Assignment 5_MSF

Jason Zhongyun Zhang

9/17/2020

```r
library('tidyverse')
```

```
## -- Attaching packages ---------------

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.1
## v tidyr   1.1.1     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0

## -- Conflicts ------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
setwd('P:/Mgt_of_FI/Assignment5/')
msf <- read.csv('msf.csv')
```

```r
library("lubridate")
```

```
##
## Attaching package: 'lubridate'

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

```r
library("readxl")
library("zoo")
```

```
##
## Attaching package: 'zoo'

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

```r
colnames(msf)[1] <- "Date"
msf$Date <- msf$DATE %>% as.Date(format="%m/%d/%Y") %>% as.yearmon()
cpi <- read_excel(paste("P:/Mgt_of_FI/Assignment5/CPIAUCSL.xls"),skip = 10)
colnames(cpi)[1] <- "Date"
colnames(cpi)[2] <- "CPI"
cpi$Date <- cpi$Date %>% as.yearmon
```

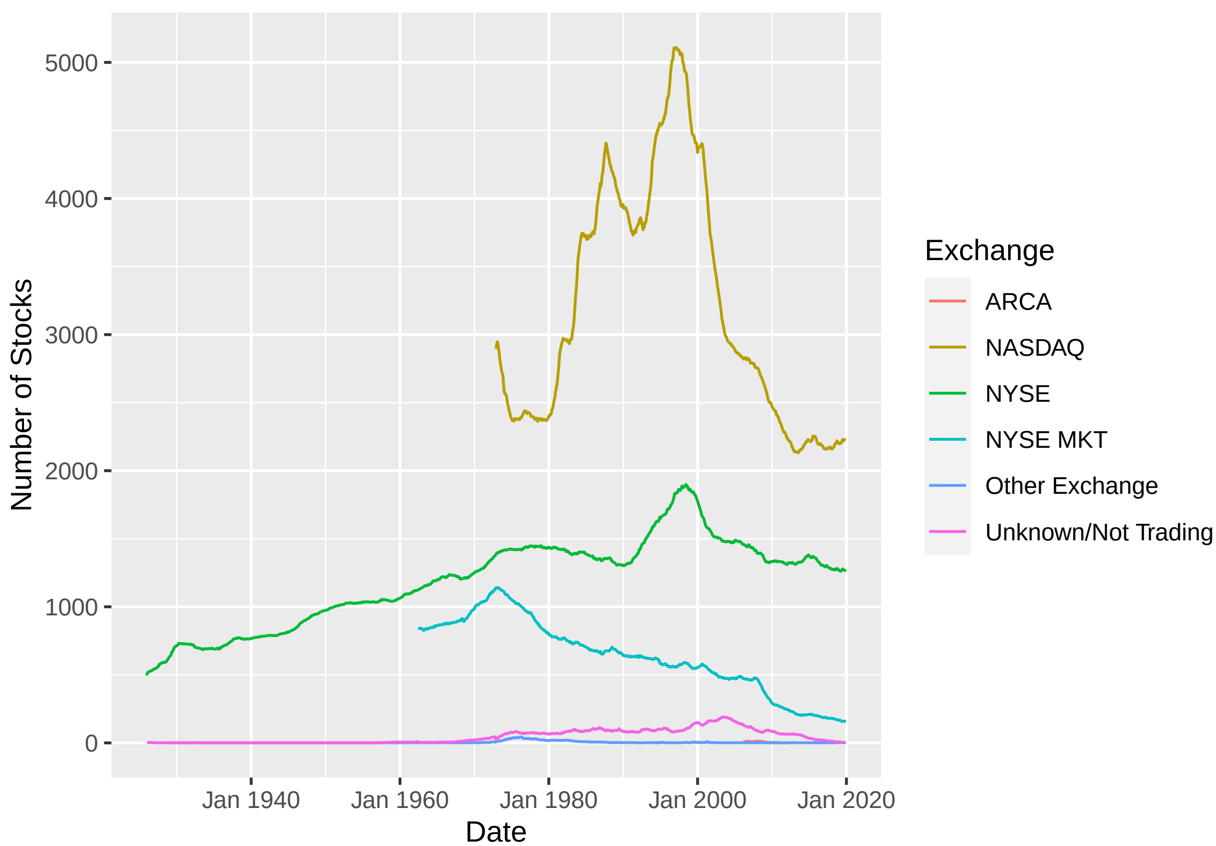
```r
msf <- msf %>% left_join(cpi, by = "Date")
```

```r
#adjust the price based on December 2015 CPI
cpi_dec_2015 <- msf$CPI[msf$Date=="Dec 2015"][1]
msf$price_adj <- msf$PRC/msf$CPI*cpi_dec_2015

msf$MKTCAP <- abs(msf$price_adj)*msf$SHROUT
```

```r
msf$Exchange <- ifelse(msf$EXCHCD==1 | msf$EXCHCD ==31, "NYSE",
                       ifelse(msf$EXCHCD==2 | msf$EXCHCD ==32, "NYSE MKT",
                       ifelse(msf$EXCHCD==3 | msf$EXCHCD ==33, "NASDAQ",
                       ifelse(msf$EXCHCD==4 | msf$EXCHCD ==34, "ARCA",
                       ifelse(msf$EXCHCD==0, "Unknown/Not Trading",
                              "Other Exchange")))))
```

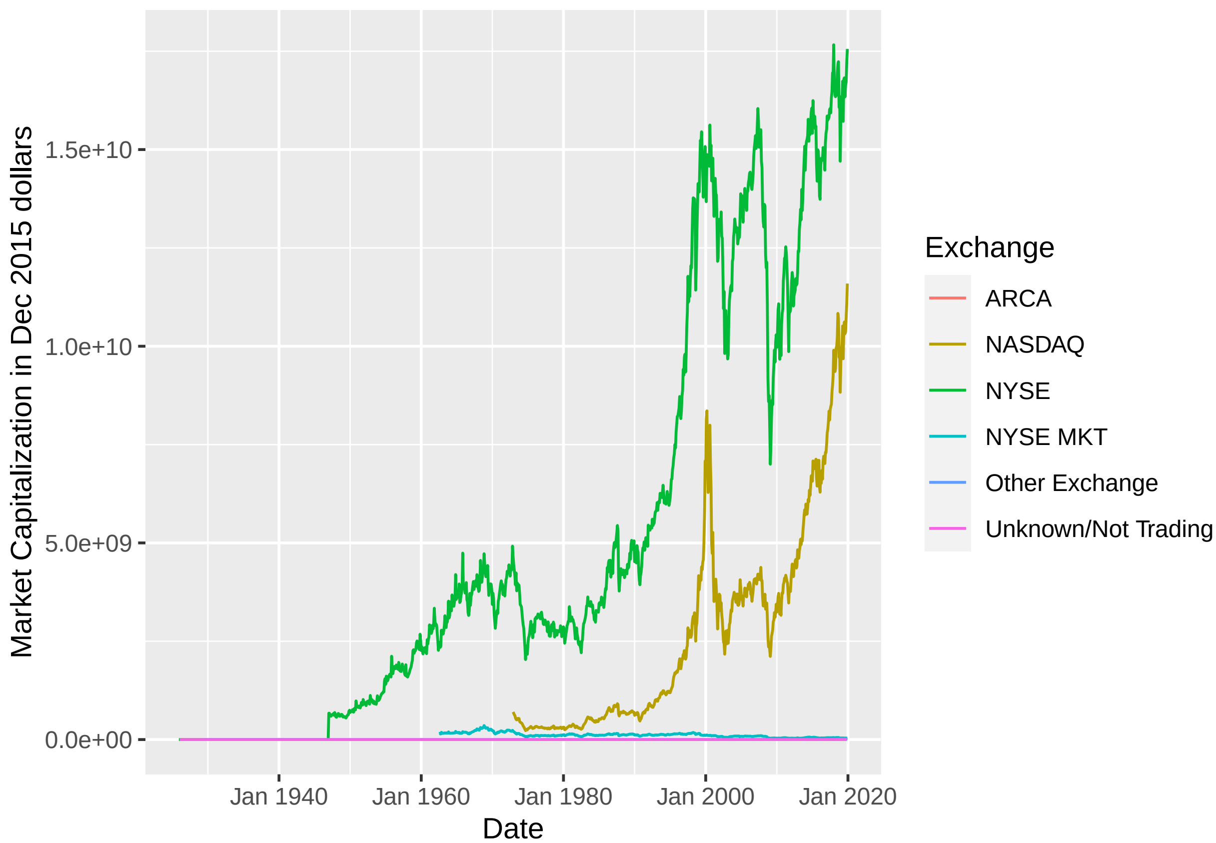
To plot the number of stocks in the sample by exchange

```r
msf %>% dplyr::filter (Date < 2020) %>% group_by(Date, Exchange) %>%
  summarize(n_stocks = n_distinct(PERMNO)) %>%
  ggplot(aes(x = Date, y = n_stocks, color = Exchange)) +
  geom_line()+ylab("Number of Stocks")
```



