Financial Data Science_PS1

Yihan Li

2023-09-18

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.1
## v readr   2.1.2     v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(lubridate)
```

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

```
rm(list=ls())
```

Question 1

The longest available data is from 1927/12/30 on Bloomberg and I have extracted the dataset from 1927/12/30 to 2023/8/31. Nevertheless, the dataset is not complete. I found that data before 1982/4/21 are the same for open, high, low and last, so I trimmed the data to range from 1982/4/21 to 2023/8/31.

1a

```
SP_500 <- read.csv("S&P 500.csv", header = TRUE)

# convert calendar date into date type
SP_500 <- SP_500 %>%
  mutate(date = as.Date(Dates))

head(SP_500)
```

```
##        Dates PX_OPEN PX_HIGH PX_LOW PX_LAST       date
## 1 1927/12/30   17.66   17.66  17.66   17.66 1927-12-30
## 2   1928/1/3   17.76   17.76  17.76   17.76 1928-01-03
## 3   1928/1/4   17.72   17.72  17.72   17.72 1928-01-04
## 4   1928/1/5   17.55   17.55  17.55   17.55 1928-01-05
```

```
## 5   1928/1/6   17.66   17.66  17.66   17.66 1928-01-06
## 6   1928/1/9   17.50   17.50  17.50   17.50 1928-01-09
```

```
# check the number of duplicated rows in dataframe
sum(ifelse(SP_500$PX_OPEN == SP_500$PX_HIGH &
           SP_500$PX_OPEN == SP_500$PX_LOW &
           SP_500$PX_OPEN == SP_500$PX_LAST,1,0))
```

```
## [1] 13604
```

```
# filter out dates with duplicated data
SP_500_new <- SP_500[13604:nrow(SP_500), ]
```

1b

Data Wrangling & Integrity Checks

```
SP_500_new <- SP_500_new %>%
  na_if("") %>% # check for missing data, convert missing data to NA
  drop_na() # drop NA values

# check if the high value is greater than low value
highlow <- SP_500_new$PX_HIGH >= SP_500_new$PX_LOW
length(highlow[highlow==FALSE])
```

```
## [1] 0
```

In the previous part 1a), I found that the dataframe is not complete and corrected the dataframe to include only complete values. In part 1b), I further investigate the blankness and accurancy of the dataframe, determined by the criteria (high > low)

1c

```
probability_openhigh_equal <- sum(ifelse(SP_500_new$PX_OPEN == SP_500_new$PX_HIGH, 1, 0)) / nrow(SP_500_
```

The probability is 11.064%

Methodology: find the number of days when open price equals to high price. Divide the days by the total number of days in the dataframe.

Assumption: high price equals open price if the stock market performs badly on a given day. In other words, the stock prices will never exceed the open price in a day. We should expect Prob(open=high) to be low as the stock market (S&P) had been growing in the past 40 years.

Testing Random Walk Hypothesis Intraday: cannot be used to test for Random Walk Hypothesis. If the highest value of S&P 500 occurs at the opening of a day, it means that during that day S&P 500 will decrease in value as long as the market opens, and fluctuates below the opening price. One plausible explanation is that market has sentiment and herd effect exists. If S&P 500 continues to drop intraday, it is like that investors are selling out their assets in the market that further drive down the prices. On the other hand, if random walk is true, we should expect the stock price to move in random directions and independent of past performance and sentiment in a day.

1d

```
# create a dataset ranging from 1982/4/21 to 2011/8/31 because data before 1982/4/21 have no intraday m

SP_500_intraday <- SP_500_new[SP_500_new$Dates <= "2011-8-31", ] %>%
  mutate(range = (PX_HIGH - PX_LOW)/PX_LOW)
```

```
SP_500_intraday_top20 <- SP_500_intraday %>%
  arrange(desc(range)) %>%
  slice(1:20) %>%
  arrange(date) %>%
  mutate(last_3yr = ifelse(
    date >="2008-9-1" & date <= "2011-8-31",1,0))

sum(SP_500_intraday_top20$last_3yr)
```

```
## [1] 15
```

15 days took place between 2008/9/1 and 2011/8/31

1e

```
SP_500_open <- SP_500_intraday %>%
  select("date","PX_OPEN")

SP_500_close_before <- SP_500_intraday[1:nrow(SP_500_intraday)-1,] %>%
  select("PX_LAST")

SP_500_close_before <- rbind(NA, SP_500_close_before)

