Topic 2: Predicting the number of Twitter followers

ACCT653 Forecasting and Forensic Analytics

Instructor's name: Wang Jiwei

AUTHORS
CHERN Cheng (01354502)
SUN Yifan (01461236)
TEAN Wan Ching (01462626)
YUE Zixuan (01475299)
ZHANG Erhe (01475073)

PUBLISHED
March 17, 2024

ABSTRACT
Predicting the number of social media followers for US corporations

1. Introduction

(This file requires around 10 minutes to run through.) Social media prediction (SMP) encompasses a paramount framework when conducting the prediction process, including considerations such as the source of data, algorithm used and evaluation of prediction. This topic has emerged as a powerful tool, attracting the attention of researchers and practitioners alike. Among the abundant studies, multiple predictors and features have been utilized to improve prediction. However, most studies still focus on the social behaviours and activities on social media itself. Based on this, we aim to broaden the possibilities by embedding more potential predictors to enhance results as well as better understand the function of machine learning tools.

During this project, using R programming as our coding language, we analyze and attempt to predict the daily number of twitter followers for US-listed firms during 2017. We first briefly review the literature, followed by an overview of our data sources in the data preparation section, then the explanatory description of data, sample and variables descriptive statistics. After that, we run several types of models and machine learning algorithms and evaluate the prediction results. Finally, we summarize the performance of all our models and share the key take aways from this project.

2. Literature Review

The exposure of social media account has been a subject of extensive inquiry, with operating multiple and advanced social platforms becoming the mainstream of advertising and marketing. While substantial evidence exists trying to unfold the key factors behind the influence of the social media accounts, empirical results vary and do not come to the consensus about how to better predict and utilize the influence, among which number of followers is one of the main indicators. To elucidate the achievements of previous research and the contribution of this report, we review primarily two strands of literature closely aligned with our study. The first strand of studies is focused on how financial information or statistics can be related to the popularity of twitter or influence of their social media accounts. While most studies have shed some light on how the sentiment of tweets or post contents can affect the financial and stock market, literature studying the effect of financial information on the popularity of social media accounts remain inadequate. Sayed (n.d.) utilizes accounting ratios to analyze the twitter sentiment and volume. The results shows that certain accounting ratios are not associated with the formation of twitter users, however, they do have positive impact on the twitter volume. Intuitively, twitter volume could have a strong effect on the number of twitter followers. This study provides evidence supporting our choice of financial indicators as features in our prediction model, especially considering corporation accounts as predicting object.

Naturally, number of followers and the popularity of social media accounts could be directly influenced by the content of posts, the style of the account and some exogenous event, which could bring more visibility of certain accounts. Therefore, we mainly refer to studies in this area when selecting our non-financial features. Hutto et al. (2013) finds that variables for message content, social behaviour, and network structure should be equally considered when predicting followers. Similarly, the popularity on twitter can be mainly driven by tweets, profile page. (Mueller & Stumme, 2017) Considering their variables selection and data availability, we create profile related variables to show the characteristics of corporation account features. Besides, Tsileponis et al. (2020) examine the influence of voluntary corporate press releases on financial media coverage, which gives us insight into the function of corporations' announcements. Therefore, we also consider announcements as important predictors in our models.

In summary, predicting social media followers is a multifaceted topic that has drawn great attention in the academic field. While various studies have contributed valuable insights, the field remains dynamic and is worthy of further exploration, especially considering the function of financial indicators has not been sufficiently discussed. This report contributes to the studies of this field and aims to enhance the understanding of the interplay between corporation's economic activities, social behaviours and social media popularity.

3. Data Extraction and Data Cleaning

3.1 : Data used

These are the dataset used in our group project:

1. Followers_train.csv

   - the training dataset provided on Kaggle (https://www.kaggle.com/competitions/followers/data)

2. Followers_test.csv

   - the test dataset provided on Kaggle (https://www.kaggle.com/competitions/followers/data)

3. Followers_sampleSubmission.csv

   - a sample submission file in the correct format on Kaggle (https://www.kaggle.com/competitions/followers/data)

4. profile_info.csv

- the twitter's users profile dataset is scrape from Twitter (https://twitter.com), the process of scraping data is presented in Part 1.2

5. stock_price.csv

   From S&P Global Market Intelligence > S&P Compustat Global

6. key_developments.csv

   From S&P Capital IQ > Companies > Key Developments

7. Financials.csv

   From Compustat - Capital IQ > North America > Fundamentals Quarterly

8. Financial Ratios.csv

   From Compustat - Capital IQ > North America > Financial Ratios Suite by WRDS

3.2 : Packages used

```
library(readr)
library(R.utils)
library(readxl)
pacman::p_load(rvest)
pacman::p_load(tidyverse)
pacman::p_load(ggplot2)
pacman::p_load(randomForest)
pacman::p_load(glmnet)
pacman::p_load(coefplot)
pacman::p_load(xgboost)
pacman::p_load(ParBayesianOptimization)
pacman::p_load(SmartEDA)
pacman::p_load(quanteda.textstats)
```

3.3 : Scrape Twitter profile data

Section 3.3 aims to scrape the profile information of companies' twitter accounts using RSelenium. The Selenium packages requires specific configuration (kindly refer to this: https://www.youtube.com/watch?v=GnpJujF9dBw) in order for the codes to run. This process takes a long time with some manual work needed. Hence, we recommend to skip these parts and use the scraped `profile_info.csv` (in Section 3.7) directly. You may proceed straight to Section 3.4 onwards to continue with processing and data cleaning codes.

Due to the issue of mismatch and presence of advertisements on twitter, we cannot locate a company's twitter account simply by searching their company name. In this case, we will need to search and match the unique Twitter account name of each company, stored in `company information.xlsx` and resume scraping with the matched Twitter account names.

```
# remove all the variables
rm(list=ls())


# load the required packages
```

```r
library(RSelenium)
library(wdman)
library(netstat)


my_user <- "89423301"
my_pass <- "y150500z113104X!"

# open remote driver with firefox
remote_driver <- rsDriver(browser = "firefox",
                          chromever = NULL,
                          verbose = F,
                          port = free_port())


remDr <- remote_driver$client
remDr$open()
remDr$navigate("https://twitter.com/i/flow/login")

# enter user_id
webElement <- remDr$findElement(using = "xpath", value = "//input[@type = 'text']")
webElement$sendKeysToElement(list(my_user))
webElement$sendKeysToElement(list(key = "enter"))
Sys.sleep(3)

# enter password
webElement <- remDr$findElement(using = "xpath", value = "//input[@type = 'password']")
webElement$sendKeysToElement(list(my_pass))
webElement$sendKeysToElement(list(key = "enter"))

# explanation of matching the company with username ------------------------------------------
### we try to automate the process of matching company name with its username using selenium.
### however, searching the company name can be matched to multiple accounts or zero accounts
### which also requires our manual judgement
### So we manully search and match the company name and the username of their account and stor
# collect the profile information -----------------------------------------------
accounts <- read_excel("company information.xlsx")
accounts <- accounts %>% select(conm, username)
accounts <- filter(accounts, !is.na(username))
accounts$username <- substring(accounts$username, 2)

com_info <- data.frame(matrix(vector(),0,7,
                              dimnames = list(c(),
                                              c("profile_name",
                                                "profile_bio",
                                                "profile_category",
                                                "profile_website",
                                                "profile_joining_date",
                                                "profile_following",
                                                "profile_followers"))),
                       stringsAsFactors=F)

# for each account scrape the basic information
for (j in 84:nrow(accounts)){
```

```
search_item = accounts$username[j]

search_box = remDr$findElement(using = "xpath", value = "//input[@data-testid='SearchBox_Sea
search_box$sendKeysToElement(list(search_item))
search_box$sendKeysToElement(list(key = "enter"))
Sys.sleep(5)

# turn to the