Assignment

October 4, 2025

Contents

1 Part 1 Individual Stocks

Before messing around with the stock data, the environment should install and load the dplyr and lubridate packages to perform easier data analysis.

[ ]:
```
# installation
install.packages("dplyr")
install.packages("lubridate")
install.packages("ggplot2")
```

After installing them, the packages are loaded in.

[ ]:
```
#| warning: false
library(dplyr)
library(lubridate)
library(ggplot2)
```

Now let's start actually processing and transforming the stock data. First of all, the data must be *read* into existence, so that it becomes a workable dataframe. Also to ensure correct formatting, the `datadate` variable is adjusted to the standard date format.

[6]:
```
# load csv file as stock_data but clean the date format
stock_data = read.csv("compustat_food_bev.csv")
stock_data$datadate = as.Date(stock_data$datadate, format = "%d/%m/%Y")
```

With the stock data in place, only the relevant data set gets filtered out. Our team will focus on the following tickers: SBUX (Starbucks), WEN (Wendy's), PBPB (Potbelly), CMG (Chipotle), DPZ (Domino's Pizza)

1.1 SBUX (Starbucks)

[7]:
```
#   Pick ticker & prepare base frame -------------------------------------
stock_data_sbux <- stock_data %>%
  filter(tic == "SBUX") %>%
  arrange(datadate)
```

Inbefore tackling the tasks, some new columns must be created. These are going to be loaded in using the *mutate()* function with some metrics common and some exclusive to each stock. For Starbucks: daily return, 10-day momentum, daily range, MFV, month and year.

[8]:
```
#   Features 1-6 ---------------------------------------------------------
stock_data_sbux <- stock_data_sbux %>%
  arrange(datadate) %>%
  mutate(
    # 1) daily return
    return_daily    = (prccd - lag(prccd)) / lag(prccd),
    # 2) 10-day momentum
    momentum_10day = prccd - lag(prccd, 10),
    # 3) daily range
    range_daily     = prchd - prcld,
    # 4) MFV (guard against divide-by-zero)
    MFV = if_else(range_daily == 0 | is.na(range_daily),
                  NA_real_,
                  (((prccd - prcld) - (prchd - prccd)) / range_daily) * cshtrd),
    # 5-6) month & year (use new names to avoid clash with year())
    month_num = lubridate::month(datadate),
    year_num  = lubridate::year(datadate)
  )
```

[9]:

```
# 7) Total trading volume in June 2023
trad_vol_jun23 <- stock_data_sbux %>%
  filter(year_num == 2023, month_num == 6)
total_volume <- sum(trad_vol_jun23$cshtrd, na.rm = TRUE)
print(paste("The total trading volume in June 2023 was",
            total_volume, "shares"))

# 8) Mean daily return over the entire period
mean_daily_return <- mean(stock_data_sbux$return_daily, na.rm = TRUE)
print(paste("The mean daily return over the entire period was",
            round(mean_daily_return*100, 2), "%"))

# 9. Day with largest positive high price
largest_high_price <- filter(stock_data_sbux, prchd == max(prchd, na.rm = TRUE))
print(paste("The day with the largest high price was on",
            largest_high_price$datadate,
            "with a price of", largest_high_price$prchd))

# https://investor.starbucks.com/news/financial-releases/news-details/2021/
 ↪Starbucks-Reports-Record-Q3-Fiscal-2021-Results/default.aspx
# Corporate context: Starbucks Q3 FY21 earnings (announced July 27, 2021)␣
 ↪reported record revenue.
print("This spike was likely driven by investor optimism ahead of record Q3␣
 ↪earnings, released on July 27, 2021.")

# 10. Day with the largest positive daily return
largest_daily_return <- filter(stock_data_sbux, return_daily ==␣
 ↪max(return_daily, na.rm = TRUE))
print(paste("The day with the largest daily return occurred on",
            largest_daily_return$datadate,
            "at", round(largest_daily_return$return_daily*100, 2), "%"))

# https://investor.starbucks.com/news/financial-releases/news-details/2022/
 ↪Starbucks-Reports-Q2-Fiscal-2022-Results/default.aspx
# Corporate context: Starbucks Q2 FY22 earnings (announced May 3, 2022) showed␣
 ↪strong revenue growth.
print("This jump was driven by Starbucks' Q2 FY22 earnings release on May 3,␣
 ↪2022, which beat expectations.")
```

```
highest_return_SBUX = stock_data_sbux$return_daily[which.
 ↪max(stock_data_sbux$return_daily)]*100
```

```
[1] "The total trading volume in June 2023 was 151045270 shares"
[1] "The mean daily return over the entire period was 0.03 %"
[1] "The day with the largest high price was on 2021-07-23 with a price of
126.32"
[1] "This spike was likely driven by investor optimism ahead of record Q3
earnings, released on July 27, 2021."
[1] "The day with the largest daily return occurred on 2022-05-04 at 9.83 %"
[1] "This jump was driven by Starbucks' Q2 FY22 earnings release on May 3, 2022,
which beat expectations."
```

1.2 WEN (Wendy's)

[10]:

```
# construct new data table with stock data exclusively tied to Wendy's
stock_data_wen = filter(stock_data, tic=="WEN")
```

Columns for Wendy's: daily return, overnight return, volume change, MFV, month and year.

