ECON 432 Homework 1

Madelyn Caufield

Jan 20,2021

Contents

```r
getwd()
```

```
## [1] "/Users/madelyncaufield/Desktop/ECON432 Data Science for Financial Engineering/Homework 1"
```

```r
setwd("/Users/madelyncaufield/Desktop/ECON432 Data Science for Financial Engineering/Homework 1")
getwd()
```

```
## [1] "/Users/madelyncaufield/Desktop/ECON432 Data Science for Financial Engineering/Homework 1"
```

```r
df <- read.csv("sbuxPrices.csv")
head(df)
```

```
##         Date     Open     High      Low    Close Adj.Close    Volume
## 1 1998-03-01 2.472656 2.902344 2.382813 2.832031  2.367074 297006400
## 2 1998-04-01 2.835938 3.062500 2.656250 3.007813  2.513996 330811200
## 3 1998-05-01 3.015625 3.117188 2.781250 3.000000  2.507466 233537600
## 4 1998-06-01 2.976563 3.421875 2.910156 3.339844  2.791515 362950400
## 5 1998-07-01 3.351563 3.746094 2.601563 2.617188  2.187503 594307200
## 6 1998-08-01 2.636719 2.703125 1.929688 1.972656  1.648790 652120000
```

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

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

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

sbux.df = df %>%
          select(Date, Adj.Close)
```

Part II

Question 1

```
head(sbux.df)
```

```
##         Date Adj.Close
## 1 1998-03-01  2.367074
## 2 1998-04-01  2.513996
## 3 1998-05-01  2.507466
## 4 1998-06-01  2.791515
## 5 1998-07-01  2.187503
## 6 1998-08-01  1.648790
```

This gives us the first 6 rows of the data.

```
tail(sbux.df)
```

```
##           Date Adj.Close
## 176 2012-10-01  19.94681
## 177 2012-11-01  22.54120
## 178 2012-12-01  23.40302
## 179 2013-01-01  24.48960
## 180 2013-02-01  23.93541
## 181 2013-03-01  24.94520
```

This gives us the last 6 rows of the data.

```
sbuxPrices.df = sbux.df[, "Adj.Close", drop=FALSE]
sbuxPrices.df
```

```
##    Adj.Close
## 1   2.367074
## 2   2.513996
## 3   2.507466
## 4   2.791515
## 5   2.187503
## 6   1.648790
## 7   1.890395
## 8   2.265861
## 9   2.409518
## 10  2.931906
```

```
## 11   2.719687
## 12   2.762131
## 13   2.931906
## 14   3.859147
## 15   3.852618
## 16   3.924447
## 17   2.429108
## 18   2.389929
## 19   2.589089
## 20   2.840489
## 21   2.775190
## 22   2.533586
## 23   3.343288
## 24   3.669783
## 25   4.681910
## 26   3.158820
## 27   3.552243
## 28   3.989744
## 29   3.917916
## 30   3.826499
## 31   4.185639
## 32   4.668849
## 33   4.760267
## 34   4.623142
## 35   5.217360
## 36   4.975753
## 37   4.433774
## 38   4.043289
## 39   4.078811
## 40   4.805977
## 41   3.769558
## 42   3.525080
## 43   3.121796
## 44   3.577317
## 45   3.702692
## 46   3.980602
## 47   4.966872
## 48   4.808066
## 49   4.833141
## 50   4.768364
## 51   5.073440
## 52   5.192544
## 53   4.101796
## 54   4.200006
## 55   4.317021
## 56   4.981499
## 57   4.542691
## 58   4.258514
## 59   4.747469
## 60   4.900007
## 61   5.382694
## 62   4.912546
## 63   5.154932
## 64   5.129857
```

