5200HW1

Jesse Smith, Lianne Chin, Qianyun Liu, Tianfeng Wang

2024-10-02

a. Extracts data for SP500 return index from YFinance up to 12/31/2023, only selecting the closing prices of last day of each month

```r
rm(list=ls())
library(tidyverse)
library(tidyquant)
library(dplyr)
library(lubridate)

sp500_tr_index <- tq_get("^SP500TR", get = "stock.prices",
  from = "1988-01-04", to = "2023-12-31") |>
    select(date, tr_index = close)

monthly_sp500_tr_index <- sp500_tr_index |>
    mutate(year = year(date), month = month(date)) |>
    group_by(year, month) |>
    filter(date == max(date)) |>
    ungroup() |>
    select(-year, -month) |>
    drop_na()

head(monthly_sp500_tr_index)
```

```
## # A tibble: 6 x 2
##   date       tr_index
##   <date>        <dbl>
## 1 1988-01-29     257.
## 2 1988-02-29     269.
## 3 1988-03-31     261.
## 4 1988-04-29     264.
## 5 1988-05-31     266.
## 6 1988-06-30     279.
```

b. Downloads Shiller data, converts P and D to numeric values, formats dates

```r
library(readxl)
library(stringr)
aux <- tempfile(fileext = ".xls")
download.file(url = "http://www.econ.yale.edu/~shiller/data/ie_data.xls",
                 destfile = aux, mode='wb')
rshiller_raw <- read_excel(aux, sheet = "Data", skip = 7)

rshiller <- rshiller_raw |>
  mutate(P = as.numeric(P),
         D = as.numeric(D),
         Date = format(Date, nsmall = 2),
         Date = str_replace(Date, "\\.", "-"),
         Date = ceiling_date(ym(Date), "month") - days(1)) |>
  rename(date = Date)

head(rshiller)
```

```
## # A tibble: 6 x 22
##   date             P     D     E   CPI Fraction `Rate GS10` Price...8 Dividend
##   <date>     <dbl> <dbl> <dbl> <dbl>    <dbl>       <dbl>     <dbl>    <dbl>
## 1 1871-01-31  4.44  0.26   0.4  12.5    1871.        5.32      109.     6.39
## 2 1871-02-28  4.5   0.26   0.4  12.8    1871.        5.32      107.     6.20
## 3 1871-03-31  4.61  0.26   0.4  13.0    1871.        5.33      108.     6.11
## 4 1871-04-30  4.74  0.26   0.4  12.6    1871.        5.33      116.     6.34
## 5 1871-05-31  4.86  0.26   0.4  12.3    1871.        5.33      121.     6.48
## 6 1871-06-30  4.82  0.26   0.4  12.1    1871.        5.34      122.     6.59
## # i 13 more variables: Price...10 <dbl>, Earnings...11 <dbl>,
## #   Earnings...12 <dbl>, CAPE <chr>, ...14 <lgl>, `TR CAPE` <chr>, ...16 <lgl>,
## #   Yield <dbl>, Returns...18 <dbl>, Returns...19 <dbl>,
## #   `Real Return...20` <dbl>, `Real Return...21` <dbl>, Returns...22 <dbl>
```

c. Calculates monthly total return using Shiller data

```
rshiller <- rshiller |>
  mutate(monthly_D = D/12,
         monthly_tr = (P + monthly_D) / lag(P) - 1)

head(rshiller)
```

```
## # A tibble: 6 x 24
##   date             P     D     E   CPI Fraction `Rate GS10` Price...8 Dividend
##   <date>     <dbl> <dbl> <dbl> <dbl>    <dbl>       <dbl>     <dbl>    <dbl>
## 1 1871-01-31  4.44  0.26   0.4  12.5    1871.        5.32      109.     6.39
## 2 1871-02-28  4.5   0.26   0.4  12.8    1871.        5.32      107.     6.20
## 3 1871-03-31  4.61  0.26   0.4  13.0    1871.        5.33      108.     6.11
## 4 1871-04-30  4.74  0.26   0.4  12.6    1871.        5.33      116.     6.34
## 5 1871-05-31  4.86  0.26   0.4  12.3    1871.        5.33      121.     6.48
## 6 1871-06-30  4.82  0.26   0.4  12.1    1871.        5.34      122.     6.59
## # i 15 more variables: Price...10 <dbl>, Earnings...11 <dbl>,
## #   Earnings...12 <dbl>, CAPE <chr>, ...14 <lgl>, `TR CAPE` <chr>, ...16 <lgl>,
## #   Yield <dbl>, Returns...18 <dbl>, Returns...19 <dbl>,
## #   `Real Return...20` <dbl>, `Real Return...21` <dbl>, Returns...22 <dbl>,
## #   monthly_D <dbl>, monthly_tr <dbl>
```

d-1. Calculates cumulative returns since 1988 using Shiller data

```
rshiller_1988 <- rshiller |>
  filter(date >= "1988-01-01") |>
  mutate(cum_tr = cumprod(1 + monthly_tr) - 1,
         year_month = floor_date(date, "month")) |>
  select(date, year_month, monthly_tr, cum_tr)

head(rshiller_1988)
```

```
## # A tibble: 6 x 4
##   date       year_month monthly_tr cum_tr
##   <date>     <date>          <dbl>  <dbl>
## 1 1988-01-31 1988-01-01    0.0425  0.0425
## 2 1988-02-29 1988-02-01    0.0333  0.0772
## 3 1988-03-31 1988-03-01    0.0323  0.112
## 4 1988-04-30 1988-04-01   -0.00883 0.102
## 5 1988-05-31 1988-05-01   -0.0219  0.0781
## 6 1988-06-30 1988-06-01    0.0600  0.143
```

d-2. Calculates cumulative returns using Yahoo Finance tr index data

```
monthly_sp500_tr_index <- monthly_sp500_tr_index |>
  mutate(year_month = floor_date(date, "month"),
         monthly_r = tr_index / lag(tr_index) - 1,
         cum_r = tr_index / first(tr_index) - 1)

head(monthly_sp500_tr_index)
```

```
## # A tibble: 6 x 5
##   date       tr_index year_month monthly_r  cum_r
##   <date>        <dbl> <date>         <dbl>  <dbl>
## 1 1988-01-29     257. 1988-01-01  NA       0
## 2 1988-02-29     269. 1988-02-01   0.0466  0.0466
## 3 1988-03-31     261. 1988-03-01  -0.0309  0.0143
## 4 1988-04-29     264. 1988-04-01   0.0111  0.0255
## 5 1988-05-31     266. 1988-05-01   0.00864 0.0344
## 6 1988-06-30     279. 1988-06-01   0.0459  0.0818
```

