Assignment 2 - Stock Data Analysis

Barry, Bella, Emma, Maxim, Robin

2025-10-18

Table of contents

List of Figures

List of Tables

Before messing around with the stock data, the environment should install and load the dplyr and lubridate packages as well as others to perform easier data analysis. Additionally, we disable any warning messages for cleaner output. We also remove any rows with NA values in the prcod column.

```
library(dplyr)
library(readr)
library(lubridate)
library(ggplot2)
options(warn=-1)
```

```
data = fread("compressed_data.csv.gz") %>%
    filter(!is.na(prcod)) %>%
        mutate(datadate = as.Date(datadate, "%d/%m/%Y"))
head(data |> select(-conm, -gvkey))
```

A data.table: 6 × 9

Table 1: Compressed dataset

tic <chr>	datadate <date>	exchg <int>	sic <int>	cshtrd <dbl>	prccd <dbl>	prchd <dbl>	prcld <dbl>	prcod <dbl>
PNW	2023-01-03	11	4911	1442534	74.63	76.4125	73.380	76.25
PNW	2023-01-04	11	4911	954218	75.39	76.0950	74.630	75.10
PNW	2023-01-05	11	4911	994775	73.65	75.0950	73.305	74.88
PNW	2023-01-06	11	4911	729808	75.46	76.0200	74.480	74.49
PNW	2023-01-09	11	4911	656127	75.55	76.4800	75.240	75.24
PNW	2023-01-10	11	4911	763254	75.65	75.6950	74.880	75.31

Part 1 Questions & Answers

1. How many unique tickers are in your data?

```
cat("1. There are", length(unique(data$tic)), "unique tickers.")
```

```
1. There are 502 unique tickers.
```

2. How many unique companies are in your data?

```
cat("\n2. There are", length(unique(data$conm)), "unique company names.")
```

```
2. There are 499 unique company names.
```

3. Display the top 5 companies by largest mean trading volume, in a table.

```
data_3 = data %>%
    group_by(tic) %>%
    summarise(mean_trading_v = mean(cshtrd, na.rm = TRUE)) %>%
    ungroup() %>%
        arrange(desc(mean_trading_v))
data_3[1:5,]
```

A tibble: 5 × 2

Table 2: Top 5 companies by largest mean trading volume

tic <chr>	mean_trading_v <dbl>
TSLA	115314383
NVDA	113131835
PLTR	60056251
AAPL	57736403
AMD	57143415

4. Display the total trading volume of the top 3 exchanges (table).

```
data_4 = data %>%
    group_by(exchg) %>%
    summarise(total_trading_v = sum(cshtrd, na.rm = TRUE)) %>%
    ungroup() %>%
        arrange(desc(total_trading_v))
data_4[1:3,]
```

A tibble: 3 × 2

Table 3: Total trading volume of the top 3 exchanges

exchg <int>	total_trading_v <dbl>
11	681415756062
14	570830885382
21	385399362

5. Visualise the total trading volume of the top 3 exchanges (bar plot).

```
ggplot(data_4, aes(x = as.character(exchg), y = total_trading_v/1000000)) +
  geom_bar(stat = "identity", color = "darkblue", fill = "darkblue") +
  geom_text(aes(label = round(total_trading_v/1000000)),
            vjust = -0.3,                # position above the bar
            size = 5) +                  # text size
  labs(title = "Total Trading Volume of the Top 3 Exchanges",
  x= "exchange", y="Total Trading Volume in illions") +
  theme(plot.title = element_text(hjust = 0.5))
```