```
## 65    5.710755
## 66    5.932247
## 67    6.017919
## 68    6.602994
## 69    6.722100
## 70    6.928964
## 71    7.649861
## 72    7.814933
## 73    7.913146
## 74    8.132547
## 75    8.483591
## 76    9.087473
## 77    9.818820
## 78    9.035233
## 79    9.499116
## 80   11.049570
## 81   11.755837
## 82   13.030468
## 83   11.283600
## 84   10.825988
## 85   10.794641
## 86   10.347478
## 87   11.448672
## 88   10.794641
## 89   10.980611
## 90   10.245090
## 91   10.468672
## 92   11.818524
## 93   12.725391
## 94   12.541508
## 95   13.247780
## 96   15.178531
## 97   15.725989
## 98   15.575548
## 99   14.898525
## 100  15.780325
## 101  14.305095
## 102  12.959421
## 103  14.229874
## 104  15.776139
## 105  14.748080
## 106  14.802410
## 107  14.601812
## 108  12.913448
## 109  13.105693
## 110  12.963603
## 111  12.040015
## 112  10.965987
## 113  11.149868
## 114  11.513449
## 115  10.949268
## 116  11.149868
## 117   9.774938
## 118   8.554641
```

```
## 119  7.902697
## 120  7.514042
## 121  7.313445
## 122  6.782696
## 123  7.601803
## 124  6.577919
## 125  6.139113
## 126  6.502696
## 127  6.214338
## 128  5.487173
## 129  3.731944
## 130  3.953439
## 131  3.945081
## 132  3.823885
## 133  4.642992
## 134  6.042995
## 135  6.013739
## 136  5.804784
## 137  7.397027
## 138  7.936127
## 139  8.629864
## 140  7.931952
## 141  9.152253
## 142  9.637027
## 143  9.106283
## 144  9.574342
## 145 10.142697
## 146 10.857329
## 147 10.864536
## 148 10.197306
## 149 10.428110
## 150  9.643379
## 151 10.778244
## 152 12.048012
## 153 12.908585
## 154 13.611482
## 155 13.357297
## 156 13.971573
## 157 15.716125
## 158 15.397117
## 159 15.648073
## 160 16.856457
## 161 17.112574
## 162 16.485100
## 163 15.973932
## 164 18.145773
## 165 18.625546
## 166 19.786406
## 167 20.607794
## 168 20.883020
## 169 24.120111
## 170 24.758835
## 171 23.688551
## 172 23.081621
```

```
## 173 19.601198
## 174 21.475607
## 175 22.037100
## 176 19.946812
## 177 22.541204
## 178 23.403025
## 179 24.489603
## 180 23.935408
## 181 24.945196
```

This gives us just the column for adjusted closing prices and drops the column Date column to make a new data frame called sbuxPrices.df.

