164 3.094560 2.869039 3.238983
##  [5,] 2.698128 2.464979 3.903675 2.917164 4.026047 4.106539 2.872609 2.997540
##  [6,] 2.893145 2.355934 4.544110 3.094560 4.106539 4.973704 3.285512 3.262371
##  [7,] 2.671363 2.512455 4.148853 2.869039 2.872609 3.285512 3.708725 3.400436
##  [8,] 2.715179 1.953645 4.095128 3.238983 2.997540 3.262371 3.400436 4.219335
##  [9,] 2.858807 2.305141 4.381575 2.966418 3.431519 4.040248 3.122851 3.516971
## [10,] 2.583574 2.822164 4.040460 2.608817 3.318642 3.689957 3.486960 2.849463
##           [,9]     [,10]
##  [1,] 2.858807 2.583574
##  [2,] 2.305141 2.822164
##  [3,] 4.381575 4.040460
##  [4,] 2.966418 2.608817
##  [5,] 3.431519 3.318642
##  [6,] 4.040248 3.689957
##  [7,] 3.122851 3.486960
##  [8,] 3.516971 2.849463
##  [9,] 4.120456 3.108424
## [10,] 3.108424 3.925532
```

```
setwd("C:/Users/loaus/OneDrive - stevens.edu/STEVENS/Financial Lab Database Design/Assignment/Assignmen

data<-read.csv("stock_data-1.csv")
head(data)
```

```
##            X     AAPL     AMGN      AXP       BA      CAT CRM     CSCO      CVX
## 1 1996-01-02 0.286830 14.56250 12.10832 39.93750 14.87500  NA 4.243055 26.43750
## 2 1996-01-03 0.286830 14.40625 12.10832 39.56250 15.12500  NA 4.076389 26.50000
## 3 1996-01-04 0.281808 13.78125 11.99890 38.56250 15.00000  NA 3.923611 27.25000
## 4 1996-01-05 0.305804 14.09375 11.96243 39.25000 15.25000  NA 3.972222 27.68750
## 5 1996-01-08 0.309152 13.85938 11.96243 40.12500 15.18750  NA 3.934028 27.81250
## 6 1996-01-09 0.292411 13.53125 11.78008 39.67969 14.78125  NA 3.631944 27.92188
##        DIS DOW GS        HD      IBM     INTC      JNJ      JPM       KO
## 1 20.01773  NA NA 10.527778 22.71875 7.328125 21.06250 19.58333 18.75000
## 2 20.14104  NA NA 10.333333 22.31250 7.218750 21.90625 19.58333 18.90625
## 3 19.89442  NA NA 10.305555 21.71875 7.187500 21.68750 18.75000 18.75000
## 4 20.26435  NA NA 10.055555 22.15625 7.187500 21.68750 18.66667 18.65625
## 5 20.38767  NA NA  9.777778 22.28125 7.203125 21.96875 18.66667 18.78125
## 6 20.41850  NA NA  9.666667 21.68750 6.875000 22.09375 18.20833 18.53125
##       MCD      MMM     MRK     MSFT      NKE       PG     TRV      UNH  V
## 1 22.7500 33.87500 32.1250 5.609375 4.445313 20.78125 28.2500 8.078125 NA
## 2 22.7500 33.81250 31.6875 5.429688 4.312500 21.40625 28.6250 8.109375 NA
## 3 22.8750 33.68750 31.8750 5.460938 4.265625 21.75000 29.0000 8.187500 NA
## 4 22.5000 33.75000 31.5000 5.398438 4.132813 21.84375 29.0625 7.859375 NA
## 5 22.5625 33.50000 31.9375 5.390625 4.203125 21.93750 29.1875 7.703125 NA
## 6 22.1875 33.01562 31.7500 5.011719 4.117188 21.90625 28.9375 7.265625 NA
##         VZ     WBA     WMT
## 1 30.46456 7.53125 11.6250
## 2 31.42009 7.50000 11.7500
## 3 30.85801 7.40625 11.8750
## 4 31.19526 7.68750 11.6875
## 5 30.97043 7.62500 11.6875
## 6 30.93530 7.56250 11.5000
```

```r
data <- subset(data, select = -c( which(colMeans(is.na(data)) > 0)))
head(data)
```