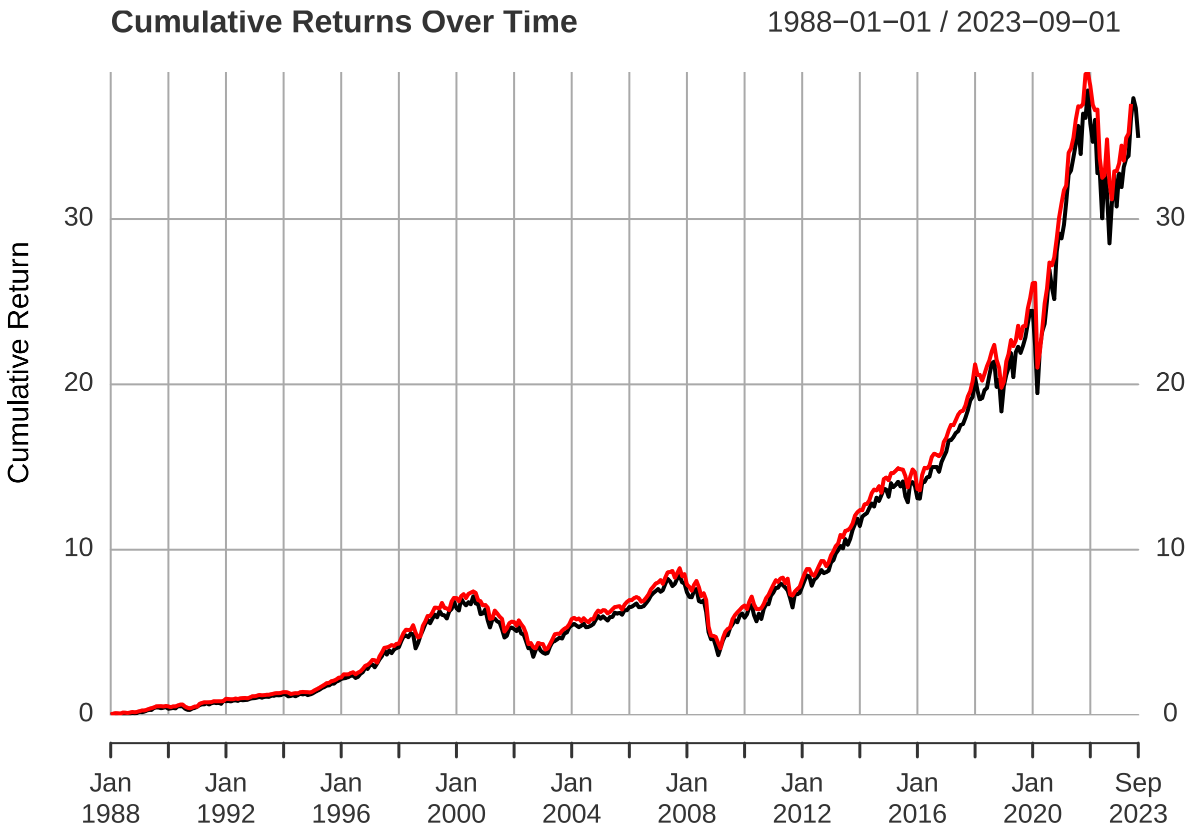
d-3. Merges two cumulative return indices, by month

```
rshiller_tr_index_merged <- rshiller_1988 |>
  left_join(monthly_sp500_tr_index, by = "year_month") |>
  rename(
    cum_r_rshiller = cum_tr,
    cum_r_tr_index = cum_r,
    monthly_r_rshiller = monthly_tr,
    monthly_r_tr_index = monthly_r
  )

head(rshiller_tr_index_merged)
```

