Group Assignment 1

Simon Yu

2025-10-05

Table of contents

List of Figures

List of Tables

```
library(dplyr)
library(ggplot2)
library(PerformanceAnalytics)
library(lubridate)
library(scales)
```

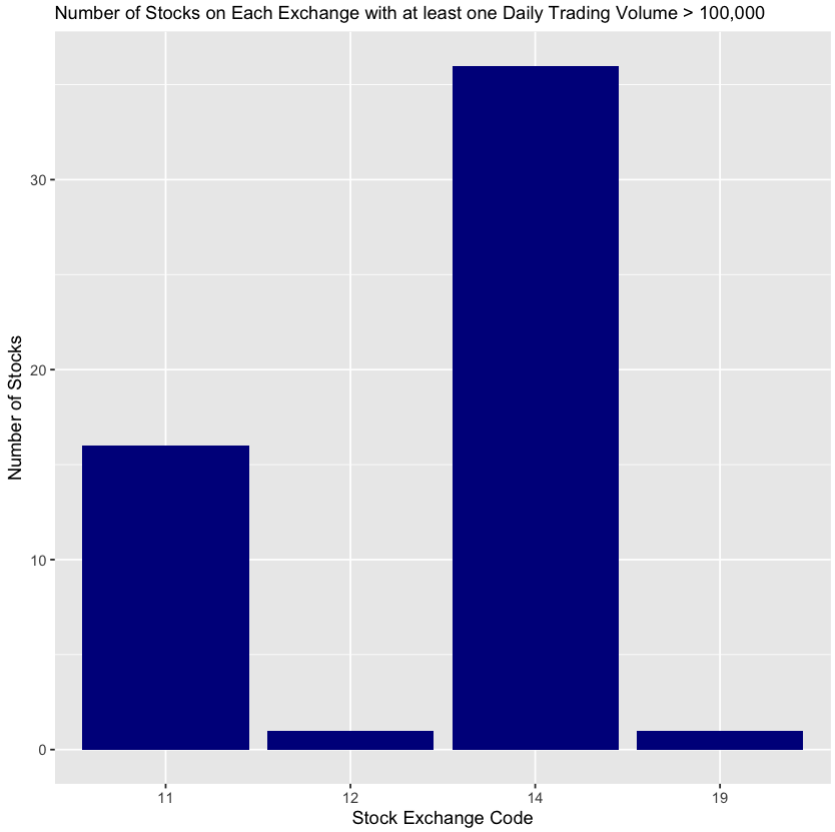
```
data <- read.csv("compustat_food_bev.csv")

# filter data for Starbucks (SBUX)
sbux_data <- filter(data, tic == "SBUX")

# filter data for Wendy's (WEN)
wen_data <- filter(data, tic == "WEN")

# filter data for Potbelly (PBPB)
pbpb_data <- filter(data, tic == "PBPB")

# filter data for Chipotle (SMG)
cmg_data <- filter(data, tic == "CMG")
```

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

```
# find all the data with daily trading volume (cshtrd) > 100,000
# group them by exchange and calulate the number of distinct tickers for each exchange
tickers_each_exchange <- data %>%
  filter(cshtrd > 100000) %>%
  group_by(exchg) %>%
  summarise(tic_num = n_distinct(tic))

# demonstrate the calulating result table
print(tickers_each_exchange)

# plot the result using column chart
ggplot(tickers_each_exchange, aes(factor(exchg), tic_num)) +
  geom_col(fill = "dark blue") +
  labs(subtitle = "Number of Stocks on Each Exchange with at least one Daily Trading Volume >
       x = "Stock Exchange Code", y = "Number of Stocks")
```

```
# A tibble: 4 × 2
```