```
##             X     AAPL     AMGN      AXP       BA      CAT     CSCO      CVX
## 1 1996-01-02 0.286830 14.56250 12.10832 39.93750 14.87500 4.243055 26.43750
## 2 1996-01-03 0.286830 14.40625 12.10832 39.56250 15.12500 4.076389 26.50000
## 3 1996-01-04 0.281808 13.78125 11.99890 38.56250 15.00000 3.923611 27.25000
## 4 1996-01-05 0.305804 14.09375 11.96243 39.25000 15.25000 3.972222 27.68750
## 5 1996-01-08 0.309152 13.85938 11.96243 40.12500 15.18750 3.934028 27.81250
## 6 1996-01-09 0.292411 13.53125 11.78008 39.67969 14.78125 3.631944 27.92188
##        DIS        HD      IBM     INTC      JNJ      JPM       KO     MCD
## 1 20.01773 10.527778 22.71875 7.328125 21.06250 19.58333 18.75000 22.7500
## 2 20.14104 10.333333 22.31250 7.218750 21.90625 19.58333 18.90625 22.7500
## 3 19.89442 10.305555 21.71875 7.187500 21.68750 18.75000 18.75000 22.8750
## 4 20.26435 10.055555 22.15625 7.187500 21.68750 18.66667 18.65625 22.5000
## 5 20.38767  9.777778 22.28125 7.203125 21.96875 18.66667 18.78125 22.5625
## 6 20.41850  9.666667 21.68750 6.875000 22.09375 18.20833 18.53125 22.1875
##        MMM     MRK     MSFT      NKE       PG     TRV      UNH       VZ     WBA
## 1 33.87500 32.1250 5.609375 4.445313 20.78125 28.2500 8.078125 30.46456 7.53125
## 2 33.81250 31.6875 5.429688 4.312500 21.40625 28.6250 8.109375 31.42009 7.50000
## 3 33.68750 31.8750 5.460938 4.265625 21.75000 29.0000 8.187500 30.85801 7.40625
## 4 33.75000 31.5000 5.398438 4.132813 21.84375 29.0625 7.859375 31.19526 7.68750
## 5 33.50000 31.9375 5.390625 4.203125 21.93750 29.1875 7.703125 30.97043 7.62500
## 6 33.01562 31.7500 5.011719 4.117188 21.90625 28.9375 7.265625 30.93530 7.56250
##       WMT
## 1 11.6250
## 2 11.7500
## 3 11.8750
## 4 11.6875
## 5 11.6875
## 6 11.5000
```

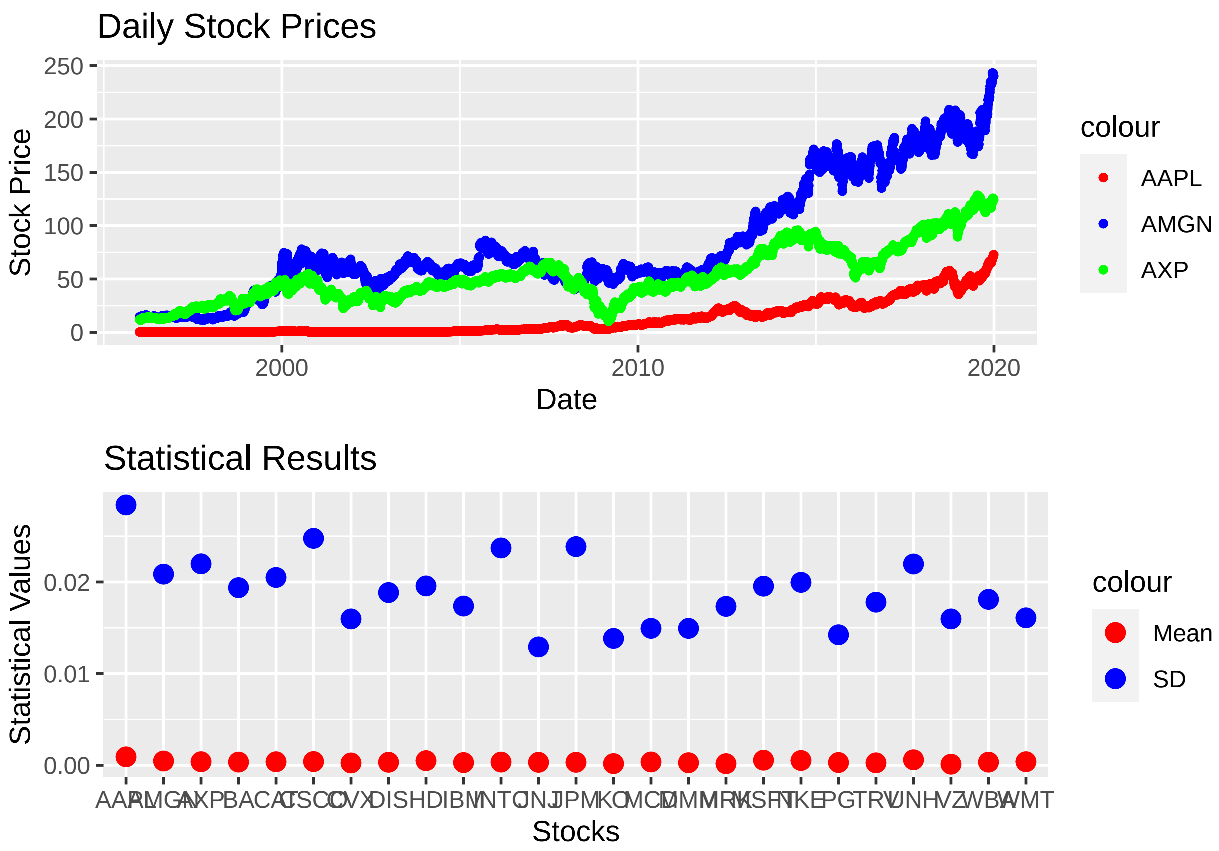
```
log_returns<-matrix(NA,nrow=6041,ncol=26)
colnames(log_returns)<-colnames(data)
log_returns<-data.frame(log_returns)

