# Applications of R for Finance
# Course Project

Anthony Li
CID: 02308277
Business School, Imperial College London

16-11-2022

# Contents



# List of Figures

# List of Tables

In this assignment, we will load data from the file `compustat_sec_2019_2022.csv`, which contains stock data for a large number of different companies.

I will be calculating and presenting the daily returns and the max monthly returns. The September 2019 returns will be categorised into decile groups with the top decile group (top 10%) in September 2019 presented in a table.

The stock "NBCC (INDIA) LTD 01W" will be selected from the original data and a simple trading strategy will be applied to this stock. I will apply a Moving Average Crossover Strategy with the functions uploaded from a separate .R script. I will backtest this strategy

## Load R packages

```
library(knitr)
library(dplyr)
library(lubridate)
library(xts)
library(data.table)
library(dtplyr)
library(ggplot2)
library(PerformanceAnalytics)

# Importing another .R file which contains functions I will be using
source("rfun.R")
```

## Load and prepare data

We load and display the dataset as follows:

Load the entire dataset from `compustat_sec_2019_2022.csv`

```
# Loading data using fread which is much faster compared to read.csv hence should
# be used for large datasets. However, the result would be a table datatype rather
# than a data frame.
big_data <- fread("compustat_sec_2019_2022.csv", drop = c("sic", "exchg"))

# Displaying first 6 rows
head(big_data)
```

```
##    gvkey iid datadate                conm cshtrd prccd
## 1:  1166 01W 20190101 ASM INTERNATIONAL NV     NA 36.20
## 2:  1166 01W 20190102 ASM INTERNATIONAL NV 136748 36.16
## 3:  1166 01W 20190103 ASM INTERNATIONAL NV 255062 33.96
## 4:  1166 01W 20190104 ASM INTERNATIONAL NV 180057 35.07
## 5:  1166 01W 20190107 ASM INTERNATIONAL NV 185777 36.67
## 6:  1166 01W 20190108 ASM INTERNATIONAL NV 218171 35.64
```

## Calculate daily returns

We use the lagged proceeds (prccd) in order to calculate daily returns of each stock in the dataset. I have defined the function to calculate returns in the rfun.R file.

```
# I am applying pipe operators to modify the data. I have used the dtplyr so that
# I can apply the normal pipe operators to the data table datatype.
data <- big_data %>%
  group_by(conm, iid) %>%
  mutate(returns = simple_daily_return(prccd) ) %>%
  ungroup() %>%
  as.data.table()

head(data)
```

```
##    gvkey iid datadate                conm cshtrd prccd      returns
## 1:  1166 01W 20190101 ASM INTERNATIONAL NV     NA 36.20           NA
## 2:  1166 01W 20190102 ASM INTERNATIONAL NV 136748 36.16 -0.001104972
## 3:  1166 01W 20190103 ASM INTERNATIONAL NV 255062 33.96 -0.060840708
## 4:  1166 01W 20190104 ASM INTERNATIONAL NV 180057 35.07  0.032685512
## 5:  1166 01W 20190107 ASM INTERNATIONAL NV 185777 36.67  0.045623040
## 6:  1166 01W 20190108 ASM INTERNATIONAL NV 218171 35.64 -0.028088356
```

## Calculate maximum daily returns for each month

We extract the month from each date using the floor_date() function and store it in a new column called return_month. We then look for the maximum daily returns for each month and store them in a new column called max_return. We only keep the rows where the maximum daily return is equal to the daily return.