```
rownames(sbuxPrices.df) = sbux.df$Date
rownames(sbuxPrices.df)
```

```
##   [1] "1998-03-01" "1998-04-01" "1998-05-01" "1998-06-01" "1998-07-01"
##   [6] "1998-08-01" "1998-09-01" "1998-10-01" "1998-11-01" "1998-12-01"
##  [11] "1999-01-01" "1999-02-01" "1999-03-01" "1999-04-01" "1999-05-01"
##  [16] "1999-06-01" "1999-07-01" "1999-08-01" "1999-09-01" "1999-10-01"
##  [21] "1999-11-01" "1999-12-01" "2000-01-01" "2000-02-01" "2000-03-01"
##  [26] "2000-04-01" "2000-05-01" "2000-06-01" "2000-07-01" "2000-08-01"
##  [31] "2000-09-01" "2000-10-01" "2000-11-01" "2000-12-01" "2001-01-01"
##  [36] "2001-02-01" "2001-03-01" "2001-04-01" "2001-05-01" "2001-06-01"
##  [41] "2001-07-01" "2001-08-01" "2001-09-01" "2001-10-01" "2001-11-01"
##  [46] "2001-12-01" "2002-01-01" "2002-02-01" "2002-03-01" "2002-04-01"
##  [51] "2002-05-01" "2002-06-01" "2002-07-01" "2002-08-01" "2002-09-01"
##  [56] "2002-10-01" "2002-11-01" "2002-12-01" "2003-01-01" "2003-02-01"
##  [61] "2003-03-01" "2003-04-01" "2003-05-01" "2003-06-01" "2003-07-01"
##  [66] "2003-08-01" "2003-09-01" "2003-10-01" "2003-11-01" "2003-12-01"
##  [71] "2004-01-01" "2004-02-01" "2004-03-01" "2004-04-01" "2004-05-01"
##  [76] "2004-06-01" "2004-07-01" "2004-08-01" "2004-09-01" "2004-10-01"
##  [81] "2004-11-01" "2004-12-01" "2005-01-01" "2005-02-01" "2005-03-01"
##  [86] "2005-04-01" "2005-05-01" "2005-06-01" "2005-07-01" "2005-08-01"
##  [91] "2005-09-01" "2005-10-01" "2005-11-01" "2005-12-01" "2006-01-01"
##  [96] "2006-02-01" "2006-03-01" "2006-04-01" "2006-05-01" "2006-06-01"
## [101] "2006-07-01" "2006-08-01" "2006-09-01" "2006-10-01" "2006-11-01"
## [106] "2006-12-01" "2007-01-01" "2007-02-01" "2007-03-01" "2007-04-01"
## [111] "2007-05-01" "2007-06-01" "2007-07-01" "2007-08-01" "2007-09-01"
## [116] "2007-10-01" "2007-11-01" "2007-12-01" "2008-01-01" "2008-02-01"
## [121] "2008-03-01" "2008-04-01" "2008-05-01" "2008-06-01" "2008-07-01"
## [126] "2008-08-01" "2008-09-01" "2008-10-01" "2008-11-01" "2008-12-01"
## [131] "2009-01-01" "2009-02-01" "2009-03-01" "2009-04-01" "2009-05-01"
## [136] "2009-06-01" "2009-07-01" "2009-08-01" "2009-09-01" "2009-10-01"
## [141] "2009-11-01" "2009-12-01" "2010-01-01" "2010-02-01" "2010-03-01"
## [146] "2010-04-01" "2010-05-01" "2010-06-01" "2010-07-01" "2010-08-01"
## [151] "2010-09-01" "2010-10-01" "2010-11-01" "2010-12-01" "2011-01-01"
## [156] "2011-02-01" "2011-03-01" "2011-04-01" "2011-05-01" "2011-06-01"
## [161] "2011-07-01" "2011-08-01" "2011-09-01" "2011-10-01" "2011-11-01"
## [166] "2011-12-01" "2012-01-01" "2012-02-01" "2012-03-01" "2012-04-01"
## [171] "2012-05-01" "2012-06-01" "2012-07-01" "2012-08-01" "2012-09-01"
## [176] "2012-10-01" "2012-11-01" "2012-12-01" "2013-01-01" "2013-02-01"
## [181] "2013-03-01"
```

This converted the Date column to row names for the sbuxPrices.df data frame.

```
head(sbuxPrices.df)
```

```
##            Adj.Close
## 1998-03-01  2.367074
## 1998-04-01  2.513996
## 1998-05-01  2.507466
## 1998-06-01  2.791515
## 1998-07-01  2.187503
## 1998-08-01  1.648790
```

This gave us the first 6 rows of the adjusted closing price column with the new row names (dates from above) from the sbuxPrices.df data frame.

```
sbux.df[101:132,]
```

```
##           Date Adj.Close
## 101 2006-07-01 14.305095
## 102 2006-08-01 12.959421
## 103 2006-09-01 14.229874
## 104 2006-10-01 15.776139
## 105 2006-11-01 14.748080
## 106 2006-12-01 14.802410
## 107 2007-01-01 14.601812
## 108 2007-02-01 12.913448
## 109 2007-03-01 13.105693
## 110 2007-04-01 12.963603
## 111 2007-05-01 12.040015
## 112 2007-06-01 10.965987
## 113 2007-07-01 11.149868
## 114 2007-08-01 11.513449
## 115 2007-09-01 10.949268
## 116 2007-10-01 11.149868
## 117 2007-11-01  9.774938
## 118 2007-12-01  8.554641
## 119 2008-01-01  7.902697
## 120 2008-02-01  7.514042
## 121 2008-03-01  7.313445
## 122 2008-04-01  6.782696
## 123 2008-05-01  7.601803
## 124 2008-06-01  6.577919
## 125 2008-07-01  6.139113
## 126 2008-08-01  6.502696
## 127 2008-09-01  6.214338
## 128 2008-10-01  5.487173
## 129 2008-11-01  3.731944
## 130 2008-12-01  3.953439
## 131 2009-01-01  3.945081
## 132 2009-02-01  3.823885
```

This gave us only rows 101-132 for the sbux.df data frame.

Question 2

```
class(sbux.df$Date)
```

```
## [1] "character"
```

```
sbux.df$Date <- as.Date(sbux.df$Date,format = '%Y-%m-%d')
class(sbux.df$Date)
```