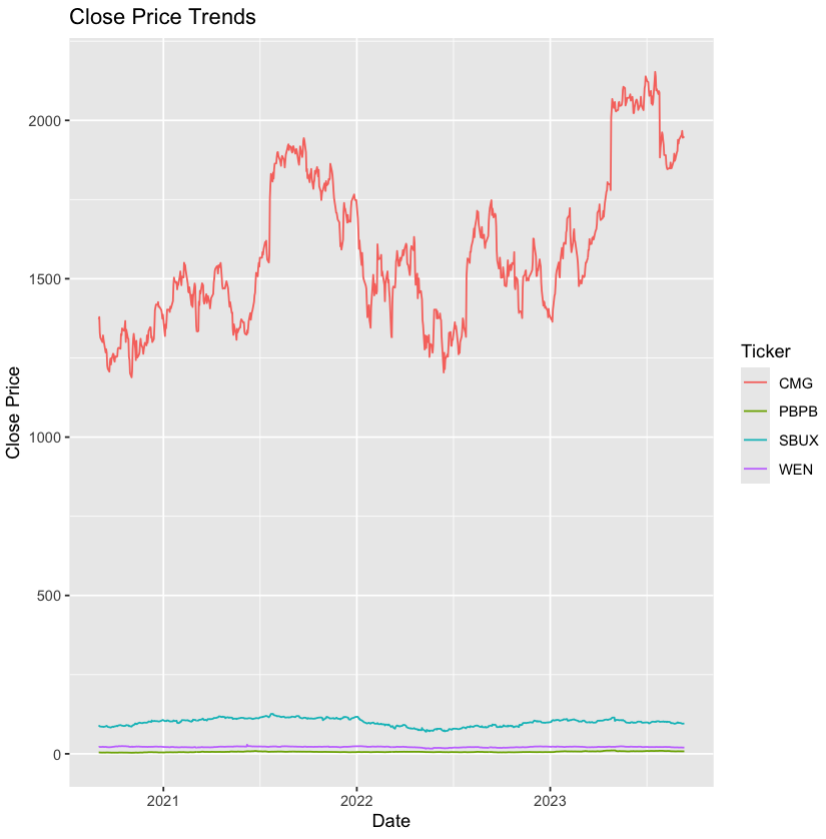
```
  exchg tic_num
  <int>   <int>
1    11      16
2    12       1
3    14      36
4    19       1
```