**Store the maximum daily returns in a new column called `max_return`**

```
# Converting the datatype of datadate from Integer to Date
data$datadate <- as.Date(as.character(data$datadate), format = "%Y%m%d")

# Extract the month from each date using floor_date(), and store in a new column
data$return_month <- floor_date(data$datadate, unit = "month",
                                week_start = getOption("lubridate.week.start",7))

# Create a new column named max_return with the maximum daily returns each month
max_daily_return <- data  %>%
  group_by(return_month, gvkey, iid, conm) %>%
  filter(is.finite(returns)) %>%
  mutate(max_return = max(returns, na.rm = TRUE) ) %>%
  ungroup() %>%
  as.data.table()

# Keep only the rows where the maximum daily return matches the daily return
max_daily_return.uni <-
  max_daily_return[max_daily_return$returns == max_daily_return$max_return ,]

max_daily_return.uni1 <- max_daily_return.uni[!duplicated(max_daily_return.uni
[,c("return_month", "max_return", "iid", "conm", "gvkey")]), ]
```

```
# Displaying max monthly returns for first 6 months
knitr::kable(
  head(max_daily_return.uni1), booktabs = TRUE,
  caption = 'Maximum monthly returns'
)
```

Table 1: Maximum monthly returns

| gvkey | iid | datadate | conm | cshtrd | prccd | returns | return_month | max_return |
|---|---|---|---|---|---|---|---|---|
| 1166 | 01W | 2019-01-24 | ASM INTERNATIONAL NV | 255440 | 40.00 | 0.0598834 | 2019-01-01 | 0.0598834 |
| 1166 | 01W | 2019-02-22 | ASM INTERNATIONAL NV | 752032 | 49.26 | 0.1195455 | 2019-02-01 | 0.1195455 |
| 1166 | 01W | 2019-03-15 | ASM INTERNATIONAL NV | 272910 | 48.48 | 0.0325879 | 2019-03-01 | 0.0325879 |
| 1166 | 01W | 2019-04-25 | ASM INTERNATIONAL NV | 836341 | 60.78 | 0.0967160 | 2019-04-01 | 0.0967160 |
| 1166 | 01W | 2019-05-14 | ASM INTERNATIONAL NV | 353243 | 56.22 | 0.0399556 | 2019-05-01 | 0.0399556 |
| 1166 | 01W | 2019-06-11 | ASM INTERNATIONAL NV | 282003 | 55.56 | 0.0315633 | 2019-06-01 | 0.0315633 |

## Categorise into decile groups

Having obtained the maximum daily returns, we take a subset made up of the September 2019 maximum daily returns and we categorise them into decile groups using the cut() function.

**Categorise into deciles in a new column called `decile_cat`**

```
# Extract a subset of the maximum daily returns (September 2019)
data_Sep2019 <- max_daily_return.uni1  %>%
  filter(return_month=="2019-09-01") %>%
  as.data.table()


# Categorise into decile groups in a new column named decile_cat
data_Sep2019$decile_cat <- as.character(cut(data_Sep2019$returns,
                           quantile(data_Sep2019$returns,
                           probs = seq(0,1, length = 11), na.rm = T, type = 5),
                           include.lowest = TRUE, labels = c("10%", "20%", "30%"
                           ,"40%", "50%", "60%", "70%", "80%", "90%", "100%" )))

top_decile_group <- data_Sep2019[data_Sep2019$decile_cat == "100%",]
```

**Presenting the top 10% decile group (i.e. the 100% decile group)**

```
# Display in the final render
knitr::kable(
  head(top_decile_group), booktabs = TRUE,
  caption = 'Top 10% decile group in Sep 2019.'
)
```

Table 2: Top 10% decile group in Sep 2019.

| gvkey | iid | datadate | conm | cshtrd | prccd | returns | return_month | max_return | decile_cat |
|---|---|---|---|---|---|---|---|---|---|
| 1661 | 01W | 2019-09-10 | NABORS INDUSTRIES LTD | NA | 2.028 | 0.1173554 | 2019-09-01 | 0.1173554 | 100% |
| 2162 | 01W | 2019-09-03 | BENGUET CORP | 60000 | 1.300 | 0.1711712 | 2019-09-01 | 0.1711712 | 100% |
| 4367 | 02W | 2019-09-20 | WEATHERFORD INTL PLC | 25000 | 0.030 | 0.9354839 | 2019-09-01 | 0.9354839 | 100% |
| 5302 | 01W | 2019-09-10 | LABYRINTH RESOURCES LIMITED | 57135 | 0.135 | 0.1250000 | 2019-09-01 | 0.1250000 | 100% |
| 7152 | 01W | 2019-09-20 | MCDERMOTT INTL INC | 9483 | 1.840 | 0.2365591 | 2019-09-01 | 0.2365591 | 100% |
| 11925 | 01W | 2019-09-10 | NOBLE CORPORATION | NA | 1.810 | 0.1312500 | 2019-09-01 | 0.1312500 | 100% |