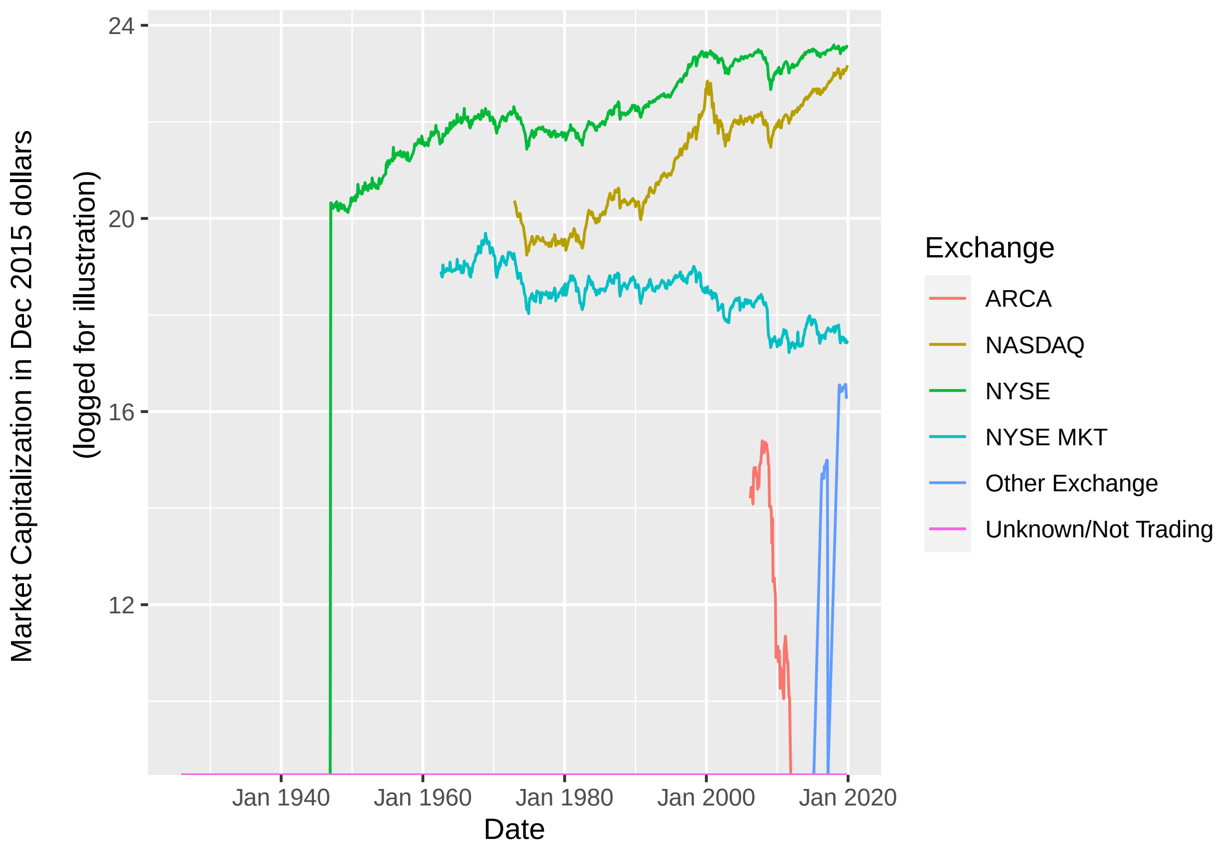
To plot market capitalization in Dec 2015 dollars by exchange

```r
msf %>% dplyr::filter (Date < 2020) %>% group_by(Date, Exchange) %>%
  summarize(summktcap = sum(MKTCAP, na.rm = T)) %>%
  ggplot(aes(x = Date, y = summktcap, color = Exchange)) +
  geom_line()+ylab("Market Capitalization in Dec 2015 dollars")
```



To plot market capitalization in Dec 2015 dollars by exchange, logged

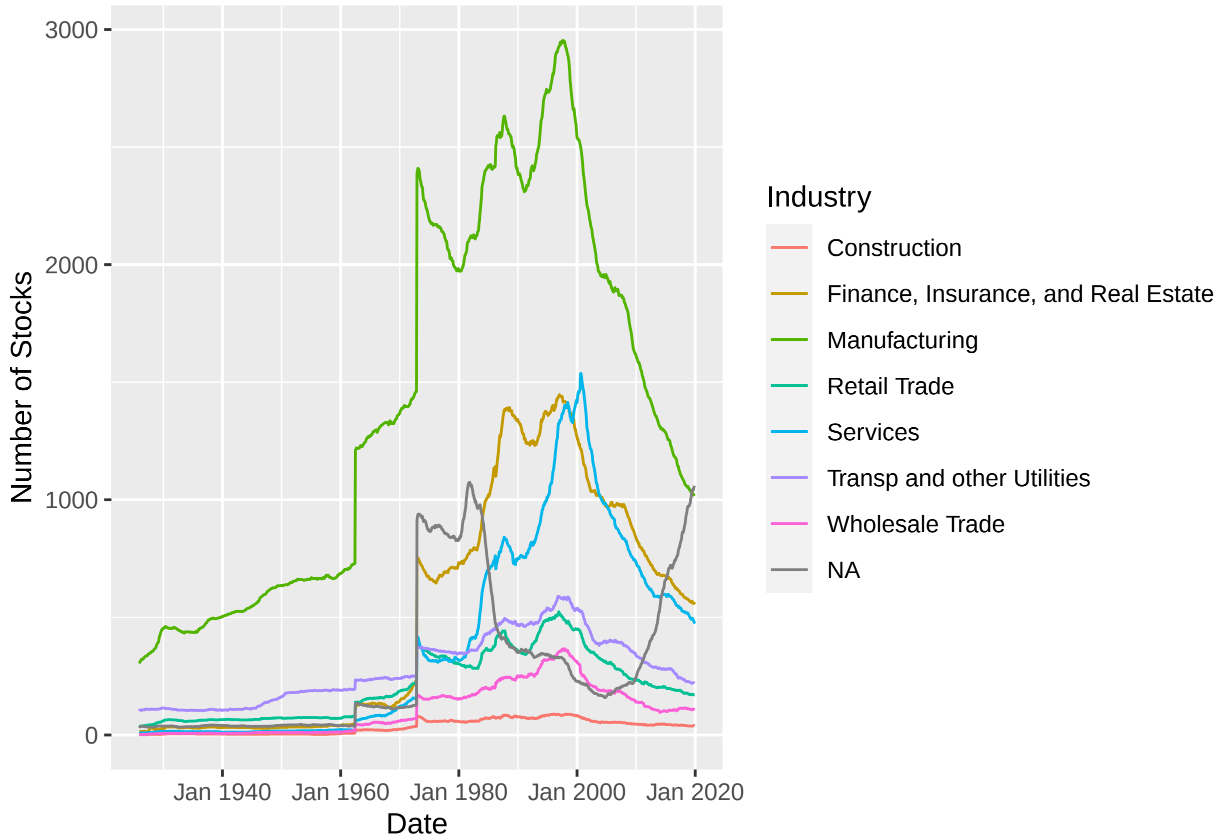
```r
msf %>% dplyr::filter (Date < 2020) %>% group_by(Date, Exchange) %>%
  summarize(summktcap = sum(MKTCAP, na.rm = T)) %>%
  ggplot(aes(x = Date, y = log(summktcap), color = Exchange)) +
  geom_line()+ylab("Market Capitalization in Dec 2015 dollars \n
                   (logged for illustration)")
```




```
#classify stocks into industries based on SIC Code
msf$Industry <- ifelse(msf$SICCD>0 & msf$SICCD < 1000, "Agr, Forestry, Fishing",
              ifelse(msf$SICCD>999 & msf$SICCD<1500, "Mining",
              ifelse(msf$SICCD>1499 & msf$SICCD<1800, "Construction",
              ifelse(msf$SICCD>1999 & msf$SICCD<4000, "Manufacturing",
              ifelse(msf$SICCD>3999 & msf$SICCD<5000, "Transp and other Utilities",
              ifelse(msf$SICCD>4999 & msf$SICCD<5200, "Wholesale Trade",
              ifelse(msf$SICCD>5199 & msf$SICCD<6000, "Retail Trade",
              ifelse(msf$SICCD>5999 & msf$SICCD<6800,
                     "Finance, Insurance, and Real Estate",
                     ifelse(msf$SICCD>6999 & msf$SICCD<9000, "Services",
                     ifelse(msf$SICCD>8999 & msf$SICCD<10000, "Public Adm", NA))))))))))
```

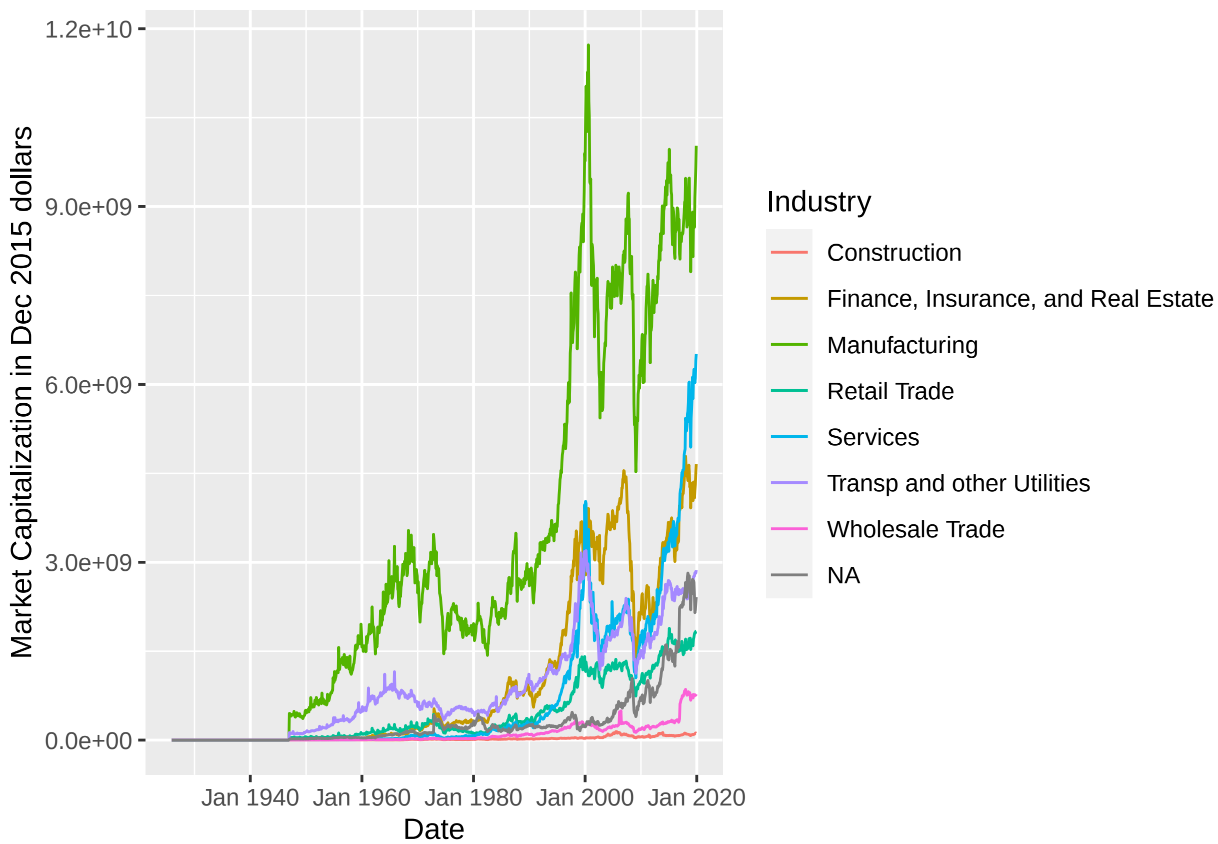
To plot the number of stocks by industry

```
msf %>% dplyr::filter (Date < 2020) %>% group_by(Date, Industry) %>%
  summarize(n_stocks = n_distinct(PERMNO)) %>%
  ggplot(aes(x = Date, y = n_stocks, color = Industry)) +
  geom_line()+ylab("Number of Stocks")
```