```
## [1] "Date"
```

```
plot(Adj.Close~Date, sbux.df, type='l',col="blue",xlab = 'Date',ylab='Starbucks Prices',
     main='Monthly Prices of Starbucks Shares')
T <- length(sbux.df$Date)
ticks <- seq(sbux.df$Date[1], sbux.df$Date[T], by = "quarters")
axis(1, at = ticks, labels = FALSE)
legend('topleft', names(sbuxPrices.df), col= "blue", lty=1, cex=.65)
```


Monthly Prices of Starbucks Shares
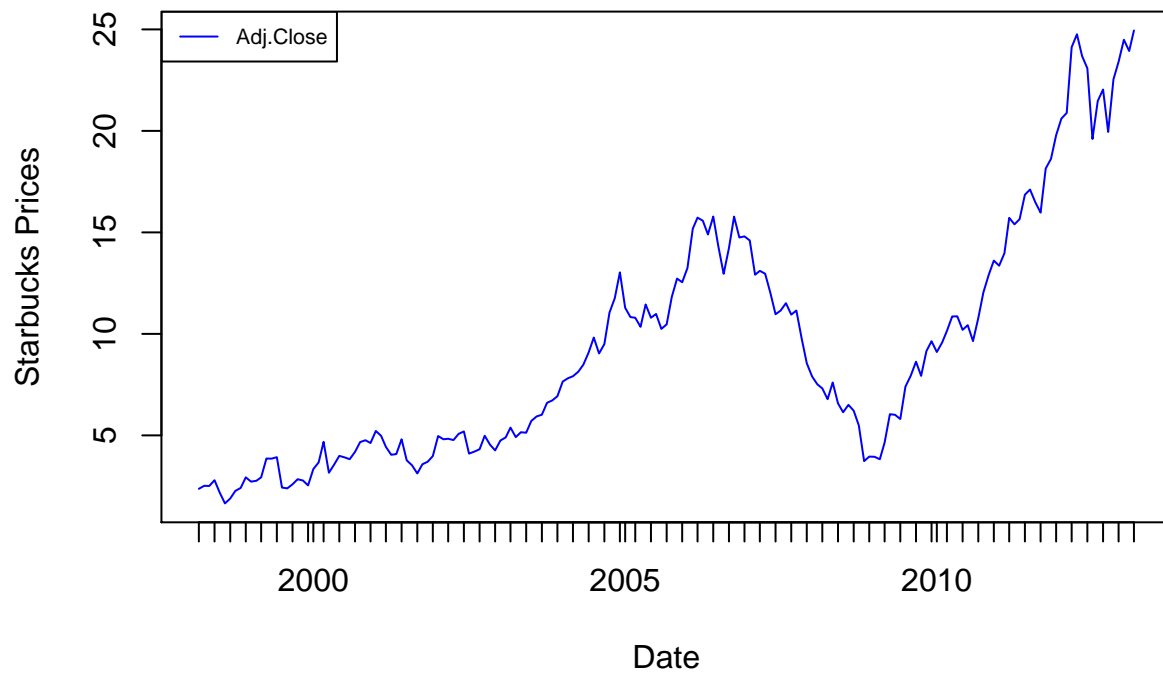
Question 3

Monthly Simple Return

```
sbux.df$ret[2:T] <-  sbux.df$Adj.Close[2:T]/sbux.df$Adj.Close[1:(T-1)] - 1
head(sbux.df$ret)
```

```
## [1]           NA  0.062069035 -0.002597458  0.113281297 -0.216374263
## [6] -0.246268462
```

```
library(ggplot2)

df.g <- sbux.df[,c('Date','ret')]
df.g <- df.g[-c(1),]