SP_500_overnight <- cbind(SP_500_open,SP_500_close_before) %>%
  mutate(overnight = (PX_OPEN - PX_LAST) / PX_LAST) %>%
  mutate(year = as.numeric(format(date,'%Y')))

top20_negative <- SP_500_overnight %>%
  arrange(overnight) %>%
  slice(1:20) %>%
  arrange(date) %>%
  group_by(year) %>%
  count()

top20_positive <- SP_500_overnight %>%
  arrange(desc(overnight)) %>%
  slice(1:20) %>%
  arrange(date) %>%
  group_by(year) %>%
  count()

top20_negative
```

```
## # A tibble: 6 x 2
## # Groups:   year [6]
##    year     n
##   <dbl> <int>
## 1  1982     4
## 2  1983     2
## 3  1986     1
## 4  2008     6
## 5  2009     5
## 6  2010     2
```

```
top20_positive
```

```
## # A tibble: 8 x 2
## # Groups:   year [8]
##    year     n
##   <dbl> <int>
## 1  1982     8
## 2  1983     1
## 3  1984     1
## 4  1988     1
## 5  2006     1
## 6  2007     1
## 7  2008     6
## 8  2010     1
```

Most negative cases occur between 2008-2010 and most positive cases occur between 1982-1984

1f

```
# one-day jump j_t = r_t / sigma_t
# r_t = log(p_t/p_(t-1))
# sigma_t = SD of returns 30 days prior to the start of day t

SP_500_log_return <- SP_500_new %>%
  select("date","PX_LAST")

SP_close_prev <- SP_500_new[1:nrow(SP_500_new)-1,] %>%
  select("PX_LAST")

SP_close_prev <- rbind(NA,SP_close_prev)

SP_500_log_return <- cbind(SP_500_log_return, SP_close_prev)
names(SP_500_log_return)[3] <- "PX_PREV"
SP_500_log_return$return_log = log(SP_500_log_return$PX_LAST/SP_500_log_return$PX_PREV)
SP_500_log_return$sigmat <- NA

for (i in 32:nrow(SP_500_log_return)){
  sigma <- sd(as.numeric(unlist(SP_500_log_return$return_log))[(i-30):(i-1)])
  SP_500_log_return$sigmat[i] <- sigma
}

SP_500_log_return$jt <- SP_500_log_return$return_log / SP_500_log_return$sigmat

SP_500_log_return$abs_jt <- abs(SP_500_log_return$jt)

SP_500_log_return %>%
  arrange(desc(abs_jt)) %>%
  slice(1:20)
```

```
##          date PX_LAST PX_PREV  return_log      sigmat         jt    abs_jt
## 1  1987-10-19  224.84  282.70 -0.22899723 0.016055677 -14.262695 14.262695
## 2  1989-10-13  333.62  355.39 -0.06321316 0.005334318 -11.850280 11.850280
## 3  2018-10-10 2785.68 2880.34 -0.03341633 0.003689069  -9.058203  9.058203
## 4  2007-02-27 1399.04 1449.38 -0.03534959 0.004512128  -7.834351  7.834351
```

```
## 5  2016-09-09 2127.81 2181.30 -0.02482775 0.003290281  -7.545783  7.545783
## 6  1997-10-27  876.99  941.64 -0.07112745 0.009712788  -7.323072  7.323072
## 7  2018-02-05 2648.94 2762.13 -0.04184256 0.006535195  -6.402650  6.402650
## 8  2016-06-24 2037.41 2113.32 -0.03658078 0.005874157  -6.227409  6.227409
## 9  1986-09-11  235.18  247.06 -0.04928004 0.008465394  -5.821352  5.821352
## 10 1991-11-15  382.62  397.15 -0.03727171 0.006428108  -5.798240  5.798240
## 11 1982-08-17  109.04  104.09  0.04645888 0.008212196   5.657303  5.657303
## 12 2019-08-05 2844.74 2932.05 -0.03023016 0.005482278  -5.514161  5.514161
## 13 2020-09-03 3455.06 3580.84 -0.03575759 0.006634494  -5.389648  5.389648
## 14 1994-02-04  469.81  480.71 -0.02293582 0.004270938  -5.370206  5.370206
## 15 2000-01-04 1399.42 1455.22 -0.03909923 0.007552547  -5.176959  5.176959
## 16 1993-02-16  433.91  444.58 -0.02429288 0.004716208  -5.150934  5.150934
## 17 2001-09-17 1038.77 1092.54 -0.05046794 0.009914958  -5.090081  5.090081
## 18 2011-08-08 1119.46 1199.38 -0.06895833 0.013587833  -5.075006  5.075006
## 19 1998-08-31  957.28 1027.14 -0.07043759 0.014062992  -5.008720  5.008720
## 20 2021-11-26 4594.62 4701.46 -0.02298704 0.004815495  -4.773558  4.773558
```

Based on the analysis, 1 data fall between the range from 2008/8/31 to 2011/8/31, and this date is 2011-08-08

1g

I have picked the following periods as comparison with the pandemic year 2020. Since the article was wrote in 2011, I have combined the impact of global financial crisis and the "swings since the start of 2010" into one period.

1980-1990: price fluctuations of 1% or more is common 1990-2000: relative calm period 2000-2007: automated trading strategy, introduction of ETF 2008-2011: global financial crisis and aftermath 2020: global pandemic COVID-19