To plot market capitalization in Dec 2015 dollars by industry

```
msf %>% dplyr::filter (Date < 2020) %>% group_by(Date, Industry) %>%
  summarize(summktcap = sum(MKTCAP, na.rm = T)) %>%
  ggplot(aes(x = Date, y = summktcap, color = Industry)) +
  geom_line()+ylab("Market Capitalization in Dec 2015 dollars")
```




```
#library(devtools)
#install_github("antshi/ffData")
#library(ffData)
#devtools::install_github("sstoeckl/ffdownload")
```

```
#download the FF data and import
#ffDataDownload(type = "USResearch", number_factors = 3, freq = "m",
               #start = "192607", end = "202006")
setwd("P:/Mgt_of_FI/Assignment5/USResearch_m_192607_202006/")

FF <- read.csv('F-F_Research_Data_Factors.csv')

FF$Date <- FF$Date %>% as.character %>% paste("01") %>%  as.Date("%Y%m%d") %>% as.yearmon()
FF <- FF[c(1,5)]
```

To compute the excess return

```
msf <- msf %>% left_join(FF, by = "Date")
msf$RET <- msf$RET %>% as.numeric
```

```
## Warning in function_list[[k]](value): NAs introduced by coercion
```

```
msf$exc_return <- as.numeric(msf$RET)-msf$RF
```

Descriptive Stats

```r
library('fBasics')
```

```
## Loading required package: timeDate

## Loading required package: timeSeries

##
## Attaching package: 'timeSeries'

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

```r
#the function to print out descpritive stats
describe <- function(input) {
  desnames <- names(input)
  ds <- NA
  for (c in desnames) {
    coln <- as.name(c)
    output <- input %>% summarize(
                              n=n(),
                              s=skewness(eval(coln),na.rm = T),
                              k=kurtosis(eval(coln),na.rm = T),
                              m=mean(eval(coln), na.rm = T),
                              min=min(eval(coln), na.rm = T),
                              q1=quantile(eval(coln), probs=0.01, na.rm = T),
                              q5=quantile(eval(coln), probs=0.05, na.rm = T),
                              q25=quantile(eval(coln), probs=0.25, na.rm = T),
                              median=median(eval(coln), na.rm = T),
                              q75=quantile(eval(coln),probs=0.75, na.rm = T),
                              q95=quantile(eval(coln), probs=0.95, na.rm = T),
                              q99=quantile(eval(coln), probs=0.99, na.rm = T),
                              max=max(eval(coln), na.rm = T),
                              std=sd(eval(coln), na.rm = T))
    ds <-  rbind(ds, output)
  }
descriptive_table <- as.data.frame(ds[-1,])
rownames(descriptive_table) = desnames
colnames(descriptive_table) = c('N','Skewness', 'Kurtosis', 'Mean', 'Min',
                                '1%','5%', 'Q25', 'Q50', 'Q75', '95%', '99%',
                                'Max', 'SD')
print(descriptive_table)
}
```

To describe variables for 1925-2019

```r
msf %>% dplyr::filter (Date < "Jan 2020") %>% select(PRC, RET, SHROUT, VWRETD, CPI, price_adj,MKTCAP,RF
```

```
##                 N    Skewness     Kurtosis         Mean          Min
## PRC       3674766 162.56719925 30229.4672770 2.708847e+01 -1832.500000
## RET       3674766   6.76192652   356.4108025 1.190072e-02    -0.993600
```