head(df.g)
```
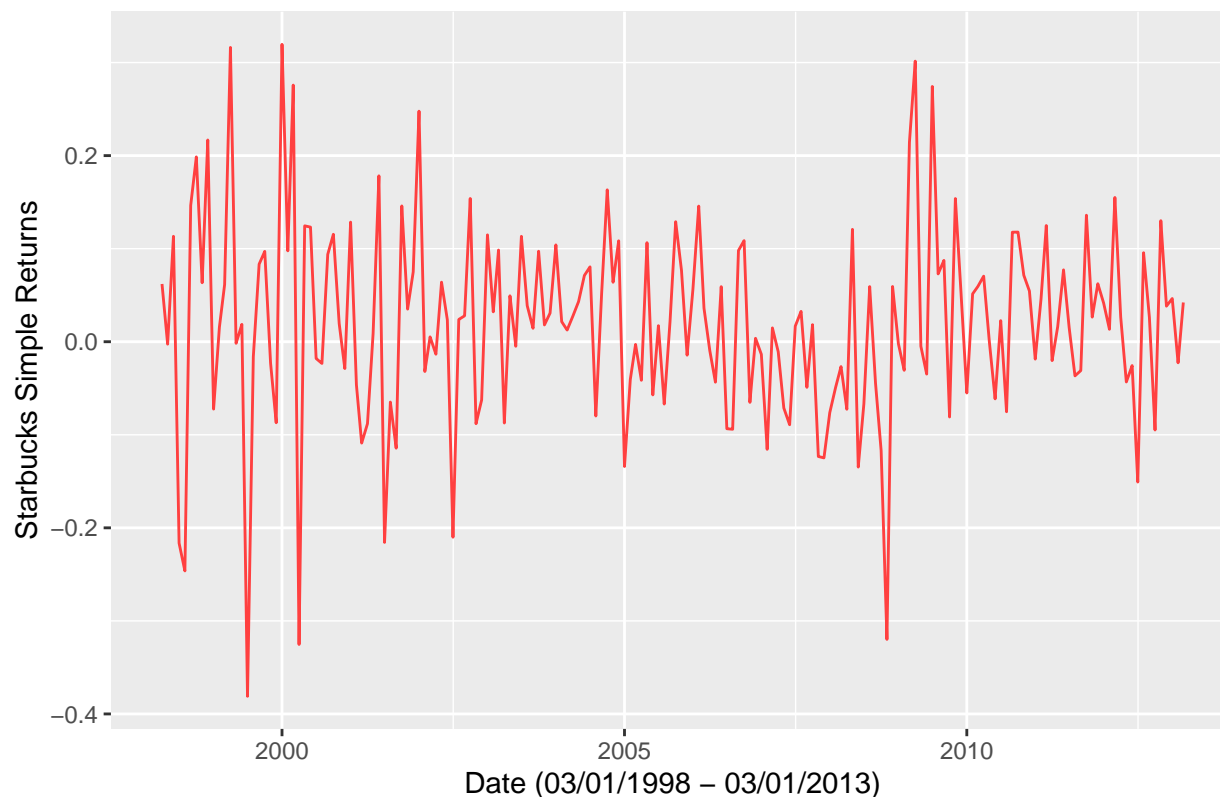
```
##         Date          ret
## 2 1998-04-01  0.062069035
## 3 1998-05-01 -0.002597458
## 4 1998-06-01  0.113281297
## 5 1998-07-01 -0.216374263
## 6 1998-08-01 -0.246268462
## 7 1998-09-01  0.146534732
```

Monthly Simple Return Plot

```
g <- ggplot(df.g, aes(x = Date, y = ret)) +
  geom_line(color="brown1") +
  ylab("Starbucks Simple Returns") +
  xlab("Date (03/01/1998 - 03/01/2013)") +
  ggtitle("Monthly Simple Returns of Starbucks Shares")
g
```

Monthly Simple Returns of Starbucks Shares

Monthly Coninuously Compounded Return

```
sbux.df$cc.ret[2:T] <-  log(sbux.df$Adj.Close[2:T]/sbux.df$Adj.Close[1:(T-1)])
head(sbux.df$cc.ret)
```

```
## [1]          NA  0.060218926 -0.002600838  0.107311778 -0.243823749
## [6] -0.282719025
```

```
df.g$cc.ret <- sbux.df$cc.ret[2:T]
head(df.g)
```

```
##        Date          ret        cc.ret
## 2 1998-04-01  0.062069035  0.060218926
## 3 1998-05-01 -0.002597458 -0.002600838
## 4 1998-06-01  0.113281297  0.107311778
## 5 1998-07-01 -0.216374263 -0.243823749
## 6 1998-08-01 -0.246268462 -0.282719025
## 7 1998-09-01  0.146534732  0.136744116
```

Monthly Continuously Compounded Return Plot