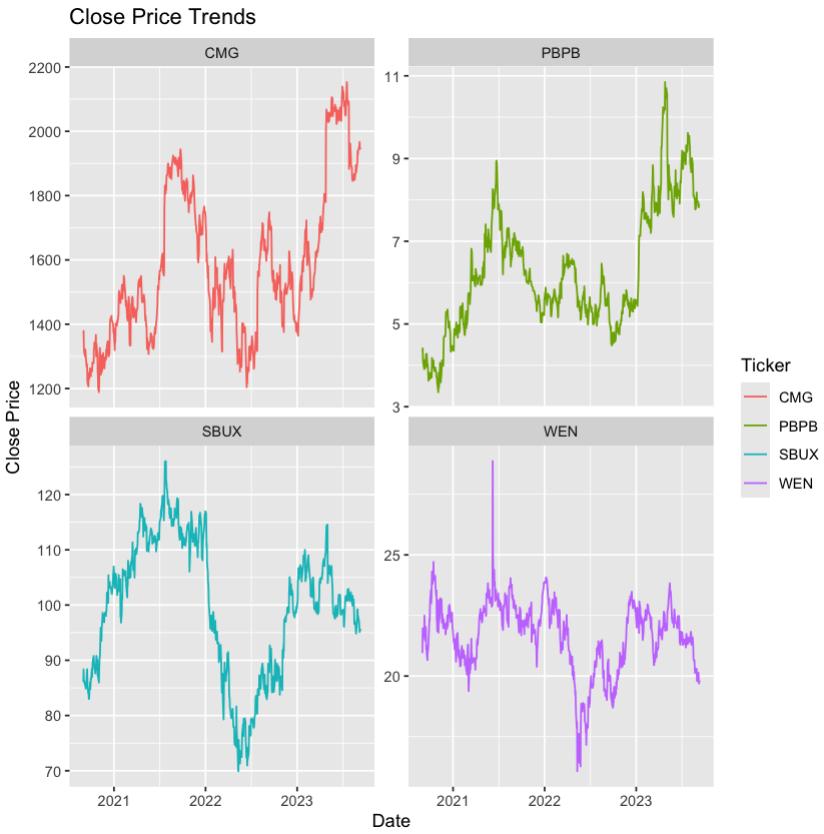


2. Visualize on one line plot the close prices of each ticker (SBUX, WEN, PBPB, CMG), over the period.

```r
# filter the tickers we analyzed, including SBUX, WEN, PBPB, CMG
subset <- filter(data, tic == "SBUX" | tic == "WEN"
                 | tic == "PBPB" | tic == "CMG")

# convert datadate to date format
subset$datadate <- as.Date(subset$datadate, format = "%d/%m/%Y")
```

```r
# plot the result using line plot
ggplot(subset, aes(datadate, prccd, colour = tic)) +
  geom_line() +
  labs(title = "Close Price Trends", colour = "Ticker",
       x = "Date", y = "Close Price")

# We can see that close price of CMG is far higher than
# the other three companies, making the line plot hard to read

# Therefore, we use facet_wrap to create separate plots for
# each ticker with free y scales
ggplot(subset, aes(datadate, prccd, colour = tic)) +
  geom_line() +
  facet_wrap(~ tic, scales = "free_y") +
  labs(title = "Close Price Trends", colour = "Ticker",
       x = "Date", y = "Close Price")
```





Finally, considering only the ticker you analysed with the highest mean daily return over the period:

3. Visualise on one line plot the high and low prices, in the year 2021.

```
# add a new column called daily_return for the tickers we analyzed
# drop the rows whose daily_return values are NA
sbux_data <- mutate(sbux_data, daily_return = (prccd - lag(prccd)) / lag(prccd))
sbux_data <- filter(sbux_data, !is.na(daily_return))

wen_data <- mutate(wen_data, daily_return = (prccd - lag(prccd)) / lag(prccd))
wen_data <- filter(wen_data, !is.na(daily_return))

pbpb_data <- mutate(pbpb_data, daily_return = (prccd - lag(prccd)) / lag(prccd))
pbpb_data <- filter(pbpb_data, !is.na(daily_return))
```

```r
cmg_data <- mutate(cmg_data, daily_return = (prccd - lag(prccd)) / lag(prccd))
cmg_data <- filter(cmg_data, !is.na(daily_return))

# calulate mean daily return for the tickers we analyzed
sbux_mean_daily_return <- mean(sbux_data$daily_return)
wen_mean_daily_return <- mean(wen_data$daily_return)
pbpb_mean_daily_return <- mean(pbpb_data$daily_return)
cmg_mean_daily_return <- mean(cmg_data$daily_return)

# demonstrate the calulating result
print(paste("mean daily return for SBUX:", sbux_mean_daily_return))
print(paste("mean daily return for WEN :", wen_mean_daily_return))
print(paste("mean daily return for PBPB:", pbpb_mean_daily_return))
print(paste("mean daily return for CMG :", cmg_mean_daily_return))
```

```
[1] "mean daily return for SBUX: 0.0002910467239313 76"
[1] "mean daily return for WEN : 0.0001164747127062 67"
[1] "mean daily return for PBPB: 0.0012798677677777 4"
[1] "mean daily return for CMG : 0.0006746876349519 14"
```

Apparnetly, PBPB has the highest mean daily return among the four tickers. Let's visualize the high and low prices of PBPB in 2021.