for (i in 2:26){
  for (j in 2:6041){
    log_returns[j,i]<-log(data[j,i]/data[j-1,i])
  }
}

log_returns<-log_returns[-1,]
log_returns[,1]<-data[-1,1]
head(log_returns)
```

```
##            X        AAPL        AMGN          AXP           BA          CAT
## 2 1996-01-03  0.00000000 -0.01078759  0.000000000 -0.009434032  0.016667052
## 3 1996-01-04 -0.01766372 -0.04435317 -0.009077177 -0.025601398 -0.008298803
## 4 1996-01-05  0.08171839  0.02242246 -0.003044156  0.017671143  0.016529302
## 5 1996-01-08  0.01088869 -0.01676954  0.000000000  0.022048137 -0.004106782
## 6 1996-01-09 -0.05567272 -0.02396007 -0.015361271 -0.011160162 -0.027113235
## 7 1996-01-10  0.04478403 -0.02573241 -0.034649121 -0.033433214 -0.010627093
##           CSCO          CVX          DIS           HD          IBM         INTC
## 2 -0.040071981  0.002361276  0.006141243 -0.018642404 -0.018043515 -0.015037877
## 3 -0.038199144  0.027908788 -0.012320435 -0.002691813 -0.026971117 -0.004338402
## 4  0.012313233  0.015927527  0.018424292 -0.024557852  0.019943681  0.000000000
## 5 -0.009661798  0.004504512  0.006066728 -0.028012958  0.005625894  0.002171554
## 6 -0.079895795  0.003924872  0.001510949 -0.011428684 -0.027009460 -0.046623316
## 7  0.020814407 -0.052265699 -0.040570375  0.011428684  0.005747142 -0.016036999
##            JNJ          JPM           KO          MCD          MMM          MRK
## 2  0.039277776  0.000000000  0.008298803  0.000000000 -0.001846723 -0.013712262
## 3 -0.010035927 -0.043485146 -0.008298803  0.005479466 -0.003703708  0.005899722
## 4  0.000000000 -0.004454386 -0.005012542 -0.016529302  0.001853569 -0.011834458
## 5  0.012884931  0.000000000  0.006677821  0.002773927 -0.007434978  0.013793322
## 6  0.005673774 -0.024859965 -0.013400536 -0.016760169 -0.014564505 -0.005888143
## 7 -0.033072748 -0.009195541 -0.017007213 -0.017045867 -0.008077972 -0.036076056
##           MSFT          NKE           PG          TRV          UNH           VZ
## 2 -0.032557631 -0.030332500  0.029631798  0.013187004  0.003861009  0.030883522
## 3  0.005738896 -0.010929071  0.015930822  0.013015368  0.009587801 -0.018051105
## 4 -0.011510917 -0.031630423  0.004301082  0.002152853 -0.040901514  0.010869672
## 5 -0.001448319  0.016870007  0.004282662  0.004291852 -0.020080996 -0.007233283
## 6 -0.072882364 -0.020657890 -0.001425517 -0.008602204 -0.058471768 -0.001134952
## 7  0.026914398 -0.003802285 -0.012921931 -0.010857870 -0.002152853 -0.007980903
##            WBA         WMT
## 2 -0.004158010  0.01069529
## 3 -0.012578782  0.01058211
## 4  0.037271395 -0.01591546
## 5 -0.008163311  0.00000000
## 6 -0.008230499 -0.01617286
## 7 -0.008298803 -0.01643873
```

```r
stat_fun<-function(x){
  c(mean(x),sd(x))
}

statistics<-sapply(log_returns[,-1],FUN=stat_fun)
rownames(statistics)<-c("mean","sd")
statistics
```