```
## SHROUT     3674766  29.75842810  1727.0701136 4.059901e+04     0.000000
## VWRETD     3674766  -0.38208217     4.5040559 9.317659e-03    -0.291731
## CPI        3674766  -0.01671695    -1.0832518 1.328728e+02    21.480000
## price_adj  3674766 139.53333414 22341.8132160 5.794795e+01 -9096.865435
## MKTCAP     3674766  22.78165058   864.0291991 2.144930e+06     0.000000
## RF         3674766   0.67190117     0.6562667 3.697385e-03    -0.000600
## exc_return 3674766   6.75487102   355.9133710 8.238926e-03    -0.995200
##                    1%           5%          Q25           Q50          Q75
## PRC          -37.250000  -15.1875000     1.750000     10.880000 2.537500e+01
## RET           -0.382353   -0.2160000    -0.063636      0.000000 6.989200e-02
## SHROUT       220.000000  543.0000000  2158.000000   6722.000000 2.308800e+04
## VWRETD        -0.120224   -0.0702510    -0.017553      0.012963 3.877500e-02
## CPI           24.050000   29.7800000    72.200000    138.600000 1.815000e+02
## price_adj   -132.013742  -41.3044560     2.515256     18.752765 5.152655e+01
## MKTCAP      2882.600260 8298.7519876 44110.445011 169232.849184 7.749806e+05
## RF             0.000000    0.0000000     0.001500      0.003900 5.100000e-03
## exc_return    -0.386021   -0.2200667    -0.067402     -0.003500 6.652900e-02
##                     95%          99%          Max           SD
## PRC        5.900000e+01 1.120000e+02 3.395900e+05 1.370616e+03
## RET        2.583097e-01 5.652412e-01 2.400000e+01 1.764852e-01
## SHROUT     1.308076e+05 5.455820e+05 2.920640e+07 2.360843e+05
## VWRETD     7.431900e-02 1.186360e-01 3.941430e-01 4.664583e-02
## CPI        2.380170e+02 2.540950e+02 2.584440e+02 6.697004e+01
## price_adj  2.416898e+02 5.055671e+02 3.124130e+05 1.451614e+03
## MKTCAP     7.002273e+06 3.501269e+07 1.200346e+09 1.329309e+07
## RF         8.100000e-03 1.210000e-02 1.350000e-02 2.638589e-03
## exc_return 2.546768e-01 5.621018e-01 2.399660e+01 1.765959e-01
```

To describe variables for 1963-2019

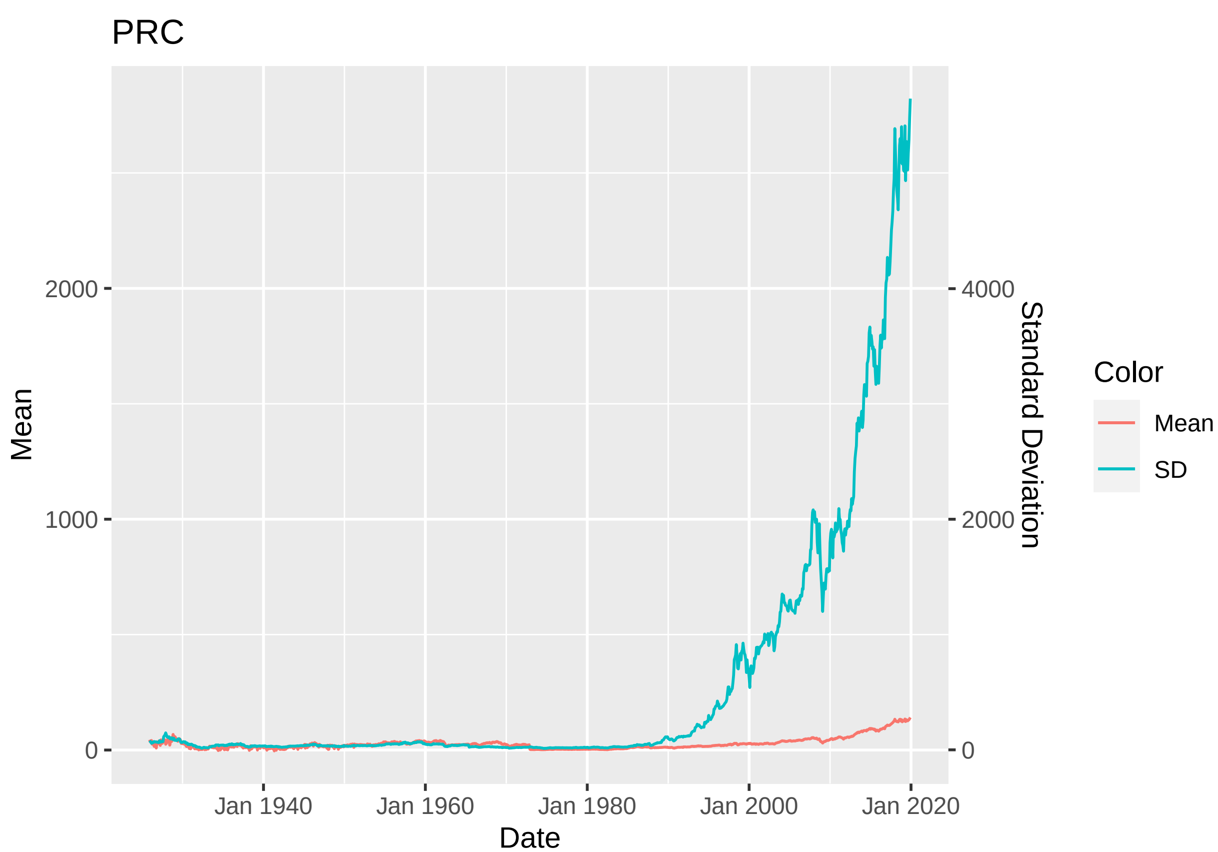
```r
msf %>% dplyr::filter(Date > "Dec 1962" & Date < "Jan 2020") %>% select(PRC, RET, SHROUT, VWRETD, CPI, p
```

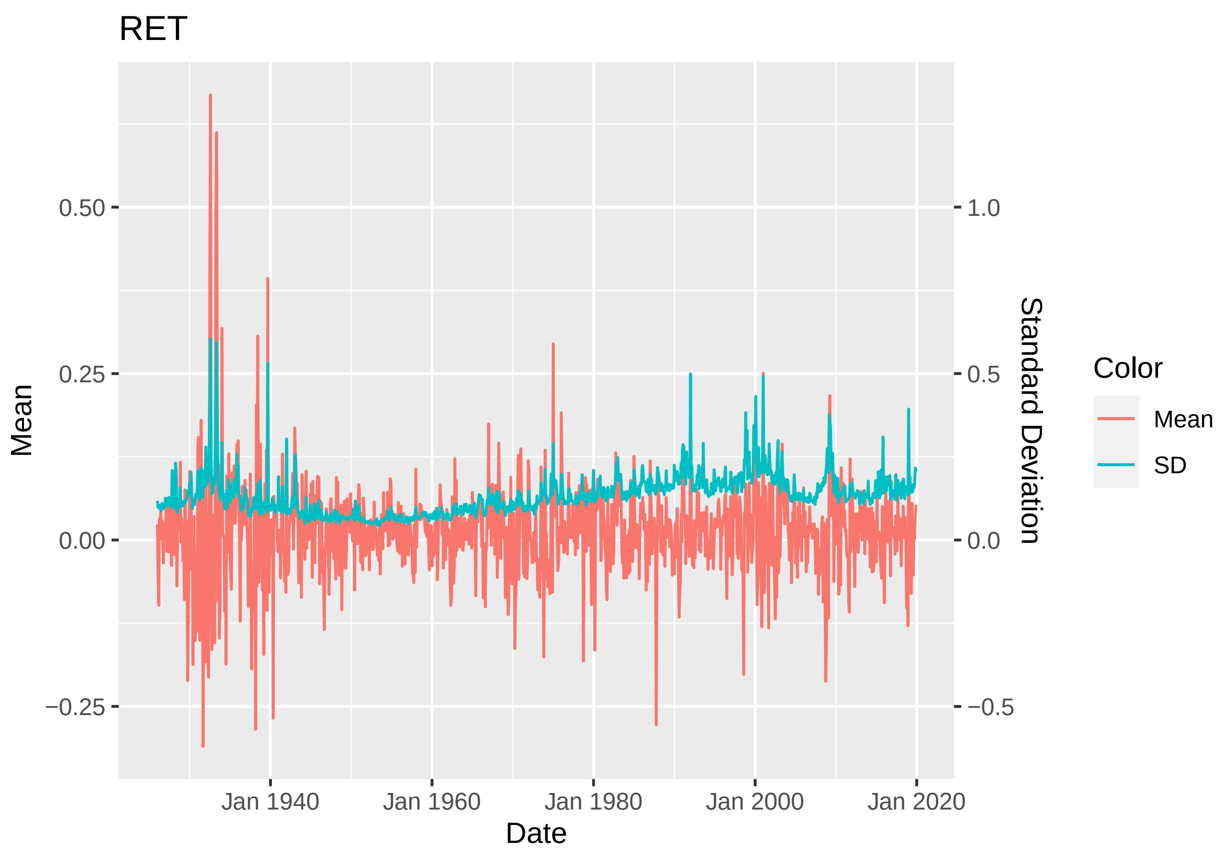
```
##                  N     Skewness      Kurtosis         Mean          Min
## PRC        3278171 153.39770621 26911.4200018 2.772175e+01 -1832.500000
## RET        3278171   6.84614795   362.9834237 1.177406e-02    -0.993600
## SHROUT     3278171  28.18830322  1548.2205084 4.520153e+04     0.000000
## VWRETD     3278171  -0.61013361     2.2860965 9.260334e-03    -0.225363
## CPI        3278171  -0.04424193    -0.9984246 1.396083e+02    30.440000
## price_adj  3278171 136.37947542 21220.8348679 4.662440e+01 -4079.560229
## MKTCAP     3278171  22.38598156   829.9108258 2.194787e+06     0.000000
## RF         3278171   0.58579334     0.7321230 4.001997e-03     0.000000
## exc_return 3278171   6.84065447   362.5916073 7.784715e-03    -0.995200
##                    1%          5%          Q25          Q50          Q75
## PRC          -31.250000  -13.750000     1.520000     10.00000 2.387500e+01
## RET           -0.389313   -0.222222    -0.066667      0.00000 7.142900e-02
## SHROUT       461.000000  855.000000  2887.000000   8362.00000 2.683000e+04
## VWRETD        -0.110995   -0.070115    -0.017001      0.01295 3.851200e-02
## CPI           30.980000   36.400000    94.700000    144.50000 1.849000e+02
## price_adj   -102.315184  -38.847772     2.111210     16.76638 4.402201e+01
## MKTCAP      2777.905514 7925.366667 41350.408119 160425.65834 7.627880e+05
## RF             0.000000    0.000000     0.002300      0.00410 5.300000e-03
## exc_return    -0.393489   -0.225922    -0.070456     -0.00400 6.782900e-02
##                     95%          99%          Max           SD
```

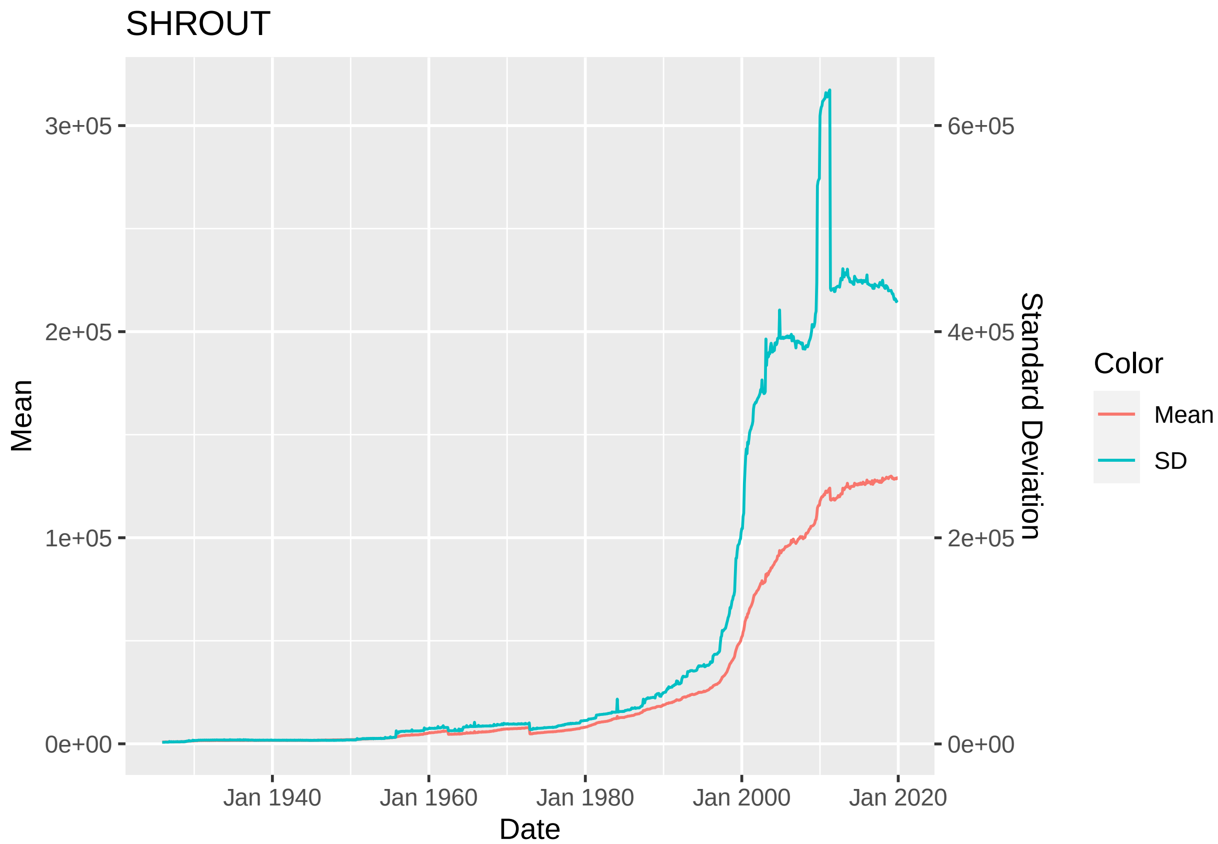
```
## PRC        5.575000e+01 1.041786e+02 3.395900e+05 1.452747e+03
## RET        2.656130e-01 5.750000e-01 2.400000e+01 1.804238e-01
## SHROUT     1.456900e+05 5.980720e+05 2.920640e+07 2.495624e+05
## VWRETD     7.418800e-02 1.140300e-01 1.655840e-01 4.457873e-02
## CPI        2.388350e+02 2.549430e+02 2.584440e+02 6.334321e+01
## price_adj  1.539903e+02 3.594497e+02 3.124130e+05 1.493912e+03
## MKTCAP     7.228154e+06 3.620663e+07 1.200346e+09 1.363592e+07
## RF         8.200000e-03 1.210000e-02 1.350000e-02 2.608493e-03
## exc_return 2.614000e-01 5.709600e-01 2.399660e+01 1.804879e-01
```

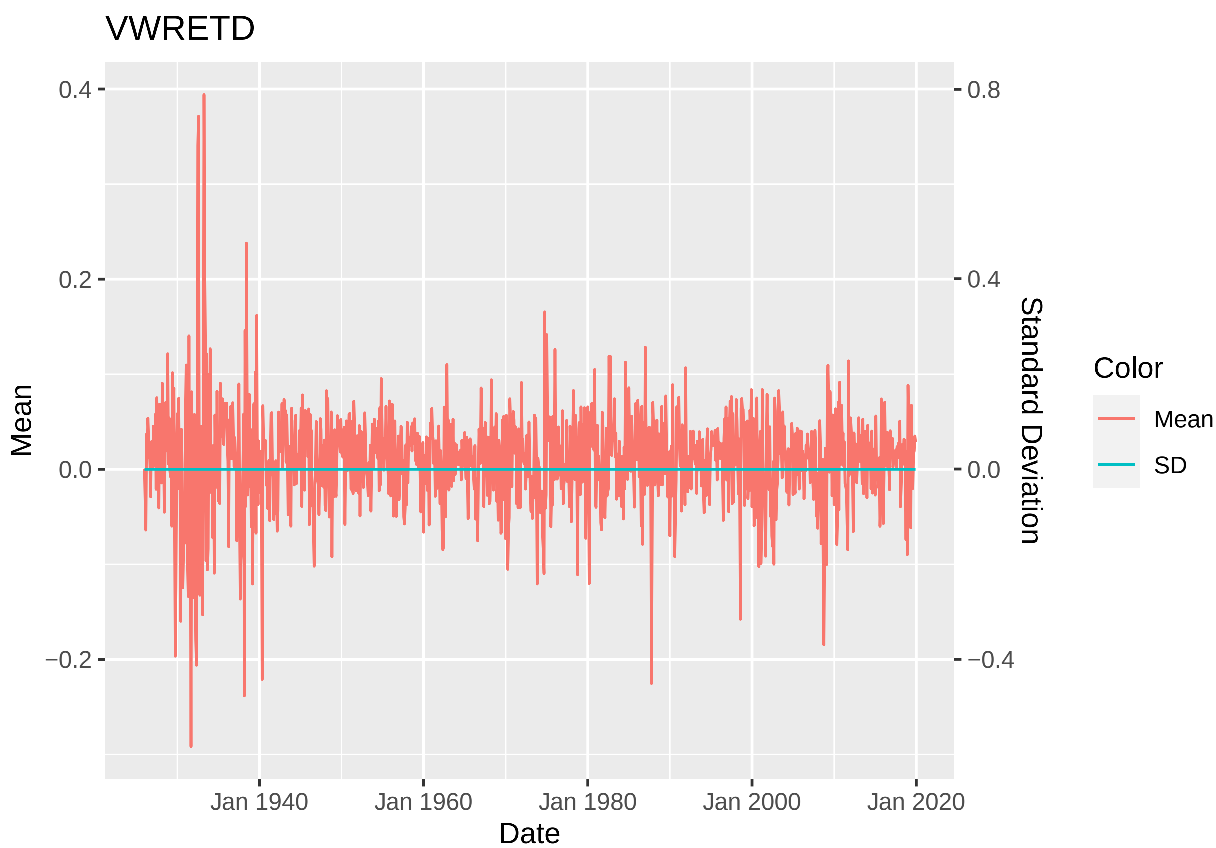
To plot the mean and standard deviation

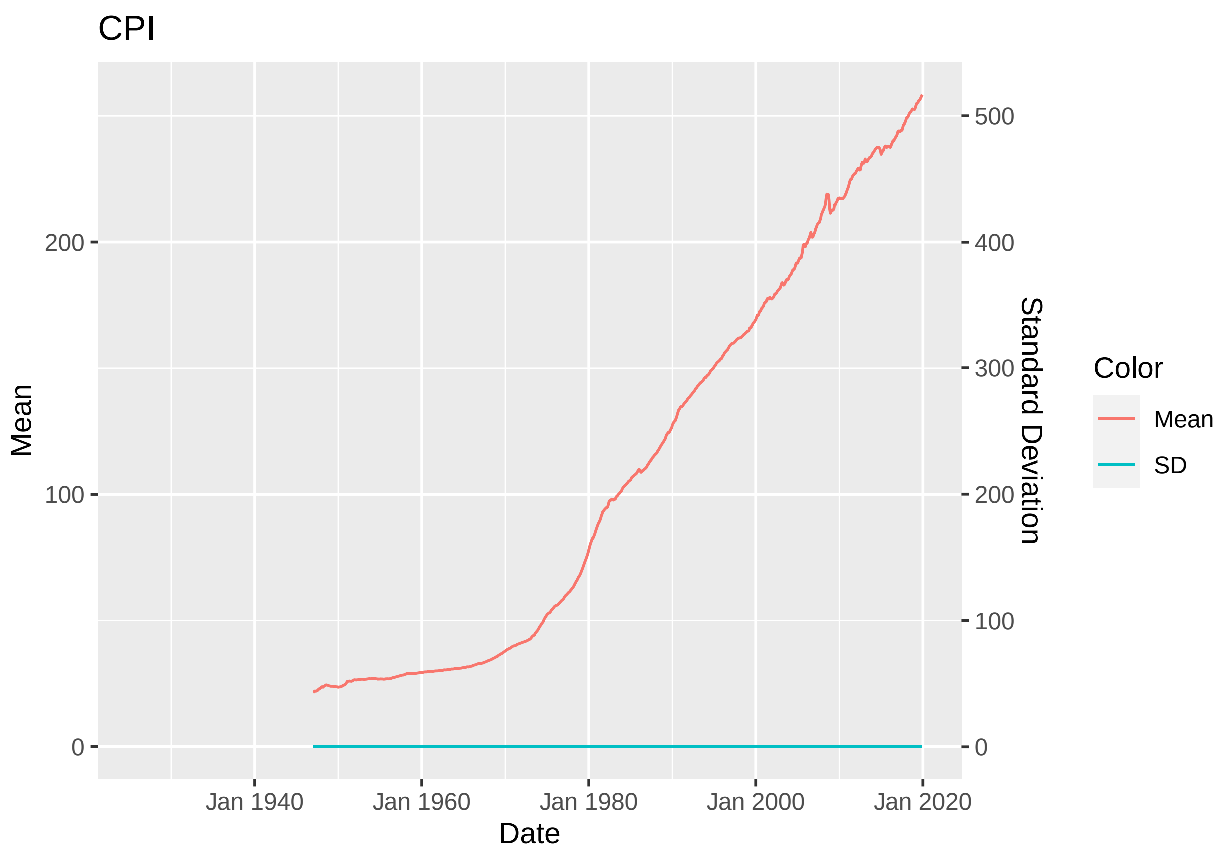
```
for (cln in c('PRC', 'RET', 'SHROUT', 'VWRETD', 'CPI',
              'price_adj', 'MKTCAP',
             'RF', 'exc_return')) {