```
SP_500_new_2020 <- SP_500_new %>%
  filter(date >= "2020-01-01" & date <= "2020-12-31") %>%
  mutate(range = PX_HIGH - PX_LOW) %>%
  mutate(fluc = (PX_HIGH - PX_LOW)/PX_LOW)

SP_500_new_2020$PX_PREV <- append(NA,SP_500_new_2020$PX_LAST[1:nrow(SP_500_new_2020)-1])
SP_500_new_2020$abs_overnight <- abs((SP_500_new_2020$PX_OPEN - SP_500_new_2020$PX_PREV) / (SP_500_new_1

SP_500_new_80s <- SP_500_new %>%
  filter(date >= "1980-01-01" & date <= "1989-12-31") %>%
  mutate(range = PX_HIGH - PX_LOW) %>%
  mutate(fluc = (PX_HIGH - PX_LOW)/PX_LOW)

SP_500_new_80s$PX_PREV <- append(NA,SP_500_new_80s$PX_LAST[1:nrow(SP_500_new_80s)-1])
SP_500_new_80s$abs_overnight <- abs((SP_500_new_80s$PX_OPEN - SP_500_new_80s$PX_PREV) / (SP_500_new_80s$

SP_500_new_90s <- SP_500_new %>%
  filter(date >= "1990-01-01" & date <= "1999-12-31") %>%
  mutate(range = PX_HIGH - PX_LOW) %>%
  mutate(fluc = (PX_HIGH - PX_LOW)/PX_LOW)

SP_500_new_90s$PX_PREV <- append(NA,SP_500_new_90s$PX_LAST[1:nrow(SP_500_new_90s)-1])
SP_500_new_90s$abs_overnight <- abs((SP_500_new_90s$PX_OPEN - SP_500_new_90s$PX_PREV) / (SP_500_new_90s$

SP_500_new_00s <- SP_500_new %>%
  filter(date >= "2000-01-01" & date <= "2007-12-31") %>%
  mutate(range = PX_HIGH - PX_LOW) %>%
```

```
  mutate(fluc = (PX_HIGH - PX_LOW)/PX_LOW)

SP_500_new_00s$PX_PREV <- append(NA,SP_500_new_00s$PX_LAST[1:nrow(SP_500_new_00s)-1])
SP_500_new_00s$abs_overnight <- abs((SP_500_new_00s$PX_OPEN - SP_500_new_00s$PX_PREV) / (SP_500_new_00s$

SP_500_new_2010 <- SP_500_new %>%
  filter(date >= "2008-01-01" & date <= "2011-12-31") %>%
  mutate(range = PX_HIGH - PX_LOW) %>%
  mutate(fluc = (PX_HIGH - PX_LOW)/PX_LOW)

SP_500_new_2010$PX_PREV <- append(NA,SP_500_new_2010$PX_LAST[1:nrow(SP_500_new_2010)-1])
SP_500_new_2010$abs_overnight <- abs((SP_500_new_2010$PX_OPEN - SP_500_new_2010$PX_PREV) / (SP_500_new_
```

```
SP_500_new_2020 <- SP_500_new_2020%>%
  drop_na()

col1 <- c('Period', 'Average Intra-day Range', 'Average Intra-day Fluctuation', 'Average Overnight Chang

col2 <- c('2020',
          mean(SP_500_new_2020$range),
          mean(SP_500_new_2020$fluc),
          mean(SP_500_new_2020$abs_overnight))

SP_500_new_80s <- SP_500_new_80s%>%
  drop_na()
col3 <- c('1980s',
          mean(SP_500_new_80s$range),
          mean(SP_500_new_80s$fluc),
          mean(SP_500_new_80s$abs_overnight))