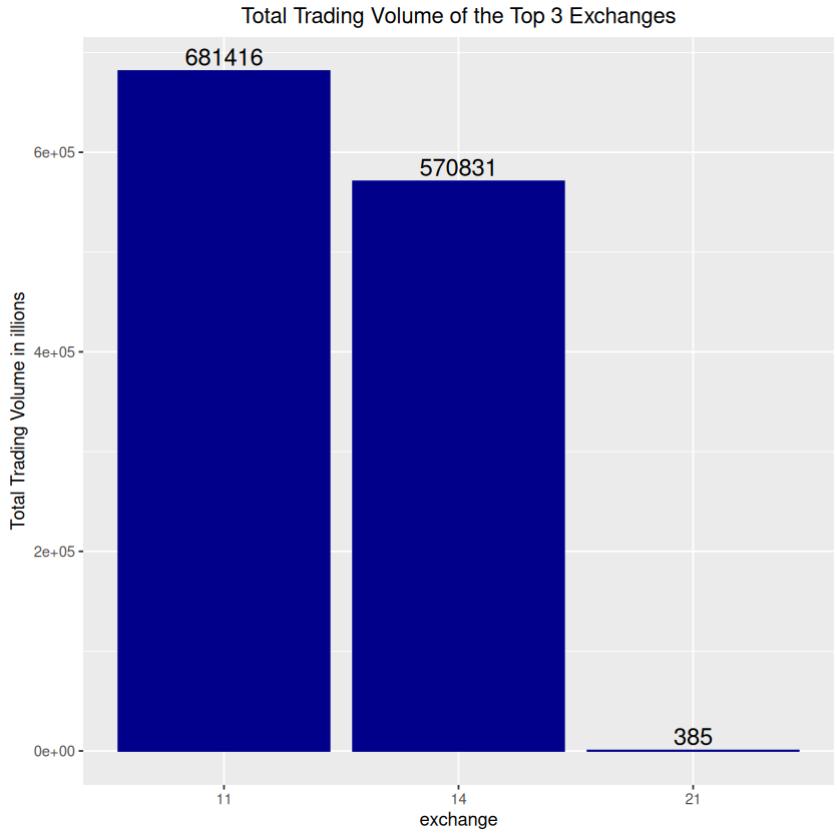

Figure 1: Total Trading Volume of the Top 3 Exchanges

6. How many companies have more than one ticker?

```
data_6 = data %>%
    group_by(conm) %>%
    summarise(no_of_tickers = n_distinct(tic)) %>%
    ungroup() %>%
    filter(!no_of_tickers == 1)
data_6[1:4,]
```

```r
nr_companies = nrow(data_6)
cat("6. There are", nr_companies, "companies with more than one ticker.")
```

A tibble: 4 × 2

Table 4: Companies with more than one ticker

conm <chr>	no_of_tickers <int>
ALPHABET INC	2
FOX CORP	2
NEWS CORP	2
NA	NA

```
6. There are 3 companies with more than one ticker.
```

7. Which ticker has the largest positive mean return (simple daily return)?

```r
# 7. Which ticker has the largest positive mean return (simple daily return)?
data = data %>%
    group_by(tic) %>%
    mutate(return = prccd/lag(prccd)-1) %>%
    ungroup()

data_7 = data %>%
    group_by(tic) %>%
    summarise(mean_return = mean(return, na.rm = TRUE)) %>%
    ungroup() %>%
        arrange(desc(mean_return))

highest_mean_return = max(data_7$mean_return)

highest_mean_return_ticker = data_7$tic[
    which.max(data_7$mean_return)
]
cat("7. The", highest_mean_return_ticker,
"ticker had the highest mean daily return.")
cat("\n-> The return was", round(highest_mean_return, 4)*100,"%.")
```

```
7. The PLTR ticker had the highest mean daily return.
-> The return was 0.58 %.
```

8. Which company has the largest positive mean return (simple daily return)?

```
highest_mean_return_company = data$conm[
    which(data$tic == highest_mean_return_ticker)[1]
]
cat("8. The", highest_mean_return_company,
"company had the highest mean daily return.")
```

```
8. The PALANTIR TECHNOLOG INC company had the highest mean daily return.
```

9. Which industry is represented by the most companies?

```
data_9 = data %>%
    group_by(sic) %>%
    summarise(no_companies = n_distinct(conm)) %>%
    ungroup() %>%
    arrange(desc(no_companies))
most_represented_industry = data_9$sic[
    which.max(data_9$no_companies)
]
no_companies_in_most_represented_industry = max(data_9$no_companies)

cat("9. The", most_represented_industry,
"SIC industry has the most companies.")
cat("\n-> There are", no_companies_in_most_represented_industry,
"companies in that industry.")
```

```
9. The 6798 SIC industry has the most companies.
-> There are 28 companies in that industry.
```

Part 2 Extended Analysis

After preparing the data we carry out the following analysis.

1. Calculate simple weekly returns for each ticker in the full dataset

```
data_weekly = data %>%
    mutate(friday = floor_date(datadate, "week")+5) %>%
    filter(friday == datadate) %>%
        group_by(tic) %>%
        mutate(simple_w_r = prccd / lag(prccd) - 1) %>%
        select(tic, conm, friday, datadate, prccd, simple_w_r) %>%
```

```
        ungroup()
head(data_weekly |> select(-conm, -friday))
```

A tibble: 6 × 4

Table 5: Simple weekly returns for each ticker

tic <chr>	datadate <date>	prccd <dbl>	simple_w_r <dbl>
PNW	2023-01-06	75.46	NA
PNW	2023-01-13	75.37	-0.0011926849
PNW	2023-01-20	75.36	-0.0001326788
PNW	2023-01-27	74.07	-0.0171178344
PNW	2023-02-03	75.88	0.0244363440
PNW	2023-02-10	74.09	-0.0235898788