## Extracting data for "NBCC (INDIA) LTD"

```
NBCC <- big_data %>%
  filter(conm == "NBCC (INDIA) LTD", iid == "01W") %>%
  as.data.frame()

NBCC$datadate <- as.Date(as.character(NBCC$datadate), format = "%Y%m%d")
```

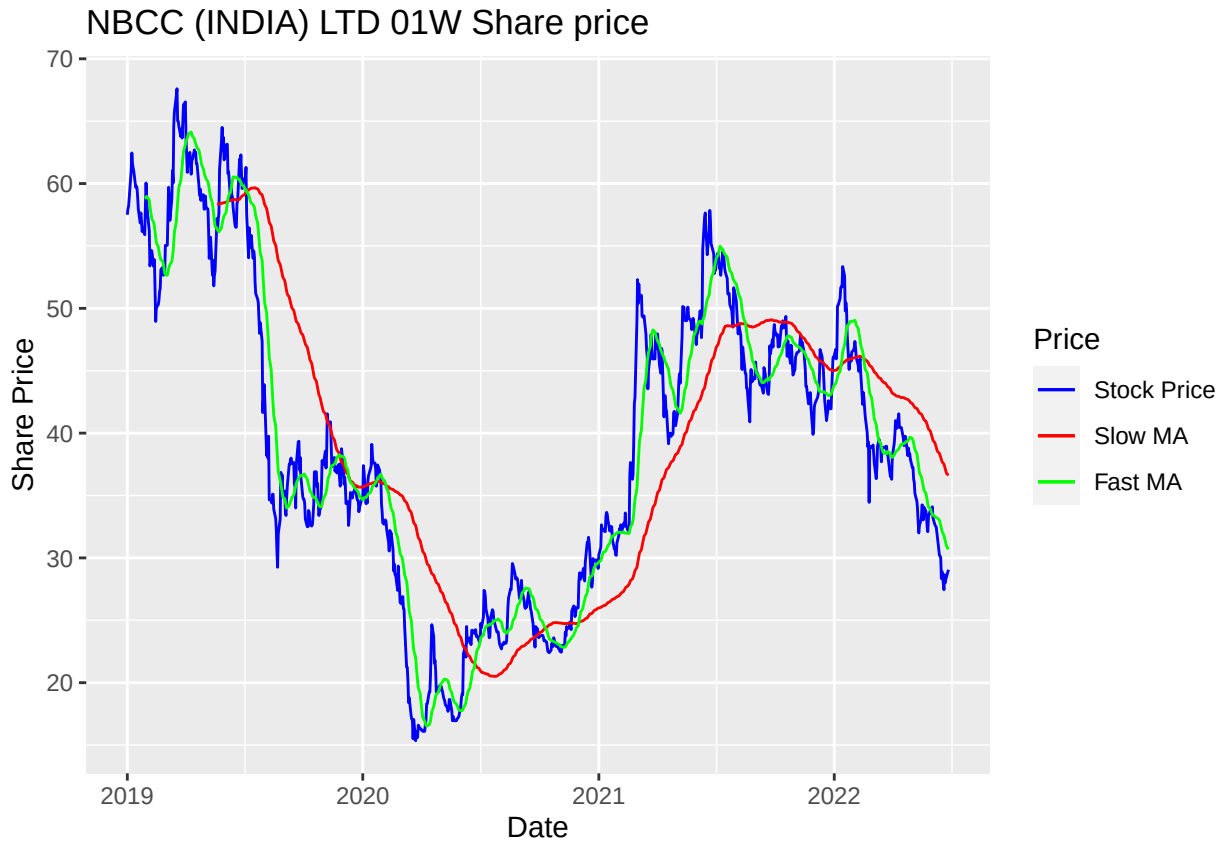
## Implementing Moving Average Crossover Strategy

```
NBCC.ts <- xts(NBCC$prccd, order.by = NBCC$datadate)
colnames(NBCC.ts) <- "NBCC"


# Using movingAvg function in the rfun.R script to calculate MA
MA_100 <- movingAvg(NBCC.ts, 100)
MA_20 <- movingAvg(NBCC.ts, 20)

# Obtaining the a buy or sell signal depending on whether the short term MA has
# crossed the long term MA from above or below
signal <- sign(MA_20-MA_100)

# No moving average for first n observations where n is the long term MA length
signal[is.na(signal)] <- 0

# Plotting graph of the Moving Averages and the stock price
ggplot()+
  geom_line(NBCC, mapping = aes(x=datadate ,y=prccd, color="Stock Price"))+
  geom_line(NBCC, mapping = aes(x=datadate ,y=MA_100, color="Slow MA"))+
  geom_line(NBCC, mapping = aes(x=datadate ,y=MA_20, color="Fast MA"))+
  geom_line()+
  ggtitle("NBCC (INDIA) LTD 01W Share price") +
```