[11]:

```
#   Features 1-6 ------------------------------------------------------------
stock_data_wen <- stock_data_wen %>%
  arrange(datadate) %>%
  mutate(
    # 1) daily return
    retd = (prccd/lag(prccd))-1,
    # 2) overnight return
    retov = (prcod/lag(prccd))-1,
    # 3) volume change
    volch = (cshtrd/lag(cshtrd))-1,
    # 4) MFV (guard against divide-by-zero)
    mfv = ((prccd-prcld)-(prchd-prcld))/(prchd-prcld),
    # 5-6) month & year (use new names to avoid clash with year())
    mon = month(datadate),
    yr  = year(datadate)
  )
```

[12]:

```
# 7) Calculate the total trading volume, in June 2023.
tv0623 = sum(stock_data_wen$cshtrd[
    stock_data_wen$mon == 6 & stock_data_wen$yr==2023
    ])
print(paste("The total trading volume in June 2023 was",
            tv0623, "shares"))

# 8) Mean daily return over the entire period
return_daily_mean = mean(stock_data_wen$retd, na.rm = TRUE)
print(paste("The mean daily return over the entire period was",
            round(return_daily_mean*100, 2), "%"))

# 9) Day with largest positive high price
date_highest_high = stock_data_wen$datadate[
    which.max(stock_data_wen$prchd)
    ]
price_highest = stock_data_wen$prchd[which.max(stock_data_wen$prchd)]
print(paste("The day with the largest high price was on",
            date_highest_high,
            "with a price of", price_highest))

# 10) Day with the largest positive daily return
date_highest_daily_return = stock_data_wen$datadate[
    which.max(stock_data_wen$retd)
    ]
retd_highest_p = stock_data_wen$retd[which.max(stock_data_wen$retd)]
print(paste("The day with the largest daily return occurred on",
            date_highest_daily_return,
            "at", round(retd_highest_p*100, 2), "%"))
```

```
[1] "The total trading volume in June 2023 was 54557454 shares"
[1] "The mean daily return over the entire period was 0.01 %"
[1] "The day with the largest high price was on 2021-06-08 with a price of
29.46"
[1] "The day with the largest daily return occurred on 2021-06-08 at 25.85 %"
```

1.3 PBPB (Potbelly)

[13]:
```
#head(stock_data)
stock_data_PBPB <- filter(stock_data, tic == "PBPB")
```

Columns for Potbelly: daily return, overnight return, price delta intraday, MFV, month and year.

[14]:
```
# 1) adding returns
stock_data_PBPB <- mutate(stock_data_PBPB, return = prccd/lag(prccd,1)-1)
# 2) adding overnight returns
stock_data_PBPB <- mutate(stock_data_PBPB, return_ovn = prcod/lag(prccd,1)-1)
# 3) adding price delta intraday
stock_data_PBPB <- mutate(stock_data_PBPB, px_delta_intra = prchd-prcld)
# 4) adding Money FLow Volume Indicator (MFV)
stock_data_PBPB <- mutate(stock_data_PBPB, MFV = ((prccd-prcld)-(prchd-prccd))/
 ↪px_delta_intra)
#head(stock_data_PBPB)

# 5-6) adding the month and year (lubridate already part of tidyverse)

stock_data_PBPB <- mutate(stock_data_PBPB, Month = month(datadate))
stock_data_PBPB <- mutate(stock_data_PBPB, Year = year(datadate))
```

```
#head(stock_data_PBPB)
```

[ ]:

```
# 7) Volume traded in June 2023
trad_vol_jun23 <- filter(stock_data_PBPB, Month == 6 & Year == 2023)
head(trad_vol_jun23)
print(paste(sum(trad_vol_jun23$cshtrd), " Total Common Shares Traded in June␣
  ↪2023"))

# 8) Average return over entire period
mean_daily_return <- mean(stock_data_PBPB$return, na.rm = TRUE)
print(paste("The mean Daily Return during the entire peroid was ",␣
  ↪round(mean_daily_return*100, 2),"%"))

# 9) Day with largest positive high price

largest_high_price <- filter(stock_data_PBPB, prchd == max(prchd))
head(largest_high_price)
print(paste("The day wiht the largest high price was on",␣
  ↪largest_high_price$datadate, "with a price of", largest_high_price$prchd))

# 10) Day with the largest positive daily return
largest_daily_return_date_PBPB = stock_data_PBPB$datadate[
    which.max(stock_data_PBPB$return)
    ]

largest_daily_return_PBPB = stock_data_PBPB$return[which.
  ↪max(stock_data_PBPB$return)]*100
print(paste("The Day with the largest daily return occured on",␣
  ↪largest_daily_return_date_PBPB, "at", round(largest_daily_return_PBPB, 2),␣
  ↪"%"))
```

1.4 CMG (Chipotle)

[16]:

```
#head(stock_data)
stock_data_CMG <- filter(stock_data, tic == "CMG")
```

Columns for Chipotle: daily return, volume change, 10-day momentum, MFV, month and year.

[17]:

```
# 1) Adding a daily return column
stock_data_CMG <- mutate(stock_data_CMG, return_daily = prccd/lag(prccd,1)-1)
# 2) Adding a volume change column
stock_data_CMG <- mutate(stock_data_CMG, change_volume = cshtrd - lag(cshtrd,␣
  ↪1))
# 3) Adding a 10-day momentum indicator column
stock_data_CMG <- mutate(stock_data_CMG, momentum_10day = prccd - lag(prccd,␣
  ↪10))
# 4) Adding a Money FLow Volume Indicator (MFV) column
```

```
stock_data_CMG <- mutate(stock_data_CMG, MFV = ((prccd-prcld)-(prchd-prccd))/
 ↪prchd-prcld)
# 5) Adding a column for the month
stock_data_CMG <- mutate(stock_data_CMG, Month = month(datadate))
# 6) Adding a column for the year
stock_data_CMG <- mutate(stock_data_CMG, Year = year(datadate))
```