plt <- msf %>% select(Date, eval(cln)) %>% group_by(Date) %>%
    summarize(Mean = mean(eval(as.name(cln)), na.rm = T),
              SD = sd(eval(as.name(cln)), na.rm = T)) %>%
  ggplot(aes(x = Date))+ geom_line(aes(y=Mean,color="Mean"))+
  geom_line(aes(y=SD/2, color = "SD"))+
  scale_y_continuous(sec.axis = sec_axis(~.*2, name="Standard Deviation"))+
  labs(y = "Mean", color = "Color", title = as.character(cln))
print(plt)
}
```

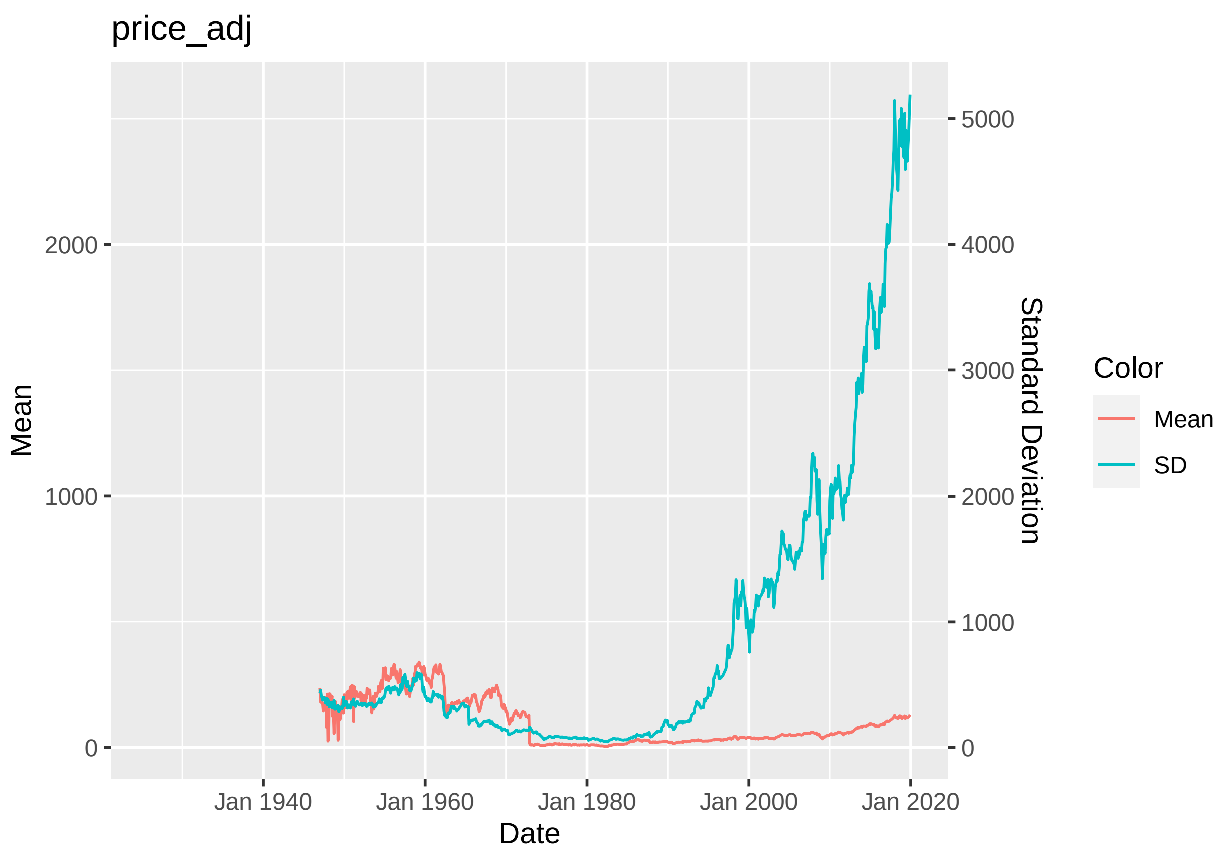


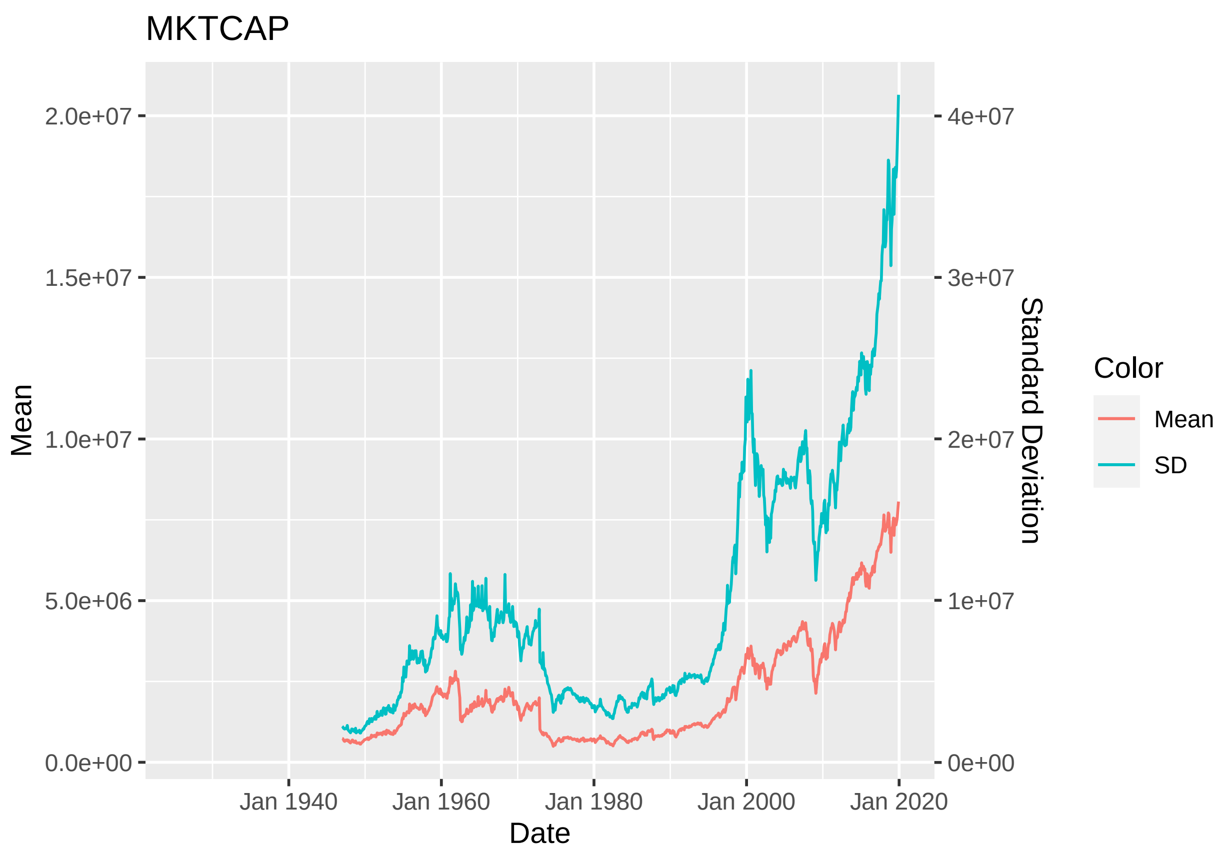

RET
Mean
0.50
0.25
0.00
−0.25
Standard Deviation
1.0
0.5
0.0
−0.5
Jan 1940
Jan 1960
Jan 1980
Jan 2000
Jan 2020
Date
Color
Mean
SD


SHROUT
Mean
Standard Deviation
3e+05
2e+05
1e+05
0e+00
6e+05
4e+05
2e+05
0e+00
Jan 1940
Jan 1960
Jan 1980
Jan 2000
Jan 2020
Date
Color
Mean
SD

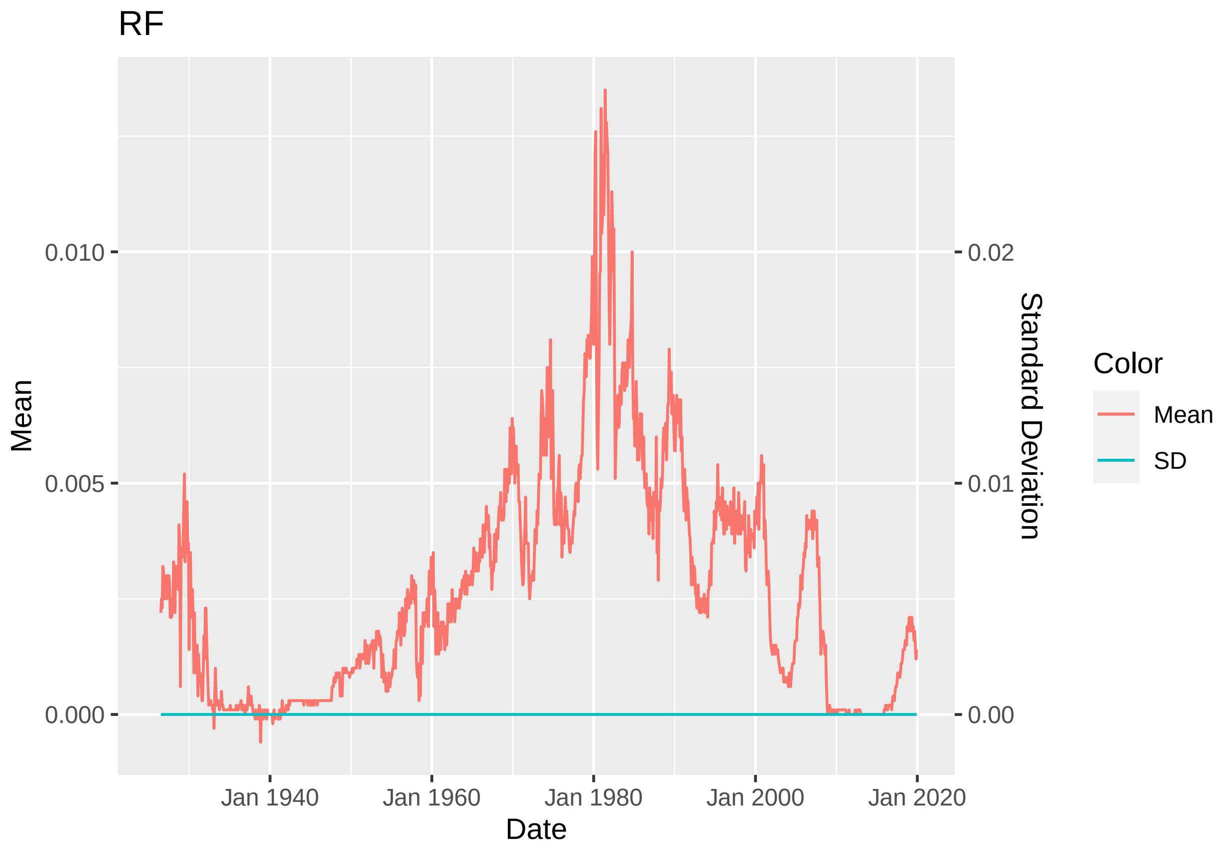

VWRETD
Mean