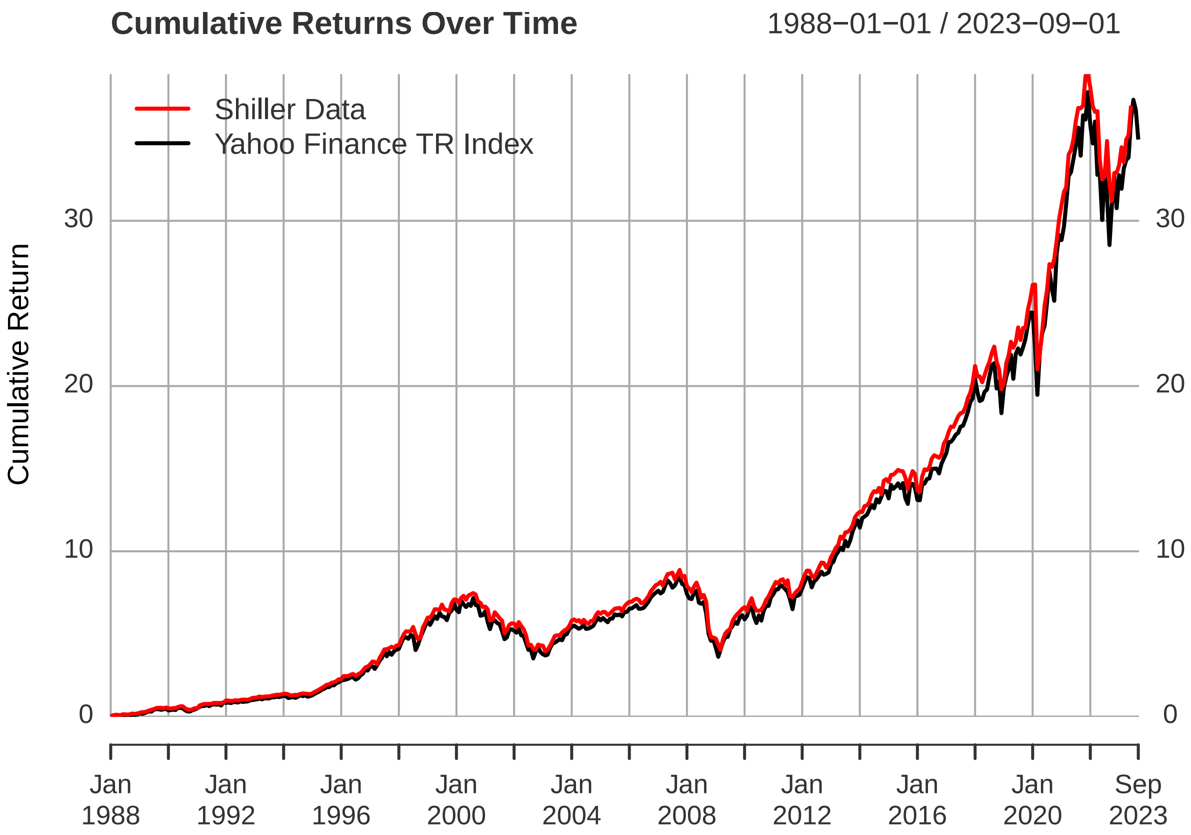
```
## # A tibble: 6 x 8
##   date.x     year_month monthly_r_rshiller cum_r_rshiller date.y     tr_index
##   <date>     <date>                  <dbl>          <dbl> <date>        <dbl>
## 1 1988-01-31 1988-01-01            0.0425          0.0425 1988-01-29     257.
## 2 1988-02-29 1988-02-01            0.0333          0.0772 1988-02-29     269.
## 3 1988-03-31 1988-03-01            0.0323          0.112  1988-03-31     261.
## 4 1988-04-30 1988-04-01           -0.00883         0.102  1988-04-29     264.
## 5 1988-05-31 1988-05-01           -0.0219          0.0781 1988-05-31     266.
## 6 1988-06-30 1988-06-01            0.0600          0.143  1988-06-30     279.
## # i 2 more variables: monthly_r_tr_index <dbl>, cum_r_tr_index <dbl>
```

d-4. Constructs time series of compounded returns, plots comparison graph

```
library(xts)
library(zoo)

cum_r_rshiller_1988_xts <- xts(rshiller_tr_index_merged$cum_r_rshiller, order.by =
                            rshiller_tr_index_merged$year_month)
cum_r_tr_index_xts <- xts(rshiller_tr_index_merged$cum_r_tr_index, order.by =
                            rshiller_tr_index_merged$year_month)

cum_r_1988_xts_merged <- merge(cum_r_rshiller_1988_xts, cum_r_tr_index_xts, all = TRUE)

# Plots time series in comparison
plot.xts(cum_r_1988_xts_merged, main = "Cumulative Returns Over Time", ylab = "Cumulative Return",
         xlab = "Date", col = c("red", "black"))
```



```
addLegend("topleft", legend.names = c("Shiller Data", "Yahoo Finance TR Index"),
          col = c("red", "black"), lty = 1, lwd = 2)
```



Shiller data and Yahoo Finance data seem to closely track each others movement. Following possibilities that explain this difference: 1. Selecting only the closing price of last day of each month for Yahoo Finance SP500 data 2. Approximating monthly dividends using Shiller's data

d-5. Reports Pearson and Spearman correlations

```
pearson_corr <- cor(rshiller_tr_index_merged$cum_r_rshiller,
                    rshiller_tr_index_merged$cum_r_tr_index,
                    method = "pearson", use = "complete.obs")

spearman_corr <- cor(rshiller_tr_index_merged$cum_r_rshiller,
                     rshiller_tr_index_merged$cum_r_tr_index,
                     method = "spearman", use = "complete.obs")

cat("Pearson Correlation:", pearson_corr, "\n")
```

```
## Pearson Correlation: 0.9992168
```

```
cat("Spearman Correlation:", spearman_corr, "\n")
```

```
## Spearman Correlation: 0.9992102
```

e-1. Calculates cumulative returns since 1928 using Shiller data

```
rshiller_1928 <- rshiller |>
  filter(date >= "1928-01-01") |>
  mutate(cum_tr = cumprod(1 + monthly_tr) - 1,
         year_month = floor_date(date, "month")) |>
  select(date, year_month, monthly_tr, cum_tr)
```

```
head(rshiller_1928)
```

```
## # A tibble: 6 x 4
##   date       year_month monthly_tr    cum_tr
##   <date>     <date>          <dbl>     <dbl>
## 1 1928-01-31 1928-01-01    0.00772  0.00772
## 2 1928-02-29 1928-02-01   -0.00826 -0.000603
## 3 1928-03-31 1928-03-01    0.0575   0.0569
## 4 1928-04-30 1928-04-01    0.0667   0.127
## 5 1928-05-31 1928-05-01    0.0344   0.166
## 6 1928-06-30 1928-06-01   -0.0456   0.113
```