[ ]:

```
# 7) Total volume traded in June 2023
trad_vol_jun23 <- filter(stock_data_CMG, Month == 6 & Year == 2023)
head(trad_vol_jun23)
print(paste("The total trading volume in June 2023 was ",␣
 ↪sum(trad_vol_jun23$cshtrd)))

# 8) Mean daily return over entire period
mean_daily_return <- mean(stock_data_CMG$return, na.rm = TRUE)
print(paste("The mean daily return during the entire period was ",␣
 ↪round(mean_daily_return*100, 2),"%"))

# 9) Day with largest positive high price
largest_high_price <- filter(stock_data_CMG, prchd == max(prchd))
head(largest_high_price)
print(paste("The day with the largest high price was on",␣
 ↪largest_high_price$datadate, "with a price of", largest_high_price$prchd))

#https://newsroom.chipotle.com/
 ↪2023-07-19-BOWLS-FOR-GOALS-IS-BACK-CHIPOTLE-TO-GIVE-AWAY-FREE-ENTREES-WHEN-THE-U-S-WOMENS-N
# On July 29, 2023, Chipotle gave free entrees for every goal scored by the US␣
 ↪Women's National Team
print(paste("On July 29, 2023, Chipotle gave free entrees for every goal scored␣
 ↪by the US Women's National Team."))

# 10) Day with the largest positive daily return
largest_daily_return_CMG <- filter(stock_data_CMG, return_daily ==␣
 ↪max(return_daily, na.rm = TRUE))
head(largest_daily_return_CMG)
print(paste("The Day with the largest daily return occured on",␣
 ↪largest_daily_return_CMG$datadate, "at",␣
 ↪round(largest_daily_return_CMG$return_daily*100, 2), "%"))

#https://newsroom.chipotle.com/
 ↪2022-07-26-CHIPOTLE-ANNOUNCES-SECOND-QUARTER-2022-RESULTS
# On July 26, 2022, Chipotle announced its second quarter 2022 results
print(paste("On July 26 2022, Chipotle announced a strong Q2 performance␣
 ↪despite inflation and consumer uncertainty."))
```

```
highest_return_CMG = stock_data_CMG$return_daily[which.
 ↪max(stock_data_CMG$return_daily)]*100
```

1.5 DPZ (Domino's Pizza)

[19]:
```
#head(stock_data)
stock_data_DPZ <- filter(stock_data, tic == "DPZ")
```

Columns for Domino's Pizza: daily return, volume change, close-open change, MFV, month and year.

[20]:
```
# 1) Add a column with the daily return
stock_data_DPZ <- mutate(stock_data_DPZ, return = prccd/lag(prccd,1) - 1)
# 2) Add a column with the volume change
stock_data_DPZ <- mutate(stock_data_DPZ, change_volume = cshtrd - lag(cshtrd,1))
# 3) Add a column with the close-open change
stock_data_DPZ <- mutate(stock_data_DPZ, change_close_open = prccd - prcod)
# 4) Add a column with the money flow volume indicator (MFV)
stock_data_DPZ <- mutate(stock_data_DPZ, MFV = ((prccd - prcld) - (prchd -␣
 ↪prccd)) / (prchd - prcld) * cshtrd)
# 5) Add a column the month
stock_data_DPZ <- mutate(stock_data_DPZ, Month = month(datadate))
# 6) Add a column the year
stock_data_DPZ <- mutate(stock_data_DPZ, Year = year(datadate))
```

[ ]:
```
# 7) The total traing volume in June 2023
trad_vol_jun23 <- filter(stock_data_DPZ, Month == 6 & Year == 2023)
head(trad_vol_jun23)
print(paste(sum(trad_vol_jun23$cshtrd, na.rm = TRUE), " Total Common Shares␣
 ↪Traded in June 2023"))

# 8) Mean return over entire period
mean_daily_return <- mean(stock_data_DPZ$return, na.rm = TRUE)
print(paste("The mean Daily Return during the entire period was ",␣
 ↪round(mean_daily_return * 100, 2), "%"))

# 9) Date with largest positive high price
largest_high_price <- filter(stock_data_DPZ, prchd == max(prchd, na.rm = TRUE))
head(largest_high_price)
print(paste("The day with the largest high price was on",␣
 ↪largest_high_price$datadate, "with a price of", largest_high_price$prchd))

# 10) Date with the largest positive daily return
largest_daily_return_DPZ <- filter(stock_data_DPZ, return == max(return, na.rm␣
 ↪= TRUE))
head(largest_daily_return_DPZ)
```

```
print(paste("The day with the largest daily return occurred on",␣
 ↪largest_daily_return_DPZ$datadate, "at",␣
 ↪round(largest_daily_return_DPZ$return * 100, 2), "%"))

# https://ir.dominos.com/news-releases/news-release-details/
 ↪dominos-pizzar-announces-second-quarter-2021-financial-resultschipo
# On July 21nd 2022, Domino's Pizza Announces Second Quarter 2021 Financial␣
 ↪Results
print(paste("On July 22nd 2021, Domino's Pizza Announces a quite strong second␣
 ↪quarter of 2021 financial results"))

highest_return_DPZ = stock_data_DPZ$return[which.max(stock_data_DPZ$return)]*100
```