Standard Deviation
Date
0.4
0.2
0.0
−0.2
0.8
0.4
0.0
−0.4
Jan 1940
Jan 1960
Jan 1980
Jan 2000
Jan 2020
Color
Mean
SD


CPI
Mean
200
100
0
Standard Deviation
500
400
300
200
100
0
Jan 1940
Jan 1960
Jan 1980
Jan 2000
Jan 2020
Date
Color
Mean
SD

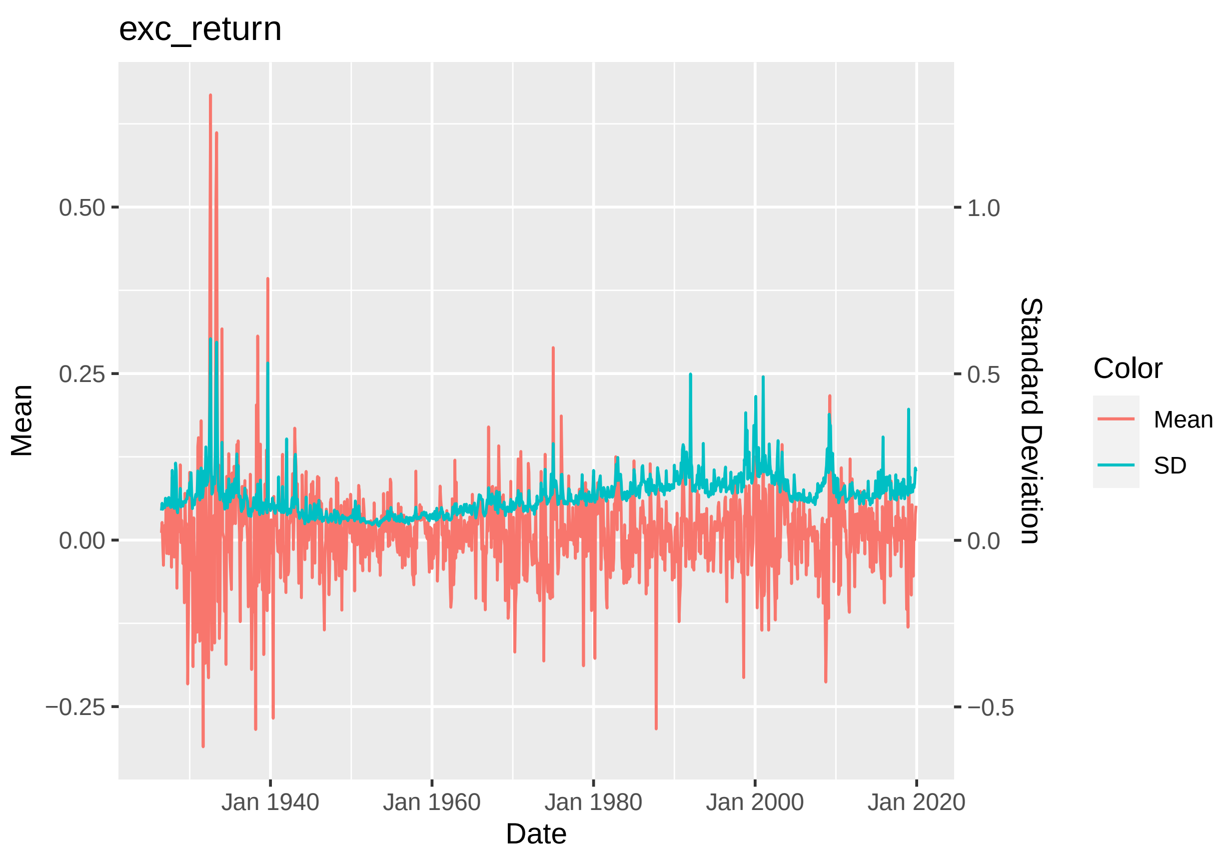

price_adj
Mean
2000
1000
0
Standard Deviation
5000
4000
3000
2000
1000
0
Jan 1940
Jan 1960
Jan 1980
Jan 2000
Jan 2020
Date
Color
Mean
SD



MKTCAP
Mean
2.0e+07
1.5e+07
1.0e+07
5.0e+06
0.0e+00
Standard Deviation
4e+07
3e+07
2e+07
1e+07
0e+00
Jan 1940
Jan 1960
Jan 1980
Jan 2000
Jan 2020
Date
Color
Mean
SD

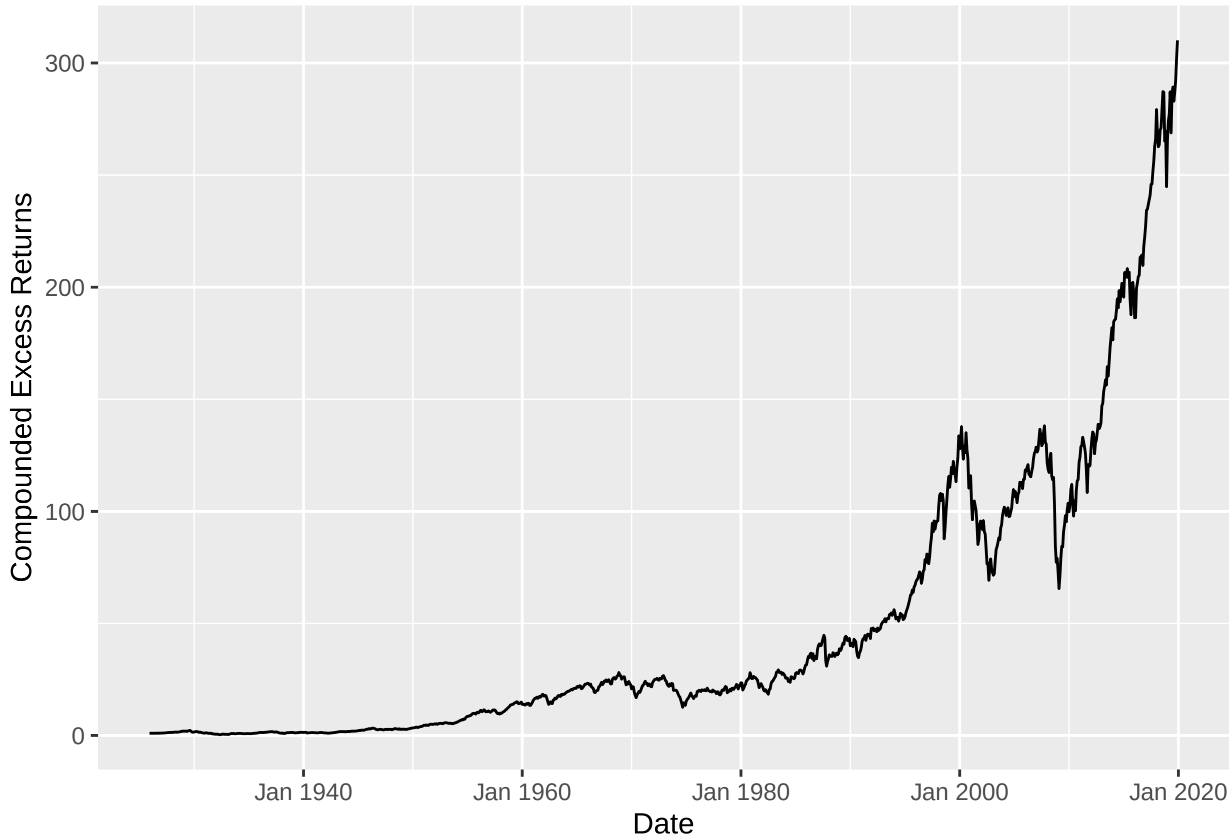


RF
Mean
0.010
0.005
0.000
Standard Deviation
0.02
0.01
0.00
Jan 1940
Jan 1960
Jan 1980
Jan 2000
Jan 2020
Date
Color
Mean
SD



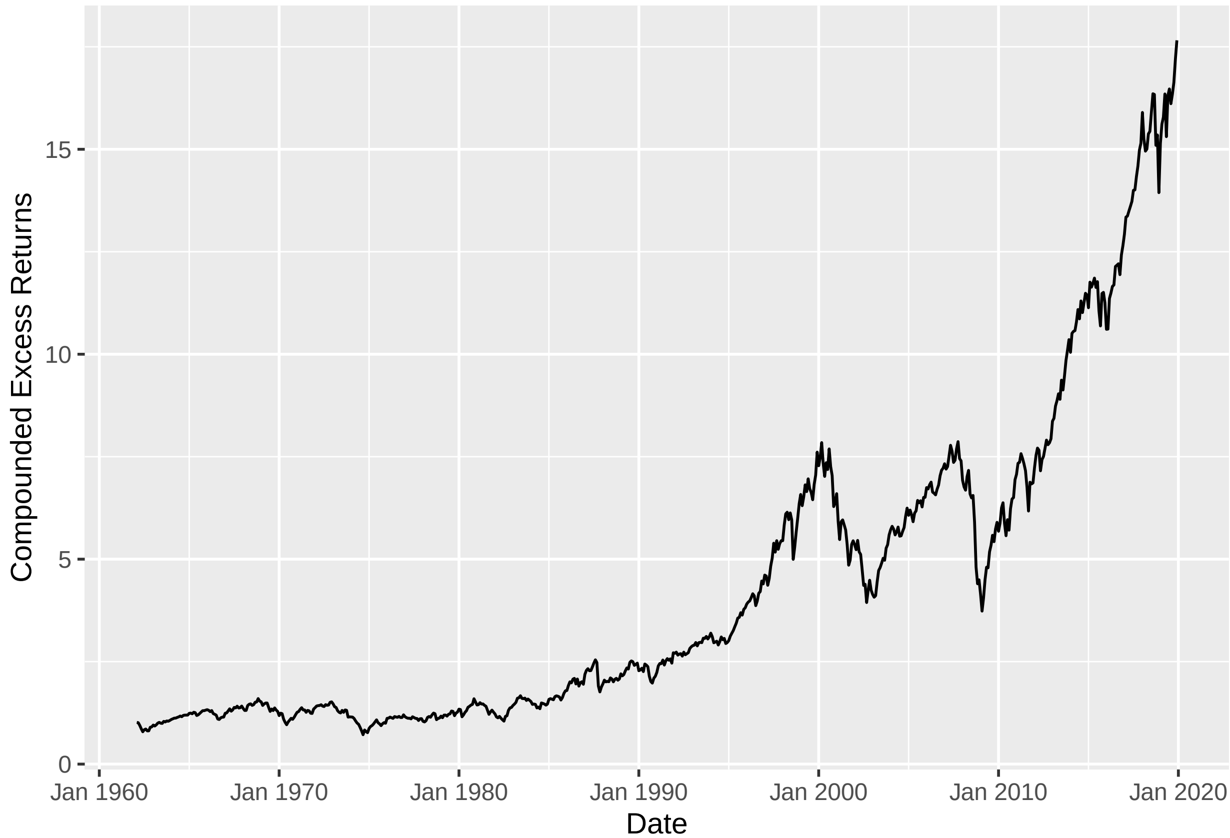
The compounded excess returns for 1925-2019