e-2. Calculates cumulative unadjusted returns from 1928 to 2023 using Yahoo Finance SP500 price index data

```
sp500_price_index <- tq_get("^GSPC", get = "stock.prices",
                            from = "1928-01-01", to = "2023-12-31") |>
  select(date, pr_index = close) |>
  drop_na()

monthly_sp500_price_index <- sp500_price_index |>
  mutate(year = year(date), month = month(date)) |>
  group_by(year, month) |>
  filter(date == max(date)) |>
  ungroup() |>
  select(-year, -month) |>
  mutate(year_month = floor_date(date, "month"),
         monthly_r_unadjusted = pr_index / lag(pr_index) - 1,
         cum_r_unadjusted = pr_index / first(pr_index) - 1)

head(monthly_sp500_price_index)
```

```
## # A tibble: 6 x 5
##   date       pr_index year_month monthly_r_unadjusted cum_r_unadjusted
##   <date>        <dbl> <date>                    <dbl>            <dbl>
## 1 1928-01-31     17.6 1928-01-01              NA              0
## 2 1928-02-29     17.3 1928-02-01              -0.0176          -0.0176
## 3 1928-03-30     19.3 1928-03-01               0.117            0.0973
## 4 1928-04-30     19.8 1928-04-01               0.0244           0.124
## 5 1928-05-31     20   1928-05-01               0.0127           0.138
## 6 1928-06-29     19.1 1928-06-01              -0.0430           0.0894
```

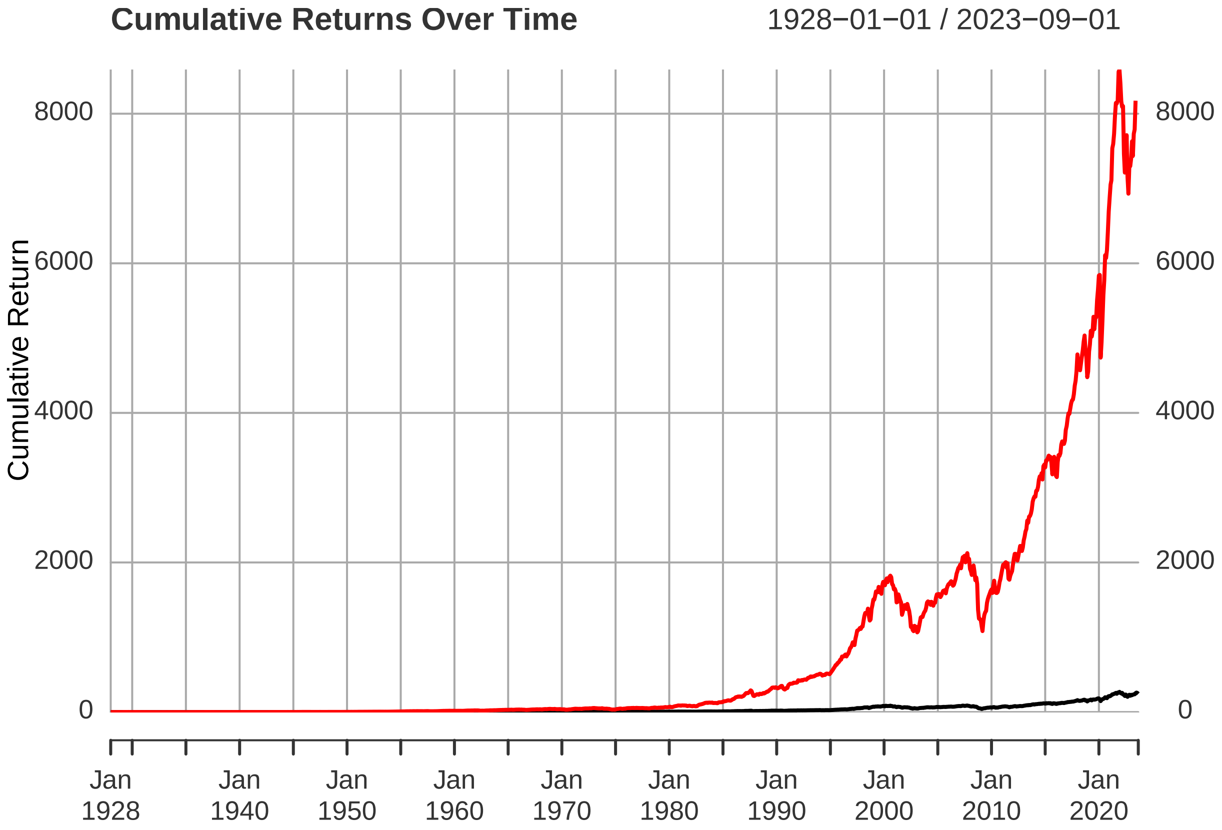
e-3. Merges two cumulative return indices, by month

```
rshiller_price_index_merged <- rshiller_1928 |>
  left_join(monthly_sp500_price_index, by = "year_month") |>
   rename(
     cum_r_rshiller = cum_tr,
     monthly_r_rshiller = monthly_tr,
   )

head(rshiller_price_index_merged)
```