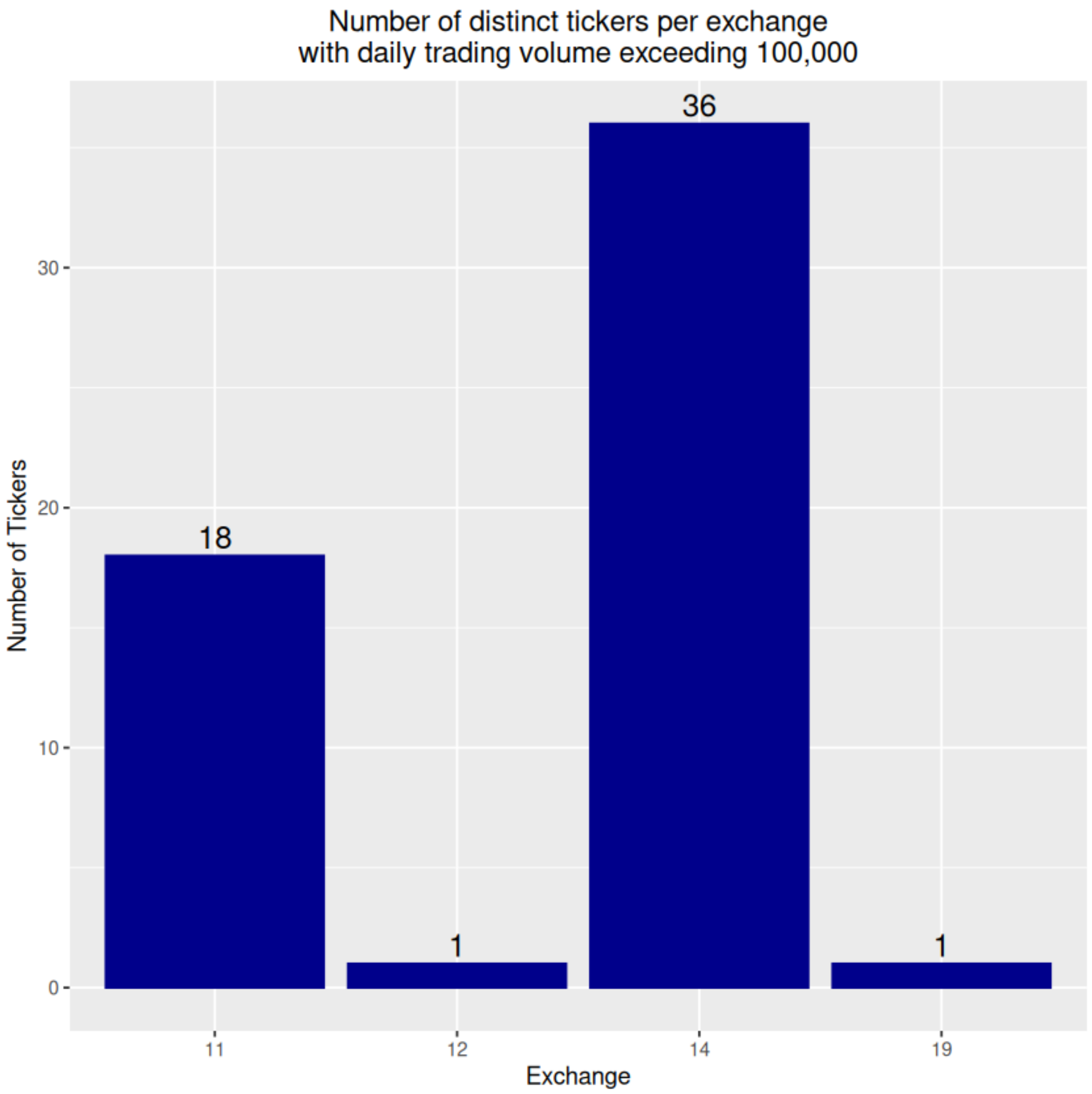
2 Part 2 Visualisations

2.1 Task 1

Visualise the number of tickers on each exchange that have had at least one trading day with a volume of more than 100000.

[42]:

```
#creating df with tickers on each exchange with one trading day of more than␣
 ↪100000
df_ticker_hvolume <- filter(stock_data, cshtrd>100000)
df_exch_tickr <- summarise(group_by(stock_data, exchg), unique_tic =␣
 ↪n_distinct(tic))

#plotting no of unique tickers on exchange with daily trading volume >100'000
ggplot(df_exch_tickr, aes(factor(exchg), unique_tic)) +
  geom_bar(stat = "identity", color = "darkblue", fill = "darkblue") +
  geom_text(aes(label = unique_tic),
            vjust = -0.3,              # position above the bar
            size = 5) +               # text size
  labs(title = "Number of distinct tickers per exchange\nwith daily trading␣
 ↪volume exceeding 100,000",
  x= "Exchange", y="Number of Tickers") +
  theme(plot.title = element_text(hjust = 0.5))
```