```
##              AAPL         AMGN          AXP           BA         CAT
## mean 0.0009168344 0.0004641248 0.0003855638 0.0003478159 0.000379848
## sd   0.0284047796 0.0208557858 0.0219789482 0.0193786437 0.020493476
##              CSCO          CVX          DIS           HD          IBM
## mean 0.0004002217 0.0002502413 0.0003264233 0.0005012099 0.0002923392
## sd   0.0247623992 0.0159682591 0.0188428129 0.0195658648 0.0173720510
##              INTC          JNJ          JPM           KO          MCD
## mean 0.0003470648 0.0003197527 0.0003240281 0.0001789796 0.0003573148
## sd   0.0237095985 0.0129114075 0.0238587249 0.0138393263 0.0149395005
##               MMM          MRK         MSFT          NKE           PG
## mean 0.0002726557 0.0001724428 0.0005522446 0.0005167696 0.0002963599
## sd   0.0149365129 0.0173385301 0.0195438394 0.0199581843 0.0142370700
##               TRV          UNH          VZ          WBA          WMT
## mean 0.0002607877 0.0005950182 0.000115521 0.0003405546 0.0003856492
## sd   0.0177976490 0.0219630363 0.015967949 0.0181026340 0.0160858239
```

```r
data$X<-as.Date(data$X)


library(ggplot2)
plot1 <- ggplot(data, aes(x =X)) +
  geom_point(aes(y = AAPL, color = "AAPL"), size = 1) +
  geom_point(aes(y = AMGN, color = "AMGN"), size = 1) +
  geom_point(aes(y = AXP, color = "AXP"), size = 1) +
  labs(title = "Daily Stock Prices",
       x = "Date",
       y = "Stock Price") +
  scale_color_manual(values = c("AAPL" = "red", "AMGN" = "blue", "AXP" = "green"))

plot2 <- ggplot() +
  geom_point(aes(x = colnames(statistics), y = statistics[1,],color="Mean"), size = 3)+
  geom_point(aes(x = colnames(statistics), y = statistics[2,],color="SD"), size = 3)+
  labs(title = "Statistical Results",
       x = "Stocks",
       y = "Statistical Values") +
  scale_color_manual(values = c("Mean" = "red", "SD" = "blue"))

library(gridExtra)
combined_plot <- grid.arrange(plot1, plot2, ncol = 1)
```



```
# Calculate log returns using pipes
library(dplyr)
```

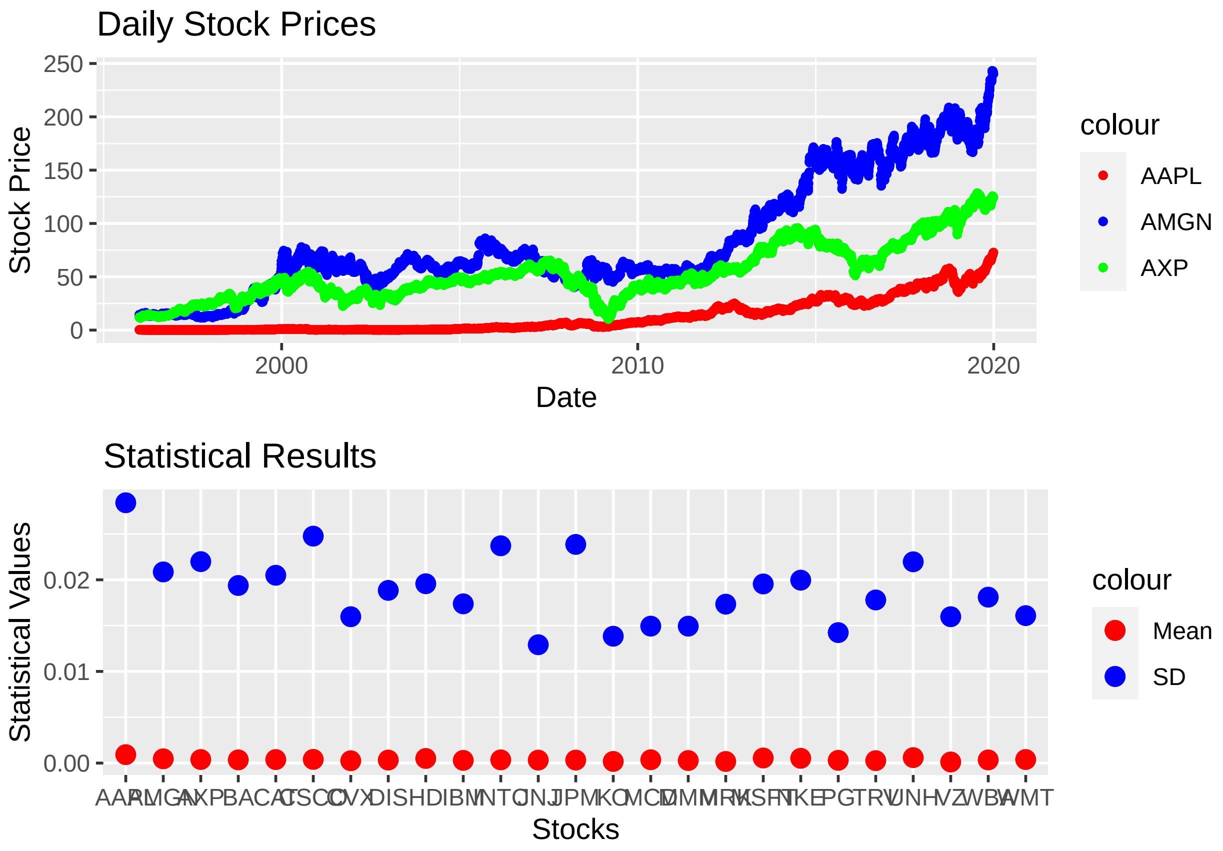
```
##
## Caricamento pacchetto: 'dplyr'