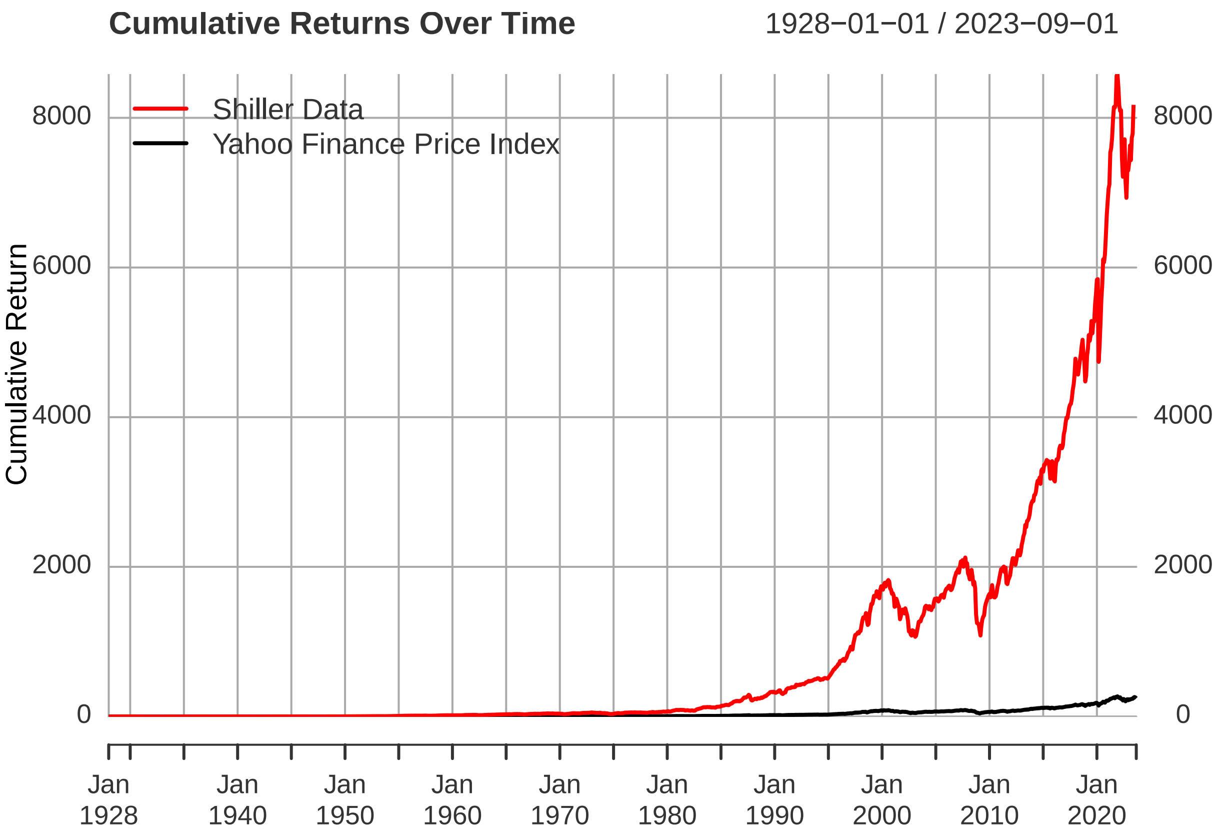
```
## # A tibble: 6 x 8
##   date.x     year_month monthly_r_rshiller cum_r_rshiller date.y     pr_index
##   <date>     <date>                  <dbl>          <dbl> <date>        <dbl>
## 1 1928-01-31 1928-01-01            0.00772        0.00772  1928-01-31     17.6
## 2 1928-02-29 1928-02-01           -0.00826      -0.000603 1928-02-29     17.3
## 3 1928-03-31 1928-03-01            0.0575        0.0569   1928-03-30     19.3
## 4 1928-04-30 1928-04-01            0.0667        0.127    1928-04-30     19.8
## 5 1928-05-31 1928-05-01            0.0344        0.166    1928-05-31     20
## 6 1928-06-30 1928-06-01           -0.0456        0.113    1928-06-29     19.1
## # i 2 more variables: monthly_r_unadjusted <dbl>, cum_r_unadjusted <dbl>
```

e-4. Creates xts objects and comparison graphics

```
cum_r_rshiller_1928_xts <- xts(rshiller_price_index_merged$cum_r_rshiller, order.by =
                            rshiller_price_index_merged$year_month)
cum_r_unadjusted_xts <- xts(rshiller_price_index_merged$cum_r_unadjusted, order.by =
                            rshiller_price_index_merged$year_month)

cum_r_1928_xts_merged <- merge(cum_r_rshiller_1928_xts, cum_r_unadjusted_xts, all = TRUE)

plot.xts(cum_r_1928_xts_merged, main = "Cumulative Returns Over Time", ylab = "Cumulative Return",
         xlab = "Date", col = c("red", "black"))
```



```
addLegend("topleft", legend.names = c("Shiller Data", "Yahoo Finance Price Index"),
          col = c("red", "black"), lty = 1, lwd = 2)
```



e-5. Reports metrics - average, standard deviation of the monthly adjusted and unadjusted returns

```
avg_sd_report <- rshiller_price_index_merged |>
  summarize(
    avg_r_rshiller_adjusted = mean(monthly_r_rshiller, na.rm= TRUE),
    sd_r_rshiller_adjusted = sd(monthly_r_rshiller, na.rm= TRUE),
    avg_r_unadjusted = mean(monthly_r_unadjusted, na.rm= TRUE),
    sd_r_unadjusted = sd(monthly_r_unadjusted, na.rm= TRUE)
  )

summary_table <- matrix(c(avg_sd_report$avg_r_rshiller_adjusted,
                          avg_sd_report$sd_r_rshiller_adjusted,
                          avg_sd_report$avg_r_unadjusted,
                          avg_sd_report$sd_r_unadjusted),
                        nrow = 2)

rownames(summary_table) <- c("Average", "Standard Deviation")
colnames(summary_table) <- c("R.Shiller Adjusted", "Unadjusted")

print(summary_table)
```

```
##                    R.Shiller Adjusted Unadjusted
## Average                   0.008886887  0.0062476
## Standard Deviation        0.044801007  0.0537306
```

It can be observed that Shiller's data presents a lower standard deviation for the monthly return, suggesting that the unadjusted data has a higher volatility. This could possibly be explained by the way that dividends are included in this adjusted data, which in comparison to the unadjusted monthly return, mitigating some level of volatility. The higher average monthly adjusted return can be explained by the inclusion of dividends.