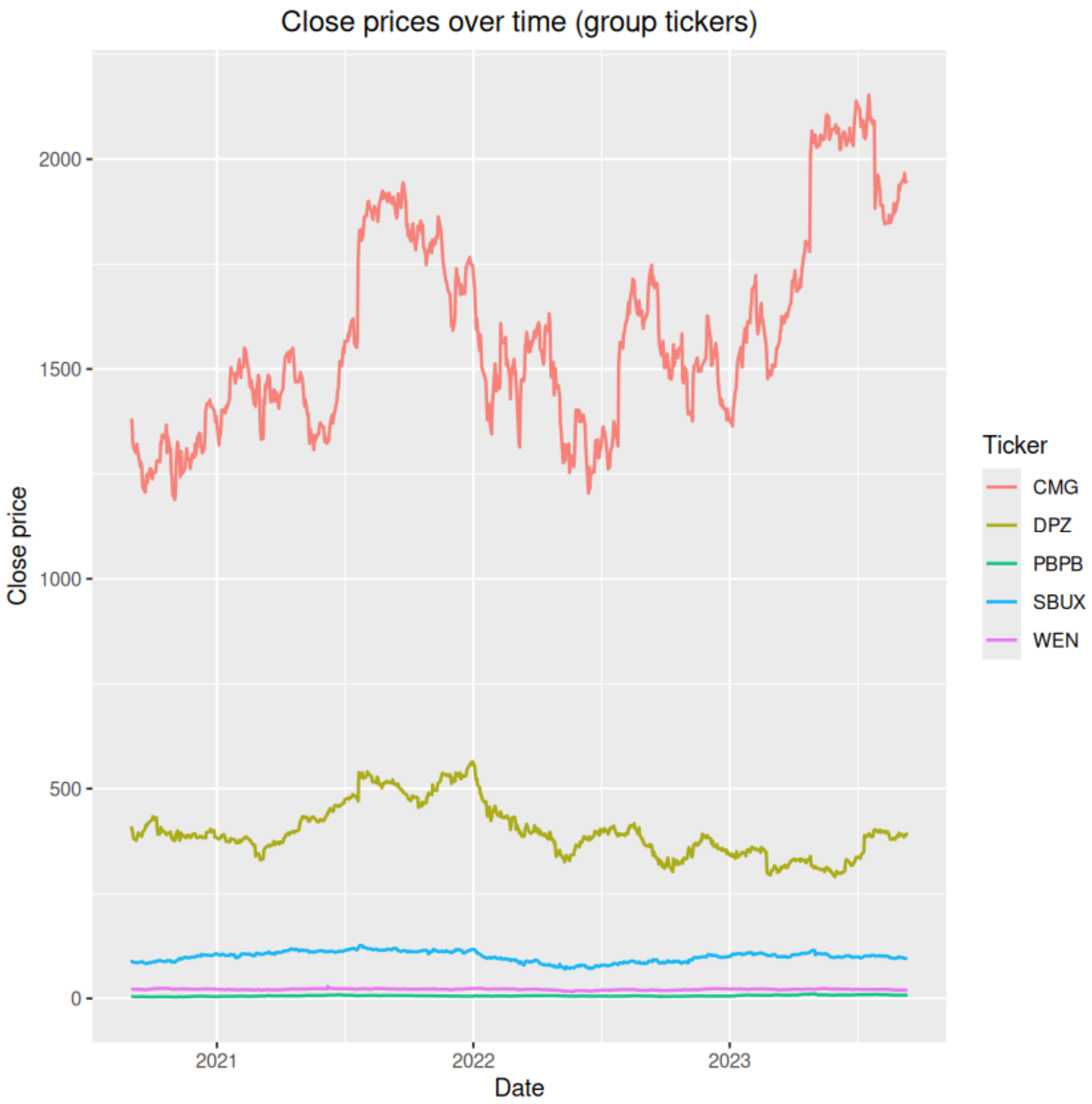
2.2 Task 2

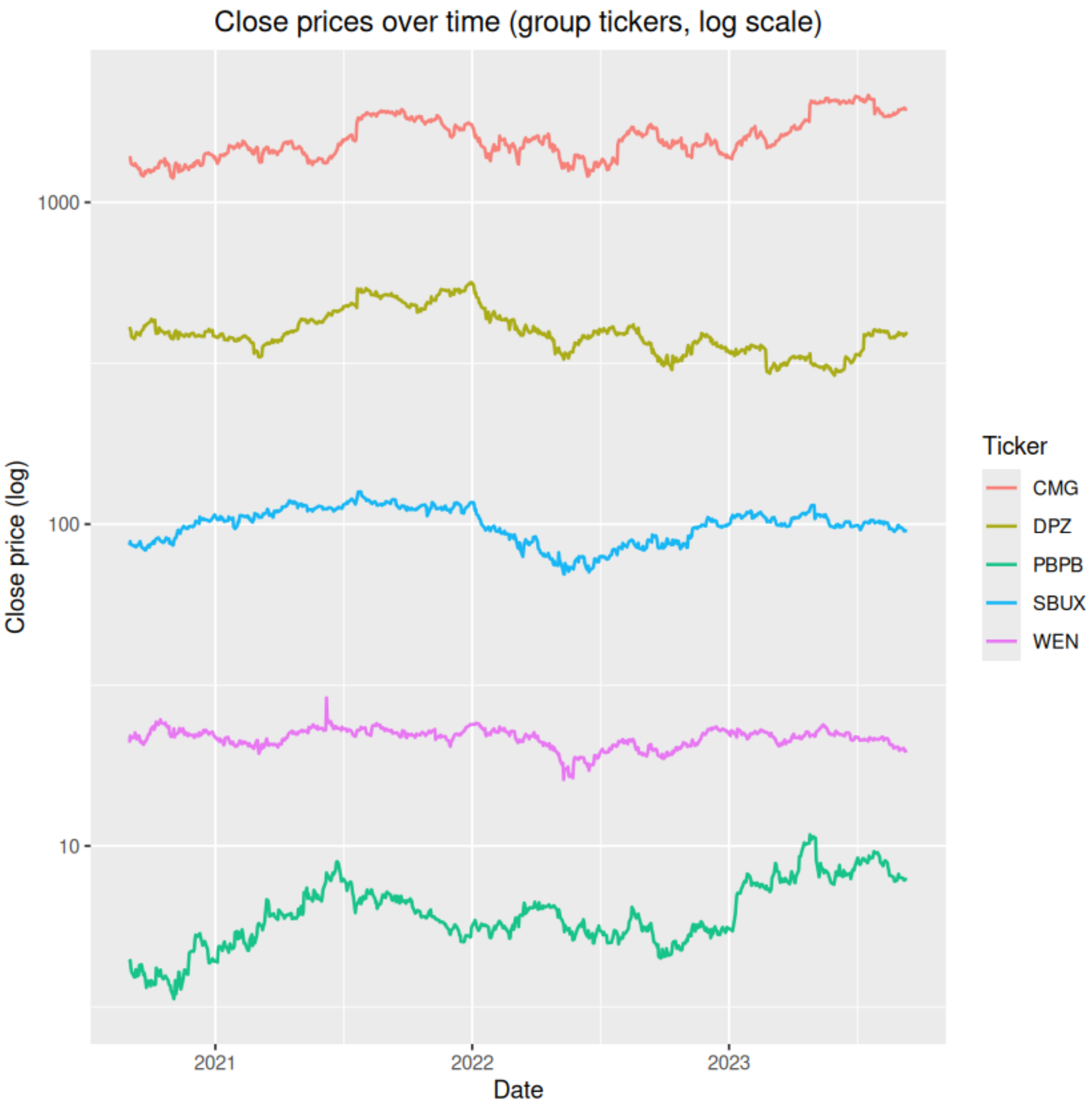
Visualise on one line plot the close prices of each ticker, over the period.