## Il seguente oggetto è mascherato da 'package:gridExtra':
##
##     combine

## I seguenti oggetti sono mascherati da 'package:stats':
##
##     filter, lag

## I seguenti oggetti sono mascherati da 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(magrittr)
log_returns <- data %>%
  select(-1) %>%
  mutate(across(everything(), ~log(. / lag(.)))) %>%
  slice(-1) %>%
  bind_cols(date = data[-1, 1], .) %>%
```

```r
  setNames(names(data))

stat_fun <- function(x) c(mean = mean(x), sd = sd(x))

statistics <- log_returns[,-1] %>%
  summarise(across(everything(), stat_fun))

rownames(statistics) <- c("mean", "sd")

head(log_returns)
```

```
##            X        AAPL        AMGN          AXP           BA          CAT
## 1 1996-01-03  0.00000000 -0.01078759  0.000000000 -0.009434032  0.016667052
## 2 1996-01-04 -0.01766372 -0.04435317 -0.009077177 -0.025601398 -0.008298803
## 3 1996-01-05  0.08171839  0.02242246 -0.003044156  0.017671143  0.016529302
## 4 1996-01-08  0.01088869 -0.01676954  0.000000000  0.022048137 -0.004106782
## 5 1996-01-09 -0.05567272 -0.02396007 -0.015361271 -0.011160162 -0.027113235
## 6 1996-01-10  0.04478403 -0.02573241 -0.034649121 -0.033433214 -0.010627093
##           CSCO          CVX          DIS           HD          IBM         INTC
## 1 -0.040071981  0.002361276  0.006141243 -0.018642404 -0.018043515 -0.015037877
## 2 -0.038199144  0.027908788 -0.012320435 -0.002691813 -0.026971117 -0.004338402
## 3  0.012313233  0.015927527  0.018424292 -0.024557852  0.019943681  0.000000000
## 4 -0.009661798  0.004504512  0.006066728 -0.028012958  0.005625894  0.002171554
## 5 -0.079895795  0.003924872  0.001510949 -0.011428684 -0.027009460 -0.046623316
## 6  0.020814407 -0.052265699 -0.040570375  0.011428684  0.005747142 -0.016036999
##            JNJ          JPM           KO          MCD          MMM          MRK
## 1  0.039277776  0.000000000  0.008298803  0.000000000 -0.001846723 -0.013712262
## 2 -0.010035927 -0.043485146 -0.008298803  0.005479466 -0.003703708  0.005899722
## 3  0.000000000 -0.004454386 -0.005012542 -0.016529302  0.001853569 -0.011834458
## 4  0.012884931  0.000000000  0.006677821  0.002773927 -0.007434978  0.013793322
## 5  0.005673774 -0.024859965 -0.013400536 -0.016760169 -0.014564505 -0.005888143
## 6 -0.033072748 -0.009195541 -0.017007213 -0.017045867 -0.008077972 -0.036076056
##           MSFT          NKE           PG          TRV          UNH           VZ
## 1 -0.032557631 -0.030332500  0.029631798  0.013187004  0.003861009  0.030883522
## 2  0.005738896 -0.010929071  0.015930822  0.013015368  0.009587801 -0.018051105
## 3 -0.011510917 -0.031630423  0.004301082  0.002152853 -0.040901514  0.010869672
## 4 -0.001448319  0.016870007  0.004282662  0.004291852 -0.020080996 -0.007233283
## 5 -0.072882364 -0.020657890 -0.001425517 -0.008602204 -0.058471768 -0.001134952
## 6  0.026914398 -0.003802285 -0.012921931 -0.010857870 -0.002152853 -0.007980903
##            WBA         WMT
## 1 -0.004158010  0.01069529
## 2 -0.012578782  0.01058211
## 3  0.037271395 -0.01591546
## 4 -0.008163311  0.00000000
## 5 -0.008230499 -0.01617286
## 6 -0.008298803 -0.01643873
```