SP_500_new_90s <- SP_500_new_90s%>%
  drop_na()
col4 <- c('1990s',
          mean(SP_500_new_90s$range),
          mean(SP_500_new_90s$fluc),
          mean(SP_500_new_90s$abs_overnight))

SP_500_new_00s <- SP_500_new_00s%>%
  drop_na()
col5 <- c('2000s',
          mean(SP_500_new_00s$range),
          mean(SP_500_new_00s$fluc),
          mean(SP_500_new_00s$abs_overnight))

SP_500_new_2010 <- SP_500_new_2010%>%
  drop_na()
col6 <- c('2008-2011',
          mean(SP_500_new_2010$range),
          mean(SP_500_new_2010$fluc),
          mean(SP_500_new_2010$abs_overnight))

data_volatility <- data.frame(col1,col2,col3,col4,col5,col6); data_volatility
```

```
##                          col1               col2               col3
```

```
## 1                         Period                2020                1980s
## 2       Average Intra-day Range    51.8149206349206     2.56779147406266
## 3 Average Intra-day Fluctuation   0.017320670658361   0.0116238628007933
## 4      Average Overnight Change 0.00692460300078875 0.000288628634434155
##                   col4                 col5                col6
## 1                1990s                2000s           2008-2011
## 2    7.91694895132568      16.203618715779    21.0322023809524
## 3  0.0111352418898106   0.0137977221271435  0.0196291058048665
## 4 5.2448458554831e-05 7.30854362543755e-05 0.00145443330074232
```

I have used three metrics to evaluate the volatility of S&P 500 in five periods. The first one is the average intra-day range in a period, which measures the difference between high and low price of S&P 500. By looking into the chart, we can see the year 2020 has witnessed the largest value for this metric, followed by the 2008-2011 Global Financial Crisis period. The second term I adopted is the average intra-day fluctuation that calculate the percentage change from low to high price in a period. The data cohort of 2008-2011 has the biggest percent change of 1.96%, while 2020 comes as the second largest at 1.73%. Lastly, I use the overnight change to reflect the overnight change in stock price, and 2020 data has the largest overnight change of 0.692%.

Question 2

2

Investigating the reasons behind conflicting results for S&P 500 OHLC data

```
SP_500_19821006 <- SP_500_new %>%
  filter(date == "1982-10-06")

SP_500_19821006
```

```
##       Dates PX_OPEN PX_HIGH PX_LOW PX_LAST       date
## 1 1982/10/6     122  126.97    122  126.97 1982-10-06
```

On Yahoo finance, the data is given by: OPEN: 122 HIGH: 125.97 LOW: 122 LAST: 125.97

While open and low prices are the same, high and low prices are off by 1. Even though the amount is not large, this discrepancy still carries significant implications because this could create arbitrage opportunity, and yield massive financial gains/losses if under leverage. There are two ways we can use to resolve the discrepancy. First, resort to other data vendors' data and determine which number come up more often. This includes not only electronic data vendors, but also data sources on newspaper or publications by the stock exchanges. Moreover, root back to the logistics of the data formation and check if the methods different data vendors use are different. Use the most reliable one.

I believe Bloomberg's data are more accurate. This is because Bloomberg is a paid terminal that provides analytical and quantitative service to largely institutional clients, whereas Yahoo Finance is a free resource that target individual users. Therefore, Bloomberg's data are more credible as they will affect how institutional traders make trading decisions, and will bear the risk of lawsuits if the data are inaccurate. On the other hand, Yahoo Finance shoulders no legal liability as it is free, and will not impact the market business as much as Bloomberg.

Question 3

On August 24, 2020, Salesforce.com (CRM) will replace Exxon Mobil, Amgen (AMGN) will replace Pfizer and Honeywell International (HON) will replace Raytheon Technologies. I have extracted the closing price of the components of Dow Jones from 2020/8/3 to 2020/8/31