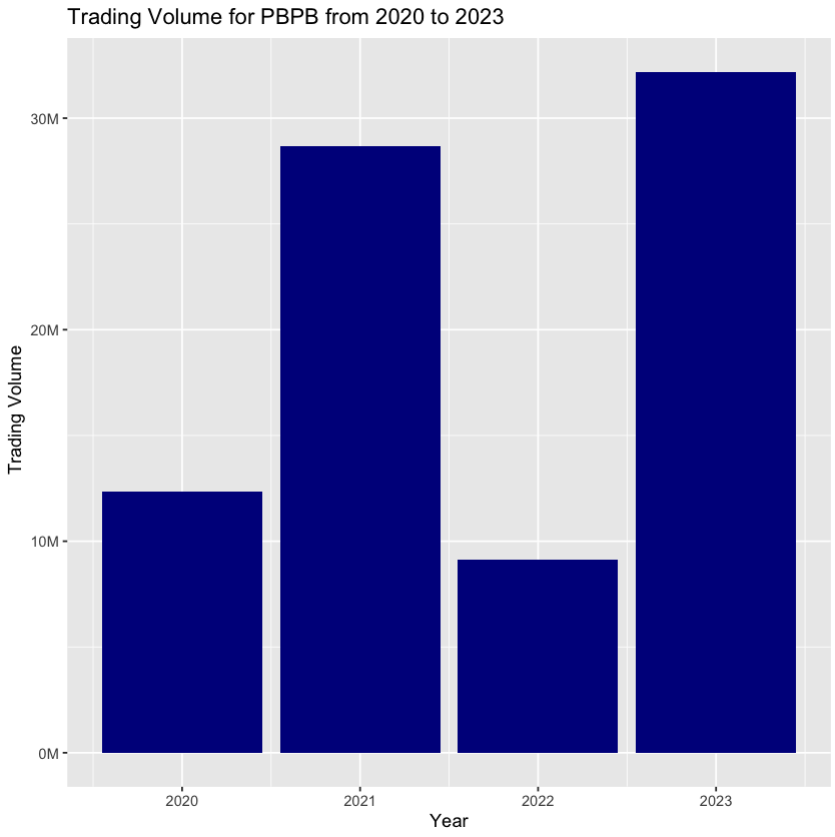
```r
# convert datadate to date format
pbpb_data$datadate <- as.Date(pbpb_data$datadate, format = "%d/%m/%Y")

# filter all the PBPB data in 2021
pbpb_data_2021 <- filter(pbpb_data, year(pbpb_data$datadate) == 2021)

# plot high and low prices using line plot
ggplot(pbpb_data_2021, aes(datadate, prchd, colour = "High Price")) +
  geom_line() +
  geom_line(aes(datadate, prcld, colour = "Low Price")) +
  labs(title = "PBPB High Price & Low Price in 2021",
       colour = "", x = "Date", y = "Price")
```

4. Visualise volume using a bar plot, over the entire period.

```
# add a new column called year to pbpb_data to represent the year of each observation
# group the pbpb_data by year and calculate the annual trading volume (cshtrd) for each year
annual_volume <- pbpb_data %>%
  mutate(year = year(datadate)) %>%
  group_by(year) %>%
  summarise(volume = sum(cshtrd))

# demonstrate the calulating result table
print(annual_volume)

# plot the result using column chart
# we add scale_y_continuous to make y axis labels more readable
# by scaling down the numbers and adding "M" suffix to represent million
ggplot(annual_volume, aes(year, volume)) +
```

```
  geom_col(fill = "dark blue") +
  scale_y_continuous(labels = label_number(scale = 1e-6, suffix = "M")) +
  labs(x = "Year", y = " Trading Volume",
       title = "Trading Volume for PBPB from 2020 to 2023")
```

```
# A tibble: 4 × 2
   year   volume
  <dbl>    <int>
1  2020 12339320
2  2021 28658498
3  2022  9150412
4  2023 32188643
```



5. Visualise, using a scatter (point) plot, the relationship between simple daily returns and volume, in the year 2021.

```
# add a new column called model to pbpb_data_2021 to store the predicted daily return values
lm_return_volume  <- lm(formula = daily_return ~ cshtrd, data = pbpb_data_2021)
pbpb_data_2021$model <- predict(lm_return_volume)

# Plot a scatter plot with regression line
ggplot(pbpb_data_2021, aes(cshtrd, daily_return)) +
  geom_point() +
  geom_line(aes(y = model, colour = "Regression Line")) +
  labs(title = "Daily Return vs Volume in 2021",
       colour = "", x = "Volume", y = "Daily Return")
```