```r
statistics<-as.matrix(statistics)

data$X<-as.Date(data$X)

plot1 <- ggplot(data, aes(x =X)) +
  geom_point(aes(y = AAPL, color = "AAPL"), size = 1) +
  geom_point(aes(y = AMGN, color = "AMGN"), size = 1) +
  geom_point(aes(y = AXP, color = "AXP"), size = 1) +
  labs(title = "Daily Stock Prices",
       x = "Date",
       y = "Stock Price") +
  scale_color_manual(values = c("AAPL" = "red", "AMGN" = "blue", "AXP" = "green"))

plot2 <- ggplot() +
  geom_point(aes(x = colnames(statistics), y = statistics[1,],color="Mean"), size = 3)+
  geom_point(aes(x = colnames(statistics), y = statistics[2,],color="SD"), size = 3)+
  labs(title = "Statistical Results",
       x = "Stocks",
       y = "Statistical Values") +
  scale_color_manual(values = c("Mean" = "red", "SD" = "blue"))

library(gridExtra)
combined_plot <- grid.arrange(plot1, plot2, ncol = 1)
```



```
##Parte 2

library(quantmod)
```

```
## Caricamento del pacchetto richiesto: xts
```

```
## Caricamento del pacchetto richiesto: zoo
```

```
##
## Caricamento pacchetto: ’zoo’

## I seguenti oggetti sono mascherati da ’package:base’:
##
##     as.Date, as.Date.numeric

##
## ######################### Warning from ’xts’ package ##########################
## #                                                                             #
## # The dplyr lag() function breaks how base R’s lag() function is supposed to  #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or       #
## # source() into this session won’t work correctly.                            #
## #                                                                             #
## # Use stats::lag() to make sure you’re not using dplyr::lag(), or you can add #
## # conflictRules(’dplyr’, exclude = ’lag’) to your .Rprofile to stop           #
## # dplyr from breaking base R’s lag() function.                                #
## #                                                                             #
## # Code in packages is not affected. It’s protected by R’s namespace mechanism #
## # Set ‘options(xts.warn_dplyr_breaks_lag = FALSE)‘ to suppress this warning.  #
## #                                                                             #
## ###############################################################################

##
## Caricamento pacchetto: ’xts’

## I seguenti oggetti sono mascherati da ’package:dplyr’:
##
##     first, last

## Caricamento del pacchetto richiesto: TTR

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

```
AMZN=get(getSymbols("AMZN", from = "2021-01-01", to = "2021-12-31"))
file_name ="AMZN.csv"
write.csv(AMZN, file = file_name, row.names = FALSE)
file.exists(file_name)
```

```
## [1] TRUE
```

```
head(AMZN)
```

```
##            AMZN.Open AMZN.High AMZN.Low AMZN.Close AMZN.Volume AMZN.Adjusted
## 2021-01-04  163.5000  163.6000  157.201   159.3315    88228000      159.3315
## 2021-01-05  158.3005  161.1690  158.253   160.9255    53110000      160.9255
## 2021-01-06  157.3240  159.8755  156.558   156.9190    87896000      156.9190
## 2021-01-07  157.8500  160.4270  157.750   158.1080    70290000      158.1080
## 2021-01-08  159.0000  159.5320  157.110   159.1350    70754000      159.1350
## 2021-01-11  157.4005  157.8190  155.500   155.7105    73668000      155.7105
```