f-1. Extracts rf and rme data from French's data library

```
library(frenchdata)
ff3 <- download_french_data("Fama/French 3 Factors")
ff3 <- ff3$subsets$data[[1]] |>
  mutate(
    date = floor_date(ymd(str_c(date, "01")), "month"),
    across(c(RF, `Mkt-RF`, SMB, HML), ~as.numeric(.) / 100),
    .keep = "none") |>
  rename_with(str_to_lower) |>
  rename(rme = `mkt-rf`) |>
  select(date,rf,rme) |>
  filter(date <= ymd("2023-12-31"))

head(ff3)
```

```
## # A tibble: 6 x 3
##   date           rf     rme
##   <date>      <dbl>   <dbl>
## 1 1926-07-01 0.0022  0.0296
## 2 1926-08-01 0.0025  0.0264
## 3 1926-09-01 0.0023  0.0036
## 4 1926-10-01 0.0032 -0.0324
## 5 1926-11-01 0.0031  0.0253
## 6 1926-12-01 0.0028  0.0262
```

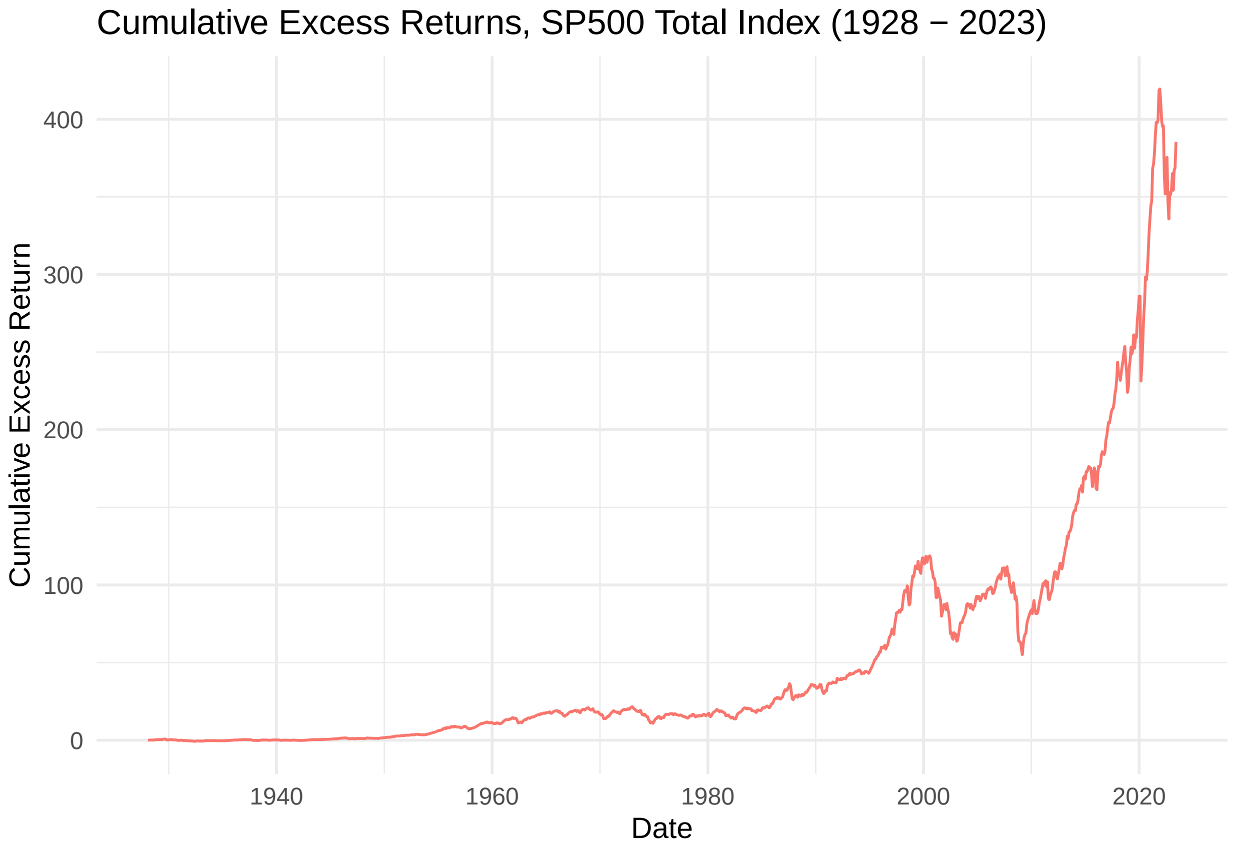
f-2. Calculates SP500 monthly total excess returns, merges with French's data. Calculates cumulative excess returns

```
monthly_sp500_excess_returns <- ff3 |>
  mutate(year_month = floor_date(date, "month"),
         year = year(date), month = month(date)) |>
  left_join(rshiller_price_index_merged, by = "year_month") |>
  drop_na() |>
  mutate(monthly_xr = monthly_r_rshiller - rf,
        cum_xr = cumprod(1 + monthly_xr) - 1,
        cum_rme = cumprod(1 + rme) - 1)

head(monthly_sp500_excess_returns)
```

```
## # A tibble: 6 x 16
##   date           rf     rme year_month  year month date.x     monthly_r_rshiller
##   <date>      <dbl>   <dbl> <date>     <dbl> <dbl> <date>                  <dbl>
## 1 1928-02-01 0.0033 -0.017  1928-02-01  1928     2 1928-02-29           -0.00826
## 2 1928-03-01 0.0029  0.0881 1928-03-01  1928     3 1928-03-31            0.0575
## 3 1928-04-01 0.0022  0.0423 1928-04-01  1928     4 1928-04-30            0.0667
## 4 1928-05-01 0.0032  0.0152 1928-05-01  1928     5 1928-05-31            0.0344
## 5 1928-06-01 0.0031 -0.0485 1928-06-01  1928     6 1928-06-30           -0.0456
## 6 1928-07-01 0.0032  0.0062 1928-07-01  1928     7 1928-07-31            0.0109
## # i 8 more variables: cum_r_rshiller <dbl>, date.y <date>, pr_index <dbl>,
## #   monthly_r_unadjusted <dbl>, cum_r_unadjusted <dbl>, monthly_xr <dbl>,
## #   cum_xr <dbl>, cum_rme <dbl>
```

