

Group Assignment 1

Simon Yu

2025-10-05

Table of contents

Question 1	3
Question 2	3
Question 3	3
Question 4	4
Question 5	4
Question 6	5
Question 7	5
Question 8	6
Question 9	6
Question 10	6

List of Figures

List of Tables

```
library(dplyr)
```

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

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

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

Question 1

```
#1. Add a new column named daily_return
#    to store daily return value

#2. 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))
```

Question 2

```
#1. Add a new column named momentum_10
#    to store 10-day momentum indicator value

#2. Drop the rows whose momentum_10 values are NA
sbux_data <- mutate(sbux_data, momentum_10 = prccd - lag(prccd, 10))
sbux_data <- filter(sbux_data, !is.na(momentum_10))
```

Question 3

```
# Add a new column named daily_range
sbux_data <- mutate(sbux_data, daily_range = prchd - prcld)
```

Question 4

```
# Add a new column named MFV
sbux_data <- mutate(sbux_data,
  mfv = ((prccd - prcld) - (prchd - prccd)) * cshtd / (prchd - prcld))

# show the table containing only the new 4 metrics columns
sbux_data_metrics <- filter(sbux_data[ ,
  c("datadate", "daily_return", "momentum_10", "daily_range", "mfv")])

head(sbux_data_metrics, 10)
```

A data.frame: 10 × 5

	datadate <date>	daily_return <dbl>	momen- tum_10 <dbl>	daily_range <dbl>	mfv <dbl>
1	2020-09-17	- 0.0184430867	-1.60	1.965	3099384.35
2	2020-09-18	- 0.0207492795	-1.53	2.640	-7430383.71
3	2020-09-21	- 0.0124779282	-2.38	2.200	6782798.69
4	2020-09-22	0.0007152223	-1.46	1.215	1229446.42
5	2020-09-23	- 0.0114353782	-2.87	2.080	-5361087.23
6	2020-09-24	0.0006024822	-1.84	2.220	43793.63
7	2020-09-25	0.0151734104	-0.97	2.470	4090548.53
8	2020-09-28	0.0209964413	-0.56	1.250	4346617.91
9	2020-09-29	- 0.0147554316	-2.91	1.625	-3184804.85
10	2020-09-30	0.0132075472	-2.46	1.690	-4730361.07

Question 5

```
library(lubridate)
```

```
# Add a new column named month
sbux_data <- mutate(sbux_data, month = month(datadate))
```

Question 6

```
# Add a new column named year
sbux_data <- mutate(sbux_data, year = year(datadate))
```

```
# show the table containing only the new 2 date columns
sbux_data_dates <- filter(sbux_data[, c("datadate", "month", "year")])
head(sbux_data_dates, 10)
```

A data.frame: 10 × 3

	datadate <date>	month <dbl>	year <dbl>
1	2020-09-17	9	2020
2	2020-09-18	9	2020
3	2020-09-21	9	2020
4	2020-09-22	9	2020
5	2020-09-23	9	2020
6	2020-09-24	9	2020
7	2020-09-25	9	2020
8	2020-09-28	9	2020
9	2020-09-29	9	2020
10	2020-09-30	9	2020

Question 7

```
# Calculate the total trade volume (cshtd) for June 2023
trade_volume_2023_06 <- filter(sbux_data, year == 2023 & month == 6)

sum(trade_volume_2023_06$cshtd)
```

151045270

Question 8

```
# Calculate the mean of daily return over the period
mean_daily_return <- mean(sbox_data$daily_return)
print(mean_daily_return)
```

```
[1] 0.0002580336
```

Question 9

```
# Find the date for maximum high price over the period
max_high_price <- max(sbox_data$prchd)

date_max_high_price <- filter(sbox_data, prchd == max_high_price)

print(date_max_high_price$datadate)
```

```
[1] "2021-07-23"
```

Question 10

```
# Find the date for largest daily return over the period
max_daily_return <- max(sbox_data$daily_return)
date_max_daily_return <- filter(sbox_data, daily_return == max_daily_return)
print(date_max_daily_return$datadate)
```

```
[1] "2022-05-04"
```