```
dow_jones <- read.csv("Dow Jones.csv", header = TRUE)
head(dow_jones)
```

```
##       Dates INDU.Index UNH.US.Equity MSFT.US.Equity HD.US.Equity GS.US.Equity
## 1  2020/8/3   26664.40        303.61         216.54       266.18       199.39
## 2  2020/8/4   26828.47        304.50         213.29       267.87       201.64
## 3  2020/8/5   27201.52        312.47         212.94       267.48       204.52
## 4  2020/8/6   27386.98        314.06         216.35       269.37       204.25
## 5  2020/8/7   27433.48        317.03         212.48       271.64       208.27
## 6 2020/8/10   27791.44        319.10         208.25       274.73       209.38
##   CAT.US.Equity MCD.US.Equity V.US.Equity BA.US.Equity AAPL.US.Equity
## 1        131.78        194.40      190.69       162.27        108.938
## 2        131.52        199.36      192.29       165.07        109.665
## 3        134.97        199.26      196.10       174.28        110.063
## 4        134.39        203.18      198.77       172.20        113.903
## 5        134.92        204.60      196.36       170.02        111.113
## 6        142.02        204.12      196.79       179.41        112.728
##   CVX.US.Equity WMT.US.Equity JNJ.US.Equity TRV.US.Equity AXP.US.Equity
## 1         84.81        129.30        147.35        114.40         93.54
## 2         86.49        131.64        147.22        113.10         93.19
## 3         87.20        129.81        148.40        115.08         95.39
## 4         87.47        129.35        147.55        113.74         95.92
## 5         86.80        129.97        148.60        117.36         99.16
## 6         89.73        131.88        148.03        119.12        101.62
##   PG.US.Equity IBM.US.Equity JPM.US.Equity MRK.US.Equity MMM.US.Equity
## 1       131.29      118.7364         96.10       78.7040        150.41
## 2       133.79      120.1978         95.55       77.8745        151.21
## 3       133.44      119.8253         97.21       77.8459        155.35
## 4       132.71      120.4653         97.24       77.2833        156.00
## 5       133.55      119.3573         99.38       77.2547        158.33
## 6       134.10      121.4109        100.64       77.1498        161.44
##   NKE.US.Equity DIS.US.Equity KO.US.Equity CSCO.US.Equity DOW.US.Equity
## 1         98.33        116.35        46.30          47.16         40.09
## 2         97.33        117.29        46.69          47.67         41.32
## 3        100.94        127.61        47.22          47.33         41.99
## 4        100.45        130.82        47.48          47.77         42.05
## 5        101.86        129.93        47.80          47.43         42.63
## 6        105.41        128.79        47.72          47.73         44.80
##   INTC.US.Equity VZ.US.Equity WBA.US.Equity XOM.US.Equity PFE.US.Equity
## 1          48.30        57.24         41.08         42.25       36.3538
## 2          49.13        57.91         40.93         43.47       36.3918
## 3          48.92        57.54         40.81         43.85       36.4486
## 4          48.57        57.83         40.89         43.64       36.2780
## 5          48.03        58.53         41.52         43.44       36.4486
## 6          49.22        58.99         42.86         44.51       36.3918
##   RTX.US.Equity CRM.US.Equity AMGN.US.Equity HON.US.Equity AMZN.US.Equity
## 1         57.51        203.19         247.36        148.53        155.594
## 2         57.51        201.41         243.59        147.33        156.942
## 3         60.03        202.64         241.47        150.82        160.252
## 4         59.87        207.79         241.55        152.58        161.250
## 5         61.23        201.05         240.69        155.11        158.373
## 6         64.23        197.16         238.17        159.43        157.408
##   BRK.A.US.Equity
## 1        298800.0
## 2        300330.0
## 3        305200.0
## 4        307454.9
```

```
## 5        314333.9
## 6        318830.0
```

```
dow_jones <- dow_jones %>%
  mutate(date = as.Date(Dates))
```

3a

```
dow_jones_old <- dow_jones[,1:32]
dow_jones_old$marketcap <- rowSums(dow_jones_old[, 3:32])
dow_jones_old$index <-  dow_jones_old$marketcap / dow_jones_old$INDU.Index
dow_jones_old$index[16:21]
```

```
## [1] 0.1320496 0.1321321 0.1319942 0.1322303 0.1323276 0.1321402
```

```
dow_jones_new <- cbind(dow_jones[,1:29], dow_jones[33:35])
dow_jones_new$marketcap <- rowSums(dow_jones_new[, 3:32])
dow_jones_new$index <-  dow_jones_new$marketcap / dow_jones_new$INDU.Index
dow_jones_new$index[16:21]
```

```
## [1] 0.1483868 0.1494980 0.1514409 0.1518353 0.1516525 0.1516532
```

If the changes were made at the market close on the announcement date, the index divisor would have increased from 0.1321 to 0.1494 on August 25, 2020, the first effective trading day. The upshift of index divisor is not wanted as we wish to maintain a steady and stable index divisor at all time to avoid market fluctuations.

3b