f-3. Plots the cumulative excess returns of the SP500 total index

```r
library(ggplot2)

ggplot(monthly_sp500_excess_returns, aes(x = as.Date(paste(year, month, "01", sep = "-")), group = 1))
  geom_line(aes(y = cum_xr, color = "Cumulative Excess Return")) +
  labs(title = "Cumulative Excess Returns, SP500 Total Index (1928 - 2023)",
       x = "Date",
       y = "Cumulative Excess Return") +
  theme_minimal() +
  theme(legend.position = "none")
```



f-4. Identifies 5 longest downturns in duration, along with time of recovery

```r
library(PerformanceAnalytics)
xr_xts <- xts(monthly_sp500_excess_returns$monthly_xr,
                   order.by = monthly_sp500_excess_returns$year_month)

drawdowns <- Drawdowns(xr_xts)

top_drawdowns <- table.Drawdowns(xr_xts)
top_drawdowns_by_length <- top_drawdowns[order(top_drawdowns$`To Trough`, decreasing = TRUE), ]

top_5_drawdowns_by_length <- top_drawdowns_by_length[1:5,]
print(top_5_drawdowns_by_length)
```

```
##          From     Trough         To   Depth Length To Trough Recovery
## 2 2000-09-01 2009-03-01 2013-02-01 -0.5311    150       103       47
## 1 1929-10-01 1932-06-01 1945-04-01 -0.8279    187        33      154
## 3 1973-02-01 1974-12-01 1985-06-01 -0.4732    149         23       126
## 4 1969-01-01 1970-06-01 1972-11-01 -0.3240     47         18        29
## 5 1987-09-01 1987-12-01 1991-04-01 -0.2746     44          4        40
```

```
#print(top_drawdowns)
```

(1). 2000-09-01(peak) to 2009-03-01(trough) to 2013-02-01(recovery), Dot-Com Bubble crisis. Downturn duration 103 months, recovery time 47 months.

(2). 1929-10-01(peak) to 1932-06-01(trough) to 1945-04-01(recovery), Great Depression. Downturn duration 33 months, recovery time 154 months.

(3). 1973-02-01 (peak) to 1974-12-01(trough) to 1985-06-01(recovery), Oil Crisis & stagflation. Downturn duration 23 months, recovery time 126 months.

(4). 1969-01-01 (peak) to 1970-06-01(trough) to 1972-11-01 (recovery), Late 60s Recession & stagflation. Downturn duration 18 months, recovery time 29 months.

(5). 1987-09-01(peak) to 1987-12-01(trough) to 1991-04-01(recovery), Black Monday. Downturn duration 4 months, recovery time 40 months.

g-1. Reports Pearson and Spearman correlations

```
pearson_corr_xr <- cor(monthly_sp500_excess_returns$monthly_xr,
                    monthly_sp500_excess_returns$rme,
                    method = "pearson", use = "complete.obs")

# Calculate Spearman correlation
spearman_corr_xr <- cor(monthly_sp500_excess_returns$monthly_xr,
                     monthly_sp500_excess_returns$rme,
                     method = "spearman", use = "complete.obs")

# Print the correlations
cat("Pearson Correlation:", pearson_corr_xr, "\n")
```

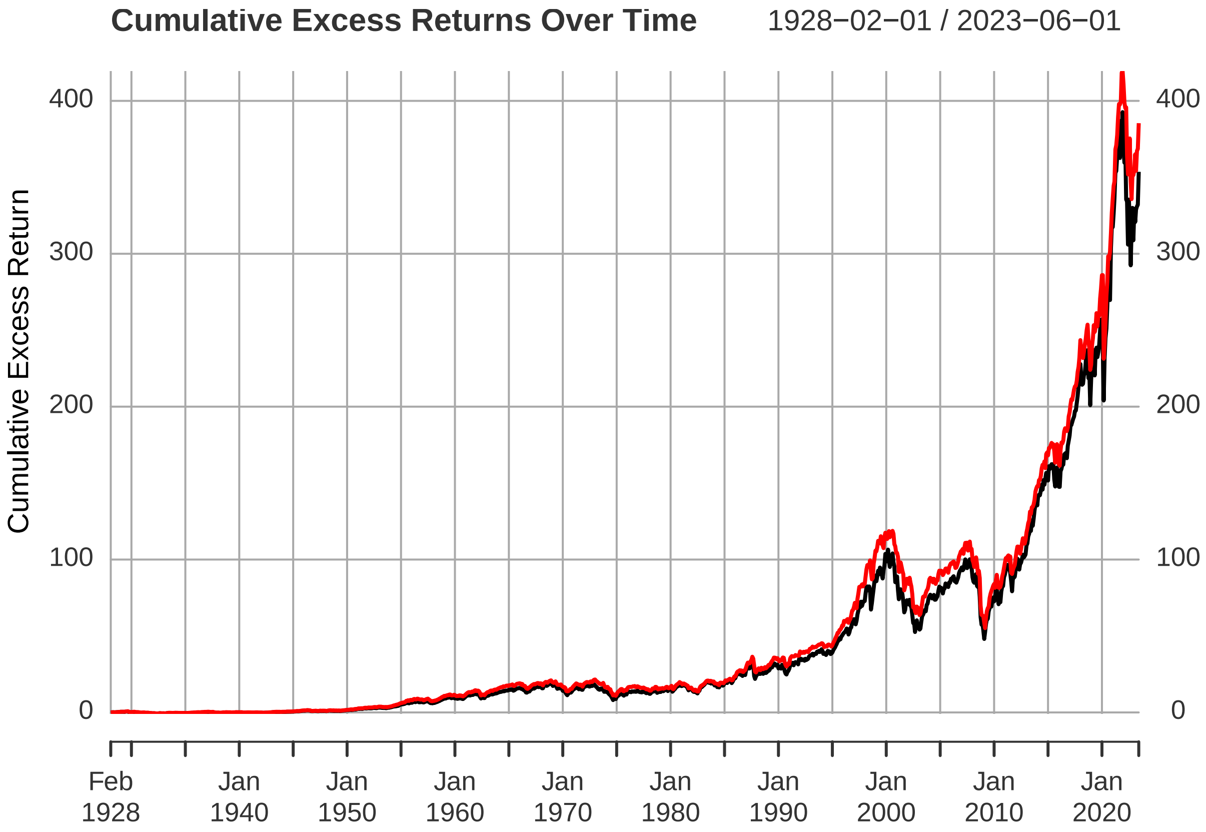
```
## Pearson Correlation: 0.6731458
```

```
cat("Spearman Correlation:", spearman_corr_xr, "\n")
```