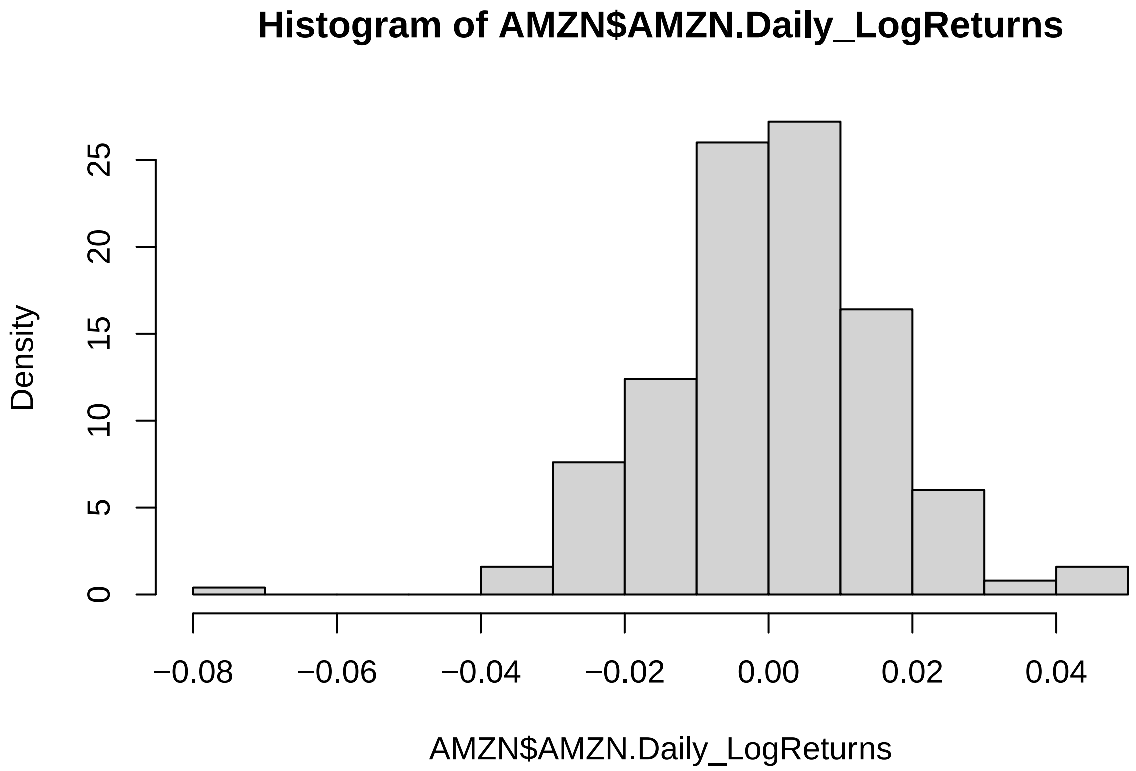
```
AMZN_weekly<-to.weekly(AMZN)
AMZN_weekly$AMZN.Weekly_LogReturns<-log(AMZN_weekly$AMZN.Adjusted/lag(AMZN_weekly$AMZN.Adjusted))
head(AMZN_weekly)
```

```
##            AMZN.Open AMZN.High AMZN.Low AMZN.Close AMZN.Volume AMZN.Adjusted
## 2021-01-08  163.5000  163.6000 156.5580   159.1350   370278000      159.1350
## 2021-01-15  157.4005  159.4975 154.3000   155.2125   356682000      155.2125
## 2021-01-22  155.3500  167.4275 154.8000   164.6115   327458000      164.6115
## 2021-01-29  166.4250  168.1945 159.2275   160.3100   376160000      160.3100
## 2021-02-05  162.1180  171.7000 161.7515   167.6075   512782000      167.6075
## 2021-02-12  167.9250  168.2500 161.6655   163.8855   264984000      163.8855
##            AMZN.Weekly_LogReturns
## 2021-01-08                     NA
## 2021-01-15            -0.02495776
## 2021-01-22             0.05879302
## 2021-01-29            -0.02647870
## 2021-02-05             0.04451550
## 2021-02-12            -0.02245692
```

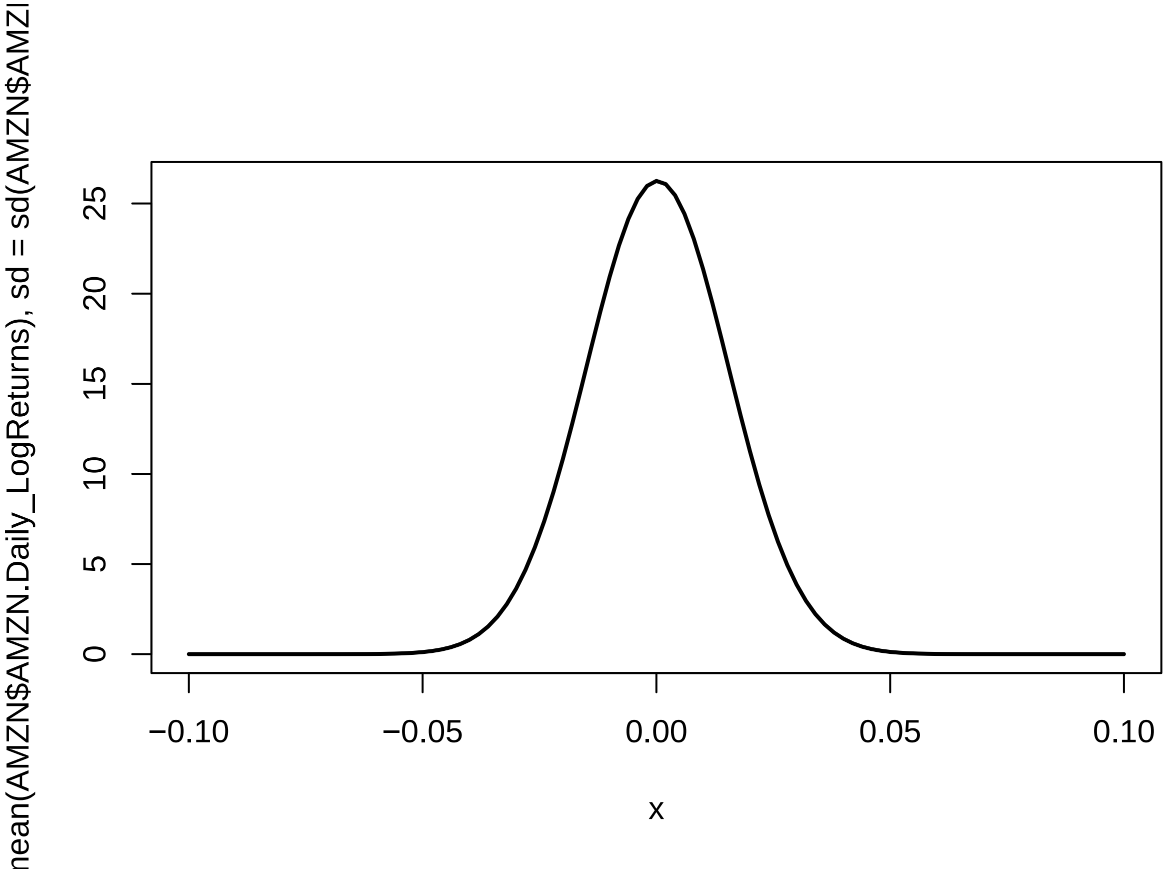
```
AMZN_weekly<-na.omit(AMZN_weekly)

stat<-function(x){
  c(mean(x),median(x),sd(x))
}

statistics<-sapply(AMZN_weekly$AMZN.Weekly_LogReturns,stat)
rownames(statistics)<-c("mean","median","sd")
statistics
```

```
##        AMZN.Weekly_LogReturns
## mean              0.001138041
## median           -0.002562139
## sd                0.033926217
```