```
dow_jones_old_0824 <- dow_jones_old[16,]
prop_old_departing = (dow_jones_old_0824$XOM.US.Equity + dow_jones_old_0824$PFE.US.Equity + dow_jones_ol

prop_old_departing
```

```
## [1] 0.03769762
```

```
prop_old_remaining <- 1-prop_old_departing; prop_old_remaining
```

```
## [1] 0.9623024
```

The departing companies make up for 3.769% of total index; the remaining 27 companies account for 96.23% of index.

3c

```
dow_jones_new_0824 <- dow_jones_new[16,]
prop_new = (dow_jones_new_0824$CRM.US.Equity + dow_jones_new_0824$AMGN.US.Equity + dow_jones_new_0824$HO

prop_new
```

```
## [1] 0.143646
```

```
prop_new_remaining <- 1-prop_new; prop_new_remaining
```

```
## [1] 0.856354
```

The newly introduced companies make up for 14.36% of total index; 27 remaining companies account for 85.63%, lesser total weight than previous value

3d

```
dow_jones_noamgn <- cbind(dow_jones_new[,1:30],dow_jones_new$HON.US.Equity)
dow_jones_amzn <- cbind(dow_jones_noamgn,dow_jones$AMZN.US.Equity)

dow_jones_amzn$marketcap <- rowSums(dow_jones_amzn[, 3:32])
dow_jones_amzn$index <-  dow_jones_amzn$marketcap / dow_jones_old$INDU.Index

amzn_divisor <- dow_jones_amzn$index[16]; amzn_divisor
```

```
## [1] 0.1459071
```

```
dow_jones_bh <- cbind(dow_jones_noamgn,dow_jones$BRK.A.US.Equity)
dow_jones_bh$marketcap <- rowSums(dow_jones_bh[, 3:32])
dow_jones_bh$index <-  dow_jones_bh$marketcap / dow_jones_bh$INDU.Index

bh_divisor <- dow_jones_bh$index[16]; bh_divisor
```

```
## [1] 11.40172
```

The divisor for amazon is 0.146 and the divisor for Berkshire Hathaway is 11.4

3e

```
dow_jones[21,]
```

```
##        Dates INDU.Index UNH.US.Equity MSFT.US.Equity HD.US.Equity GS.US.Equity
## 21 2020/8/31   28430.05        312.55         225.53       285.04       204.87
##    CAT.US.Equity MCD.US.Equity V.US.Equity BA.US.Equity AAPL.US.Equity
## 21        142.31        213.52      211.99       171.82         129.04
##    CVX.US.Equity WMT.US.Equity JNJ.US.Equity TRV.US.Equity AXP.US.Equity
## 21         83.93        138.85        153.41        116.04        101.59
##    PG.US.Equity IBM.US.Equity JPM.US.Equity MRK.US.Equity MMM.US.Equity
## 21       138.33      117.7813        100.19       81.3072        163.02
##    NKE.US.Equity DIS.US.Equity KO.US.Equity CSCO.US.Equity DOW.US.Equity
## 21        111.89        131.87        49.53          42.22         45.12
##    INTC.US.Equity VZ.US.Equity WBA.US.Equity XOM.US.Equity PFE.US.Equity
## 21          50.95        59.27         38.02         39.94        35.823
##    RTX.US.Equity CRM.US.Equity AMGN.US.Equity HON.US.Equity AMZN.US.Equity
## 21            61        272.65         253.32        165.55        172.548
##    BRK.A.US.Equity       date
## 21          327560 2020-08-31
```

By definition, a stock split does not change the shareholders' equity of a company. However, to the number of outstanding shares will increase n times while the share price will become 1/n of the original stock price. Apple experienced a 4-for-1 stock split that took effect on August 31, 2020 for shareholders of record on August 24, and may have the following impact on DJIA:

i. price adjustment: under apple's 4-for-1 stock split, shareholders would receive four shares for every unit share they previously held, and the stock price of apple per share would be 1/4. This price adjustment can affect the apple's performance on DJIA, as stock with higher prices will have a greater impact on the index

ii. maintaining continuity: apple's stock split doesn't change its weight much in the DJIA, as the latter is adjusted for such activities, but to ensure continuity in the DJIA, the Dow Jones Company may adjust the divisor of the index to compensate for stock splits. Meanwhile, the replacement of companies in

DJIA from August 24 to August 31 will also adjust the divisor to ensure continuity. The overlapping of both events contribute to the minimalization of fluctuations in the market.