[43]:

```
# subset of our tickers
group_tickers <- c("SBUX", "WEN", "PBPB", "CMG", "DPZ")


df_group_raw <- stock_data %>%
  filter(tic %in% group_tickers) %>%
  arrange(datadate)

df_group <- df_group_raw %>%
```

```r
  filter(!is.na(prccd), prccd > 0)

# one line plot: close prices of each ticker over the period
ggplot(df_group, aes(x = datadate, y = prccd, color = tic)) +
  geom_line(size = 0.7, alpha = 0.9) +
  labs(
    title = "Close prices over time (group tickers)",
    x = "Date", y = "Close price", color = "Ticker"
  ) +
  theme(plot.title = element_text(hjust = 0.5))

# CMG prices are big numbers, thus use log
suppressWarnings(
ggplot(df_group, aes(x = datadate, y = prccd, color = tic)) +
  geom_line(size = 0.7, alpha = 0.9) +
  scale_y_log10() +                             # <-- compresses CMG's big numbers
  labs(
    title = "Close prices over time (group tickers, log scale)",
    x = "Date", y = "Close price (log)", color = "Ticker"
  ) +
  theme(plot.title = element_text(hjust = 0.5))
)
```


Close prices over time (group tickers)
Close price
2000
1500
1000
500
0
2021
2022
2023
Date
Ticker
CMG
DPZ
PBPB
SBUX
WEN



2.3 Task 3

For this task we tackle the ticker with the *highest mean daily return*, that is, Potbelly. Visualise on one line plot the high and low prices, in the year 2021.

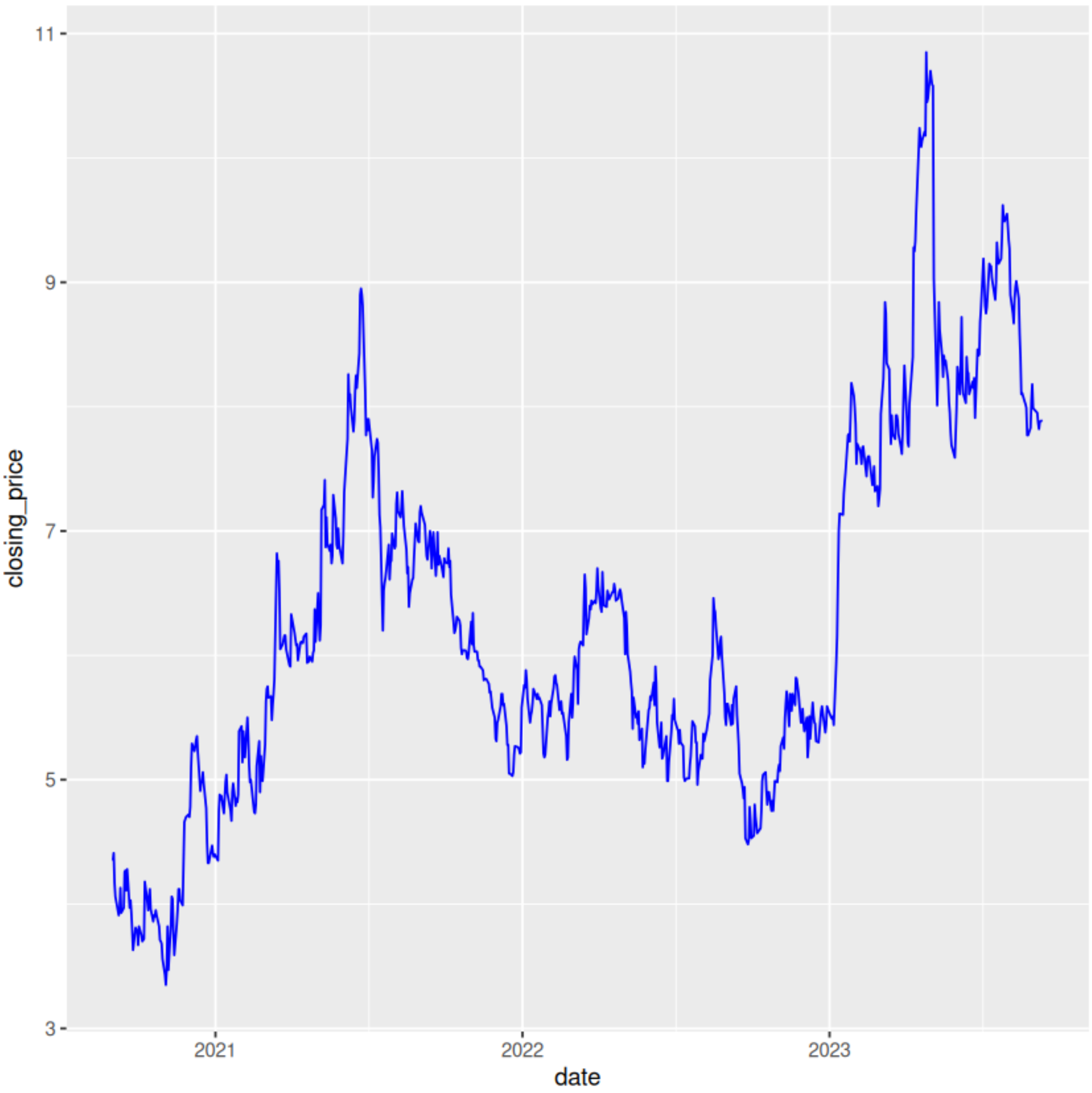
[30]:

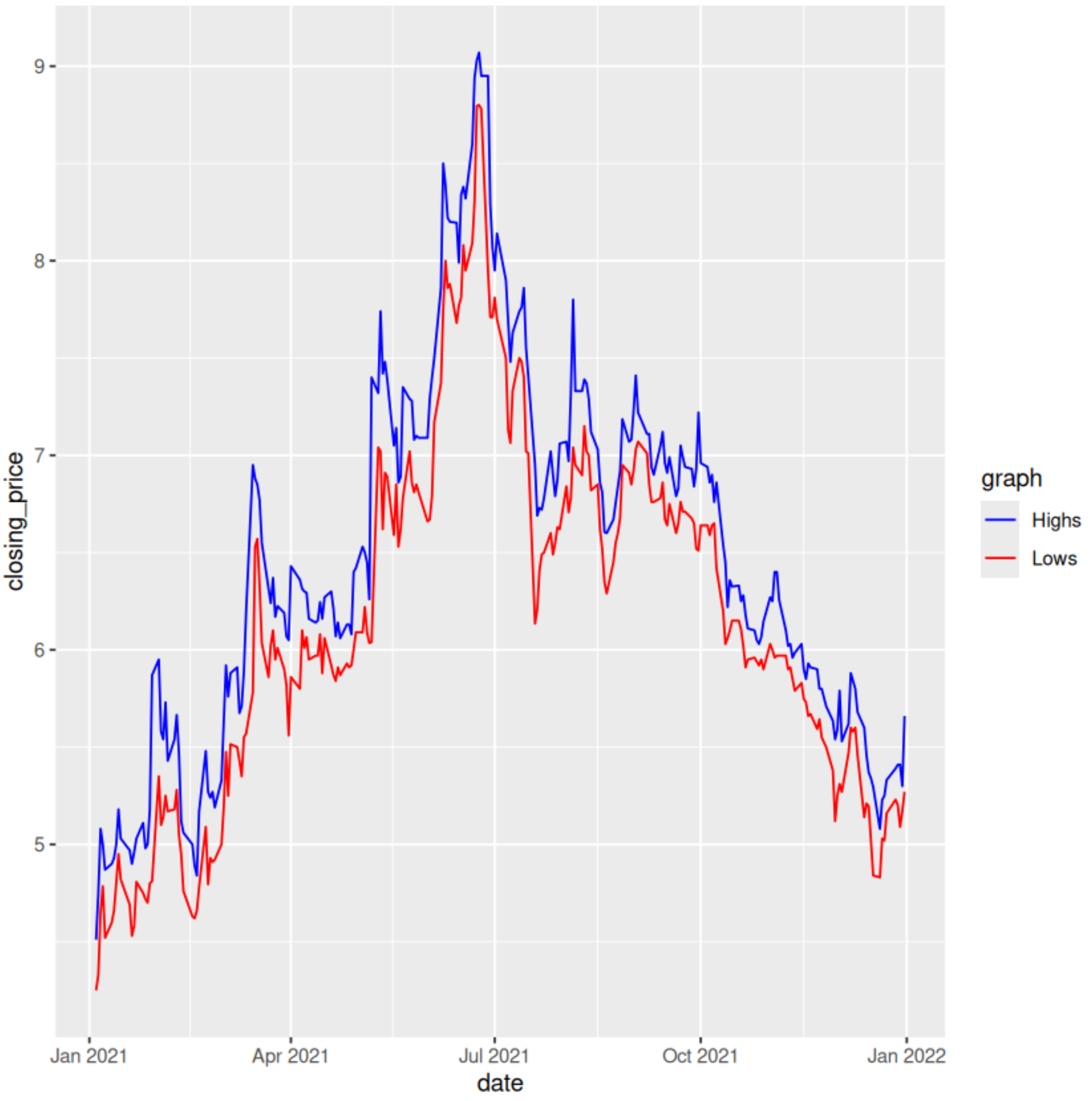
```r
# Step 3 WIP

returns_vec <- c(PBPB = largest_daily_return_PBPB, SBUX = highest_return_SBUX, CMG = highest_return_CMG, DPZ = highest_return_DPZ)
head(returns_vec)
ticker_largest_d_return <- names(which.max(returns_vec))
value_largest_d_return <- max(returns_vec)
```

