

Group Assignment 1

Simon Yu

2025-10-01

Table of contents

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

List of Figures

List of Tables

```
library(dplyr)
```

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

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

# convert datadate to Date type
wen_data$datadate <- as.Date(wen_data$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
wen_data <- mutate(wen_data, daily_return = (prccd - lag(prccd)) / lag(prccd))
wen_data <- filter(wen_data, !is.na(daily_return))
```

Question 2

```
#1. Add a new column named overnight_return
#   to store overnight return value

#2. Drop the rows whose overnight_return values are NA
wen_data <- mutate(wen_data, overnight_return = (prcod - lag(prccd)) / lag(prccd))
wen_data <- filter(wen_data, !is.na(overnight_return))
```

Question 3

```
#1. Add a new column named volume_change
#   to store daily volume change value

#2. Drop the rows whose volume_change values are NA

wen_data <- mutate(wen_data, volume_change = (cshtd - lag(cshtd)))
wen_data <- filter(wen_data, !is.na(volume_change))
```

Question 4

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

# show the table containing only the new 4 metrics columns
wen_data_metrics <- filter(wen_data[ ,
  c("datadate", "daily_return", "overnight_return", "volume_change", "mfv")])

head(wen_data_metrics)
```

A data.frame: 6 × 5

	datadate <date>	daily_return <dbl>	overnight_re- turn <dbl>	vol- ume_change <int>	mfv <dbl>
1	2020-09-04	-0.01685649	0.011389522	-929061	-2148667
2	2020-09-08	0.01251158	-0.006950880	87249	1761526
3	2020-09-09	0.02951945	0.006864989	-591443	2005638
4	2020-09-10	-0.02867304	0.002444988	-59870	-2741393
5	2020-09-11	-0.01739130	0.007322654	-462401	-664416
6	2020-09-14	0.02608291	0.007452259	1315156	2556174

Question 5

```
library(lubridate)
```

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

Question 6

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

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

A data.frame: 10 × 3

	datadate <date>	month <dbl>	year <dbl>
1	2020-09-04	9	2020
2	2020-09-08	9	2020
3	2020-09-09	9	2020
4	2020-09-10	9	2020
5	2020-09-11	9	2020
6	2020-09-14	9	2020
7	2020-09-15	9	2020
8	2020-09-16	9	2020
9	2020-09-17	9	2020
10	2020-09-18	9	2020

Question 7

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

sum(trade_volume_2023_06$cshtd)
```

54557454

Question 8

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

[1] 5.375167e-05

Question 9

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

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

print(date_max_high_price$datadate)
```

```
[1] "2021-06-08"
```

Question 10

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

```
[1] "2021-06-08"
```