2. Categorise your data into decile groups

(We do not remove zero returns from the data).

```
c_breaks = seq(0, 1, by = 0.1)
print(c_breaks)

c_labels <- paste0((1:(length(c_breaks) - 1)) * 10, "%")
print(c_labels)

#| tbl-cap: "Data categorized by decile groups"
data_weekly_deciles <- data_weekly %>%
    mutate(
        deciles = cut(
            simple_w_r,
            breaks = quantile(
            simple_w_r,
            probs = c_breaks,
            type = 9,
            na.rm = TRUE
            ),
        labels = c_labels,
        include.lowest = TRUE
        )
    ) %>%
  arrange(tic, datadate)
head(data_weekly_deciles |> select(-conm))
```

```
 [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
 [1] "10%"  "20%"  "30%"  "40%"  "50%"  "60%"  "70%"  "80%"  "90%"  "100%"
```

A tibble: 6 × 6

tic <chr>	friday <date>	datadate <date>	prccd <dbl>	simple_w_r <dbl>	deciles <fct>
A	2023-01-06	2023-01-06	147.67	NA	NA
A	2023-01-13	2023-01-13	156.92	0.062639670	100%
A	2023-01-20	2023-01-20	155.92	-0.006372674	40%
A	2023-01-27	2023-01-27	155.69	-0.001475115	50%
A	2023-02-03	2023-02-03	154.55	-0.007322243	40%
A	2023-02-10	2023-02-10	152.55	-0.012940796	40%

3. Display a table showing the top ticker in each decile group

```
df_top_ticker = data_weekly_deciles %>%
    group_by(deciles) %>%
    filter(simple_w_r == max(simple_w_r, na.rm = TRUE)) %>%
    ungroup() %>%
        arrange(desc(deciles)) %>%
        select(tic, conm, datadate, friday, prccd, simple_w_r, deciles)
df_top_ticker_show = df_top_ticker %>%
    select(-conm)
df_top_ticker_show[1:11,]
```

A tibble: 11 × 6

Table 7: Top tickers by decile group

tic <chr>	datadate <date>	friday <date>	prccd <dbl>	simple_w_r <dbl>	deciles <fct>
SMCI	2024-11-22	2024-11-22	33.15	0.784176534	100%
WFC	2024-07-19	2024-07-19	59.23	0.047576937	90%
LW	2023-03-24	2023-03-24	100.19	0.029596136	80%
EXE	2023-10-13	2023-10-13	88.92	0.018790101	70%
DHR	2024-08-02	2024-08-02	276.75	0.010368369	60%
AME	2023-04-28	2023-04-28	137.93	0.002543974	50%
GWW	2023-09-01	2023-09-01	710.78	-0.005289969	40%
BDX	2023-10-13	2023-10-13	258.70	-0.014175749	30%
UDR	2023-02-24	2023-02-24	43.63	-0.025027933	20%
FAST	2023-10-20	2023-10-20	57.61	-0.041749834	10%
NA	NA	NA	NA	NA	NA

4. Select the top ticker from the 60% decile group

We use this ticker for the rest of the assignment, including in Part 3.

```r
top_ticker_60d = as.character(df_top_ticker %>%
    filter(deciles == "60%") %>%
    select(tic)
    )
cat("4. ", top_ticker_60d)
```

4. DHR

5. Plot the autocorrelation function for this ticker's entire set of weekly returns

```r
# Filter and remove NA values
csco_data = data_weekly %>%
    filter(tic == top_ticker_60d) %>%
        na.omit()

acf(csco_data$simple_w_r, main = "Autocorrelation Function")
```