```
# The VWRETDs are the same for each month, so use mean to get the exact rate
msf %>% group_by(Date) %>% summarize (Erm = mean(VWRETD, na.rm = T),
                                      RF = mean(RF, na.rm =T)) %>%
  mutate(exc_r = Erm - RF) %>% mutate(c = cumprod(ifelse(is.na(exc_r),1,
                                                         (exc_r+1))))  %>%
  ggplot(aes(x = Date, y = c)) +
  geom_line() + ylab("Compounded Excess Returns")
```



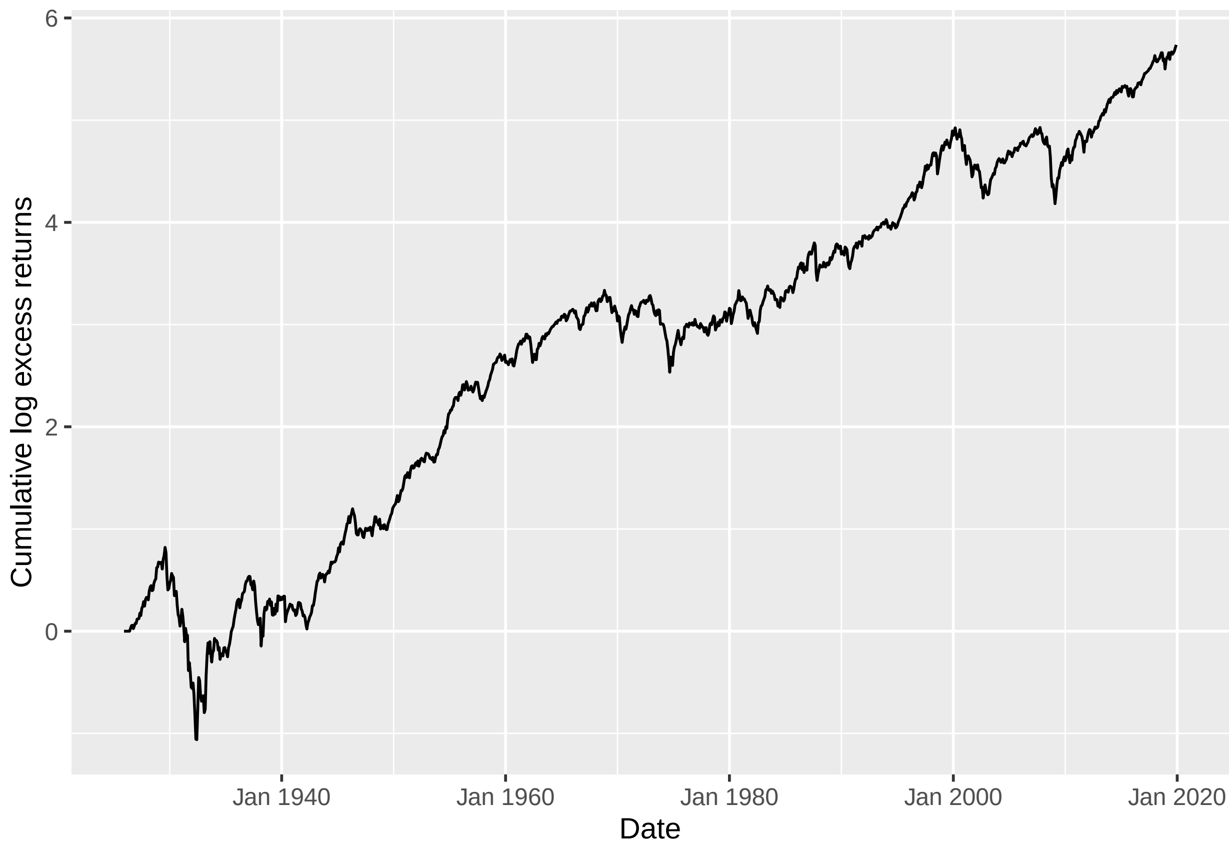
The compounded excess returns for 1963-2019