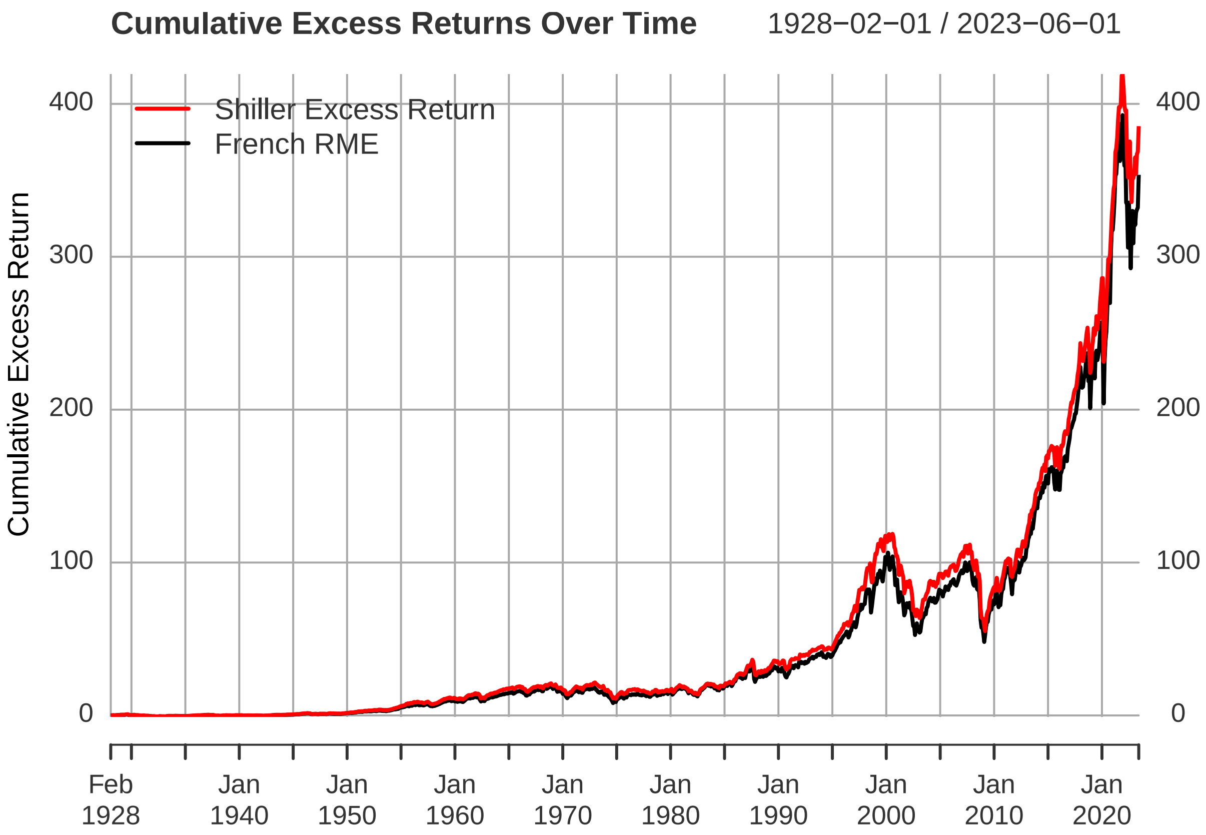
```
## Spearman Correlation: 0.6351528
```

g-2. Plots time series of cumulative excess returns for the two indices

```
cum_xr_xts <- xts(monthly_sp500_excess_returns$cum_xr, order.by =
                            monthly_sp500_excess_returns$year_month)
cum_rme_xts <- xts(monthly_sp500_excess_returns$cum_rme, order.by =
                            monthly_sp500_excess_returns$year_month)

cum_xr_xts_merged <- merge(cum_xr_xts, cum_rme_xts, all = TRUE)

plot.xts(cum_xr_xts_merged, main = "Cumulative Excess Returns Over Time", ylab = "Cumulative Excess Retu
         xlab = "Date", col = c("red", "black"))
```



```
addLegend("topleft", legend.names = c("Shiller Excess Return", "French RME"),
          col = c("red", "black"), lty = 1, lwd = 2)
```



h. Identifies 3 longest downturns in duration, along with time of recovery

```
top_3_drawdowns_by_length <- top_drawdowns_by_length[1:3, ]
print(top_3_drawdowns_by_length)
```

```
##         From     Trough         To   Depth Length To Trough Recovery
## 2 2000-09-01 2009-03-01 2013-02-01 -0.5311    150       103       47
## 1 1929-10-01 1932-06-01 1945-04-01 -0.8279    187        33      154
## 3 1973-02-01 1974-12-01 1985-06-01 -0.4732    149        23      126
```

(1). 2000-09-01(peak) to 2009-03-01(trough) to 2013-02-01(recovery), Dot-Com Bubble crisis. Downturn duration 103 months, recovery time 47 months.

(2). 1929-10-01(peak) to 1932-06-01(trough) to 1945-04-01(recovery), Great Depression. Downturn duration 33 months, recovery time 154 months.

(3). 1973-02-01 (peak) to 1974-12-01(trough) to 1985-06-01(recovery), Oil Crisis & stagflation. Downturn duration 23 months, recovery time 126 months.