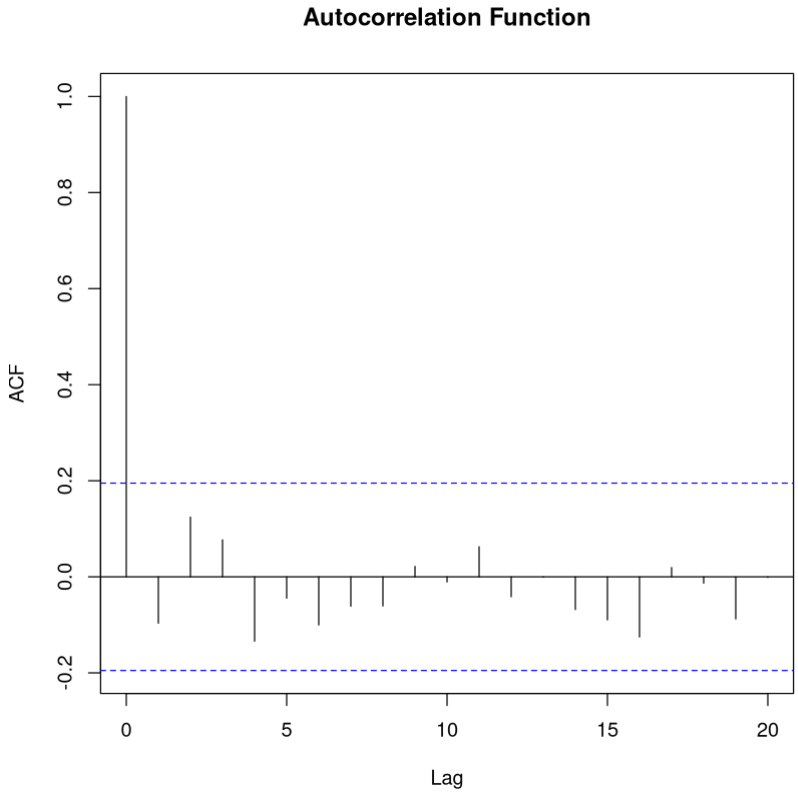

Figure 2: Autocorrelation Function

Part 3 Regression - Fama-French 3 Factor Model

1. Load and clean the weekly Fama-French 3 factor data

```
ff <- read.csv("fama_french_weekly.csv", skip = 4) %>%
  rename(x = X,
         mktrf = Mkt.RF,
         smb   = SMB,
         hml   = HML,
         rf    = RF) %>%
  mutate(
    date  = ymd(as.character(x)),
    mktrf = mktrf / 100,
    smb   = smb / 100,
    hml   = hml / 100,
    rf    = rf / 100
  ) %>%
  transmute(date, mktrf, smb, hml, rf) %>%
  filter(!is.na(date)) %>%
  arrange(date)

head(ff)
```

A data.frame: 6 × 5

Table 8: Fama-French 3 Factor Model

	date <date>	mktrf <dbl>	smb <dbl>	hml <dbl>	rf <dbl>
1	1926-07-02	0.0158	-0.0062	-0.0086	6e-04
2	1926-07-10	0.0037	-0.0090	0.0031	6e-04
3	1926-07-17	0.0098	0.0059	-0.0144	6e-04
4	1926-07-24	-0.0203	0.0002	-0.0017	6e-04
5	1926-07-31	0.0306	-0.0189	-0.0085	6e-04
6	1926-08-07	0.0204	0.0016	0.0055	6e-04

2. Fit the Fama-French 3 factor model to the weekly returns of the stock in Part 2

```
# 1) Get the chosen stock's weekly returns
ticker_data <- data_weekly %>%
  filter(tic == top_ticker_60d) %>%
  select(datadate, simple_w_r) %>%
  filter(!is.na(simple_w_r))

# 2) Join with Fama-French factors (align on week end)
ff_weekly <- ff %>% rename(datadate = date)
```

```r
merged <- ticker_data %>%
  inner_join(ff_weekly, by = "datadate") %>%
  mutate(excess_return = simple_w_r - rf)

# 3) Fit FF3: excess_return ~ Mkt.RF + SMB + HML
ff3_model <- lm(excess_return ~ mktrf + smb + hml, data = merged)

# 4) Show results
summary(ff3_model)
```

```
Call:
lm(formula = excess_return ~ mktrf + smb + hml, data = merged)

Residuals:
      Min        1Q    Median        3Q       Max
-0.110997 -0.016172  0.000858  0.016220  0.105408

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.002794   0.003211  -0.870 0.386366
mktrf        0.676790   0.185236   3.654 0.000419 ***
smb          0.630774   0.209496   3.011 0.003321 **
hml          0.175506   0.187334   0.937 0.351154
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03126 on 97 degrees of freedom
Multiple R-squared:  0.2901,    Adjusted R-squared:  0.2682
F-statistic: 13.22 on 3 and 97 DF,  p-value: 2.645e-07
```

Analysis Summary

BEWARE: THIS IS NOT FINAL, ADJUST FOR PROPER TICKER AND DATA

CNC showed a moderate positive weekly return of about 1.85%, placing it in the 60% performance decile. The autocorrelation results indicated no meaningful serial correlation, meaning CNC's returns are largely random and past movements do not predict future ones.

The Fama–French 3-factor regression showed a weak positive link with the overall market, a negative relationship with company size, and a mild positive relationship with value characteristics. However, none were statistically significant, and the low R^2 (0.05) suggests that most of CNC's return variation cannot be explained by common market factors.

This implies CNC's performance is mainly driven by firm-specific or idiosyncratic factors rather than broad market effects.