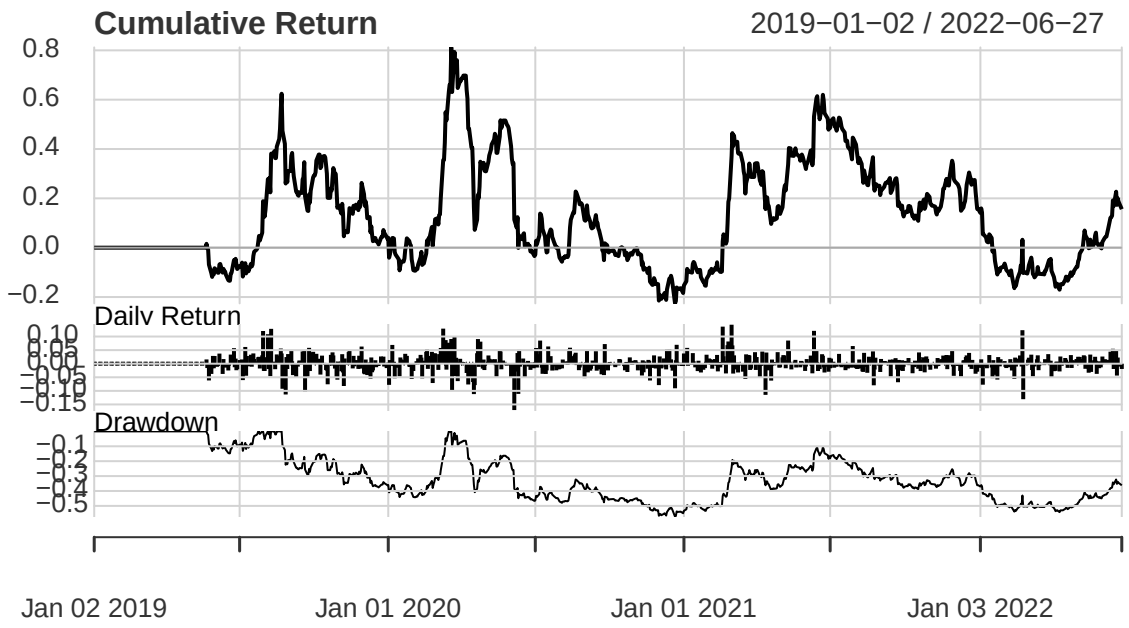
```
  ylab("Share Price") +
  xlab("Date") +
  scale_color_manual(name = c("Price", "Slow MA", "Fast MA"),
      values = c("Stock Price" = "blue", "Slow MA" = "red", "Fast MA" = "green"))
```

```
## Warning: Removed 99 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 19 row(s) containing missing values (geom_path).
```




```
results <- simpleBacktest(NBCC.ts, signal)

charts.PerformanceSummary(results)
```




## Analysis

The cumulative returns seem to be very volatile and occasionally reaches below 0. However, the period ends with a positive cumulative return showing that the strategy is profitable.