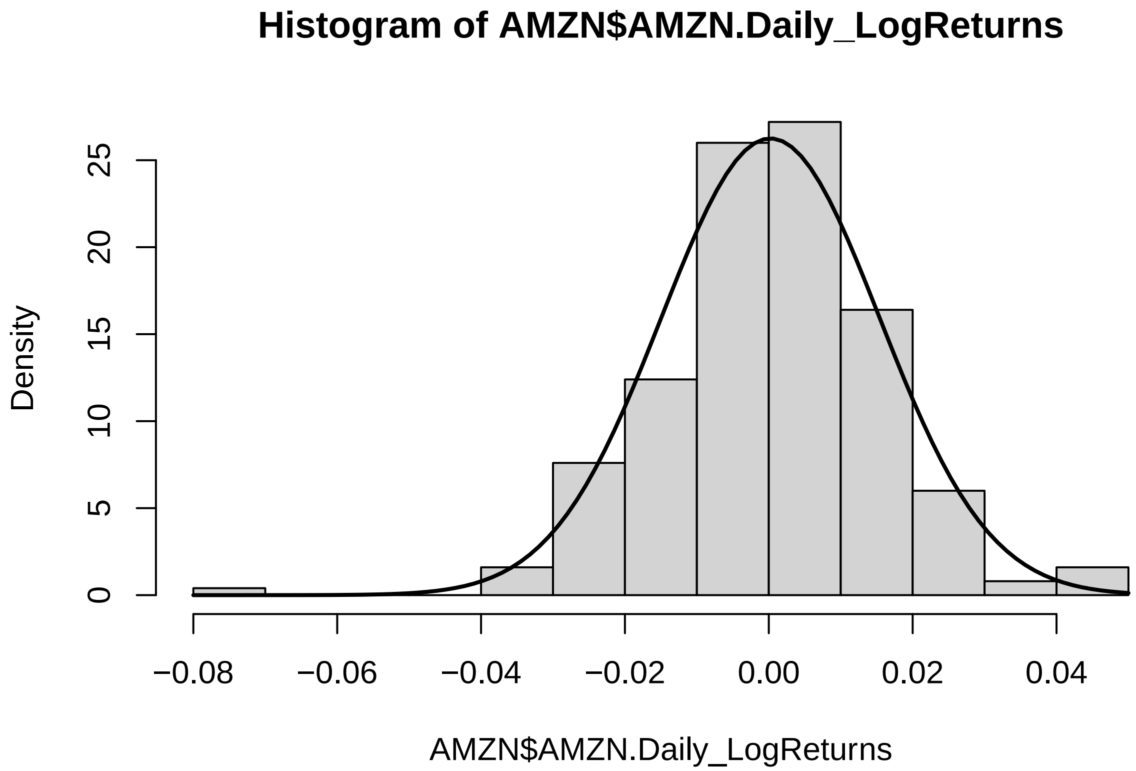
```
AMZN$AMZN.Daily_LogReturns<-log(AMZN$AMZN.Adjusted/lag(AMZN$AMZN.Adjusted))
AMZN<-na.omit(AMZN)

hist(AMZN$AMZN.Daily_LogReturns, freq=F)
```



```
x<-1:6
curve(dnorm(x, mean=mean(AMZN$AMZN.Daily_LogReturns), sd=sd(AMZN$AMZN.Daily_LogReturns)),-0.1,0.1 ,add=
```



```
{hist(AMZN$AMZN.Daily_LogReturns, freq=FALSE)
curve(dnorm(x, mean=mean(AMZN$AMZN.Daily_LogReturns), sd=sd(AMZN$AMZN.Daily_LogReturns)), add=T, lwd=2)
```



```
nrow(subset(AMZN,AMZN.Daily_LogReturns>0.01 & AMZN$AMZN.Daily_LogReturns<0.015))
```

```
## [1] 31
```