```
g <- ggplot(df.g, aes(x = Date, y = cc.ret)) +
  geom_line(color="lightseagreen") +
  ylab("Starbucks Continuously Compounded Returns") +
  xlab("Date (03/01/1998 - 03/01/2013)") +
  ggtitle("Monthly Continuously Compounded Returns of Starbucks Shares")
g
```


Monthly Continuously Compounded Returns of Starbucks Shares
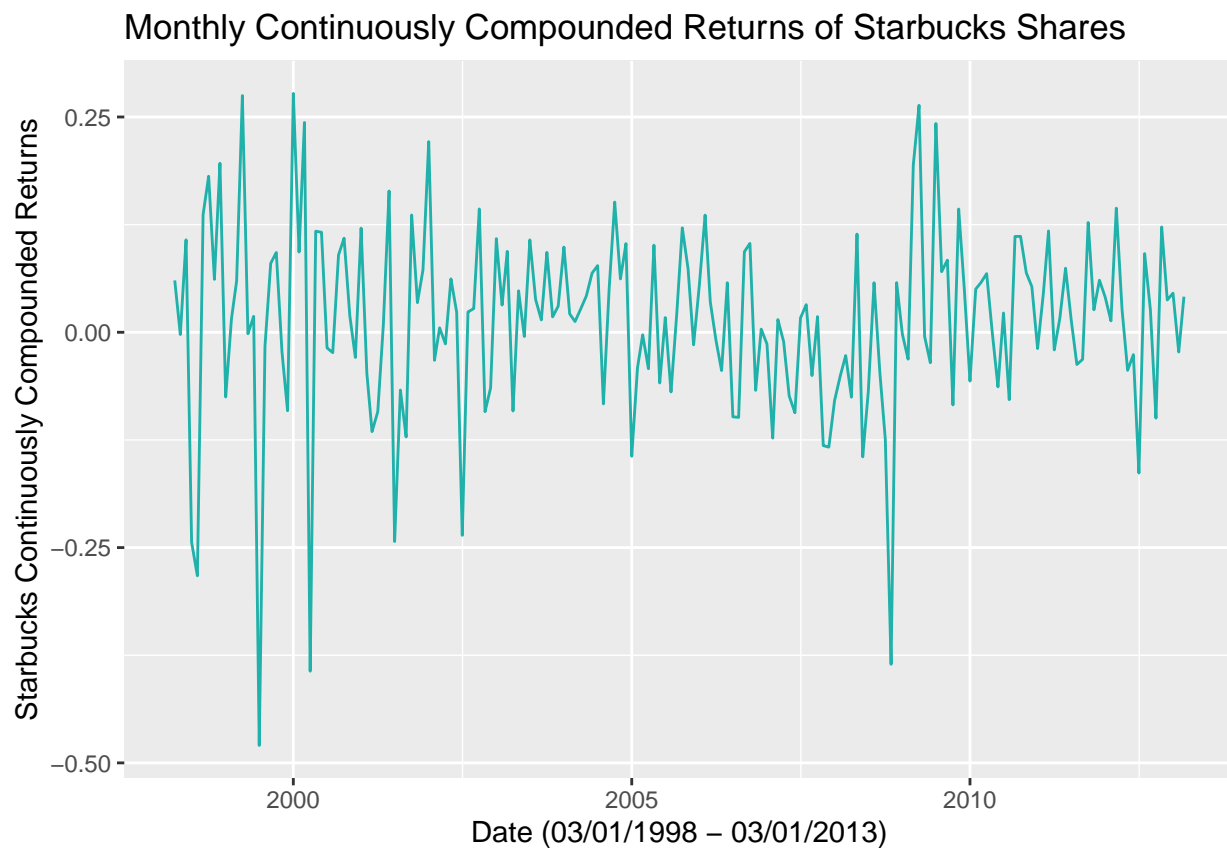
Plot of the Monthly Simple Return and Continuously Compounded Return

```
library(reshape2)
d <- melt(df.g, id.vars="Date")

ggplot(d, aes(Date, value, col=variable)) +
  geom_line() +
  ggtitle("Monthly Simple Returns and CC Returns of Starbucks Shares")
```

Monthly Simple Returns and CC Returns of Starbucks Shares