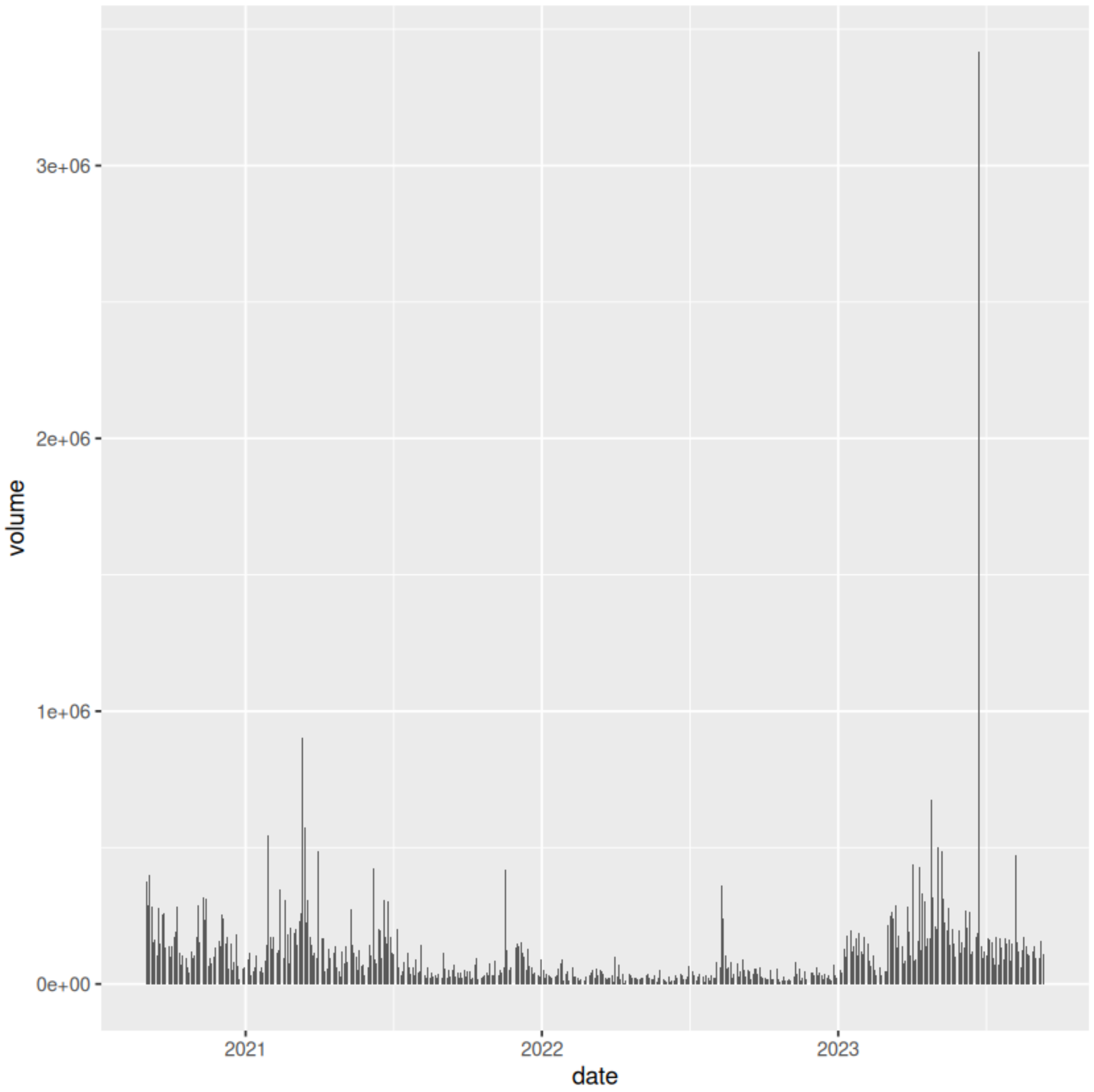
```r
print(paste("Among our five tickers, ", ticker_largest_d_return, "is the one with the largest daily return."))

# display the stock price chart for Potbelly
df_PBPB_prccd = data.frame(
   date = stock_data_PBPB$datadate,
   closing_price = stock_data_PBPB$prccd
   )

ggplot(df_PBPB_prccd, aes(x=date, y=closing_price)) +
   geom_line(color = "blue")

# display the stock price chart in terms of highs and lows for Potbelly
df_PBPB_highs = data.frame(
   date = stock_data_PBPB$datadate,
   closing_price = stock_data_PBPB$prchd,
   graph = "Highs"
   )

df_PBPB_highs <- subset(df_PBPB_highs, format(date, "%Y") == "2021")

df_PBPB_lows = data.frame(
   date = stock_data_PBPB$datadate,
   closing_price = stock_data_PBPB$prcld,
   graph = "Lows"
   )

df_PBPB_lows <- subset(df_PBPB_lows, format(date, "%Y") == "2021")

df_PBPB_high_low = rbind(df_PBPB_highs, df_PBPB_lows)
ggplot(df_PBPB_high_low, aes(x=date, y=closing_price, color=graph)) +
   geom_line() +
   scale_color_manual(values=c("Highs"="blue", "Lows"="red"))

# display the trading volume as a bar chart for Potbelly
df_PBPB_volume = data.frame(
   date = stock_data_PBPB$datadate,
   volume = stock_data_PBPB$cshtrd
   )

ggplot(df_PBPB_volume, aes(x=date, y=volume)) +
  geom_bar(stat = "identity", width=1)
```

PBPB 17.5862068965517 **SBUX** 9.83452172743173 **CMG** 14.7041620139316 **DPZ** 14.5523736632864

```
[1] "Among our five tickers,  PBPB is the one with the largest daily return."
```


closing_price
11
9
7
5
3
2021
2022
2023
date


9
8
7
6
5
closing_price
Jan 2021
Apr 2021
Jul 2021
Oct 2021
Jan 2022
date
graph
Highs
Lows


3e+06
2e+06
1e+06
0e+00
volume
2021
2022
2023
date