```
msf %>% dplyr::filter(Date>1962) %>% group_by(Date) %>%
  summarize (Erm = mean(VWRETD, na.rm = T), RF = mean(RF, na.rm =T)) %>%
  mutate(exc_r = Erm - RF) %>% mutate(c = cumprod(ifelse(is.na(exc_r),1,
                                                    (exc_r+1))))  %>%
  ggplot(aes(x = Date, y = c)) +
  geom_line() + ylab("Compounded Excess Returns")
```



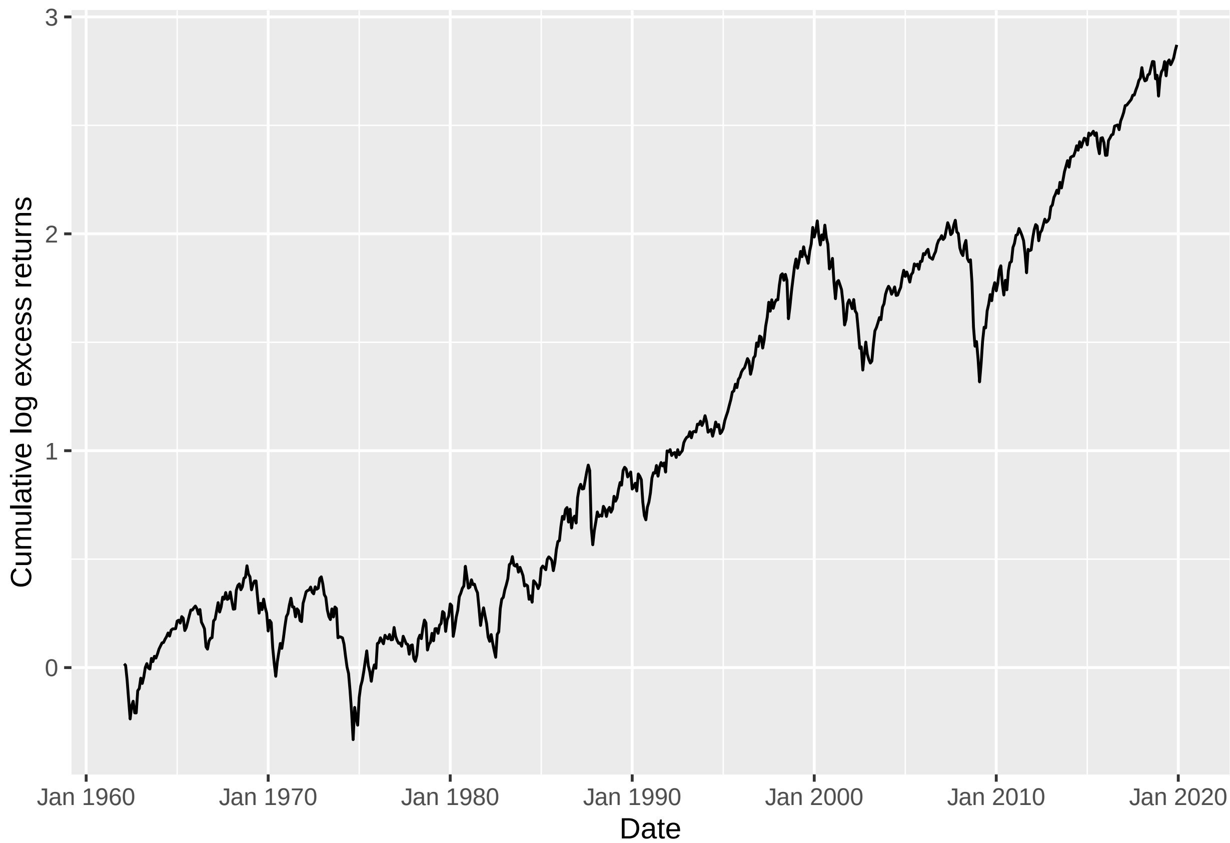
To plot the cumulative log excess returns for 1925-2019

```
msf %>% group_by(Date) %>% summarize (Erm = mean(VWRETD, na.rm = T),
                                      RF = mean(RF, na.rm =T)) %>%
  mutate(log_exc_r = log(1+ Erm - RF)) %>%
  mutate(c = cumsum(ifelse(is.na(log_exc_r),0,(log_exc_r))))  %>%
  ggplot(aes(x = Date, y = c)) +
  geom_line() + ylab("Cumulative log excess returns")
```



To plot the cumulative log excess returns for 1963-2019

```
msf %>% dplyr::filter(Date>1962) %>% group_by(Date) %>%
  summarize (Erm = mean(VWRETD, na.rm = T), RF = mean(RF, na.rm =T)) %>%
  mutate(log_exc_r = log(1+ Erm - RF)) %>%
  mutate(c = cumsum(ifelse(is.na(log_exc_r),0,(log_exc_r))))  %>%
  ggplot(aes(x = Date, y = c)) +
  geom_line() + ylab("Cumulative log excess returns")
```


Cumulative log excess returns
3
2
1
0
Jan 1960
Jan 1970
Jan 1980
Jan 1990
Jan 2000
Jan 2010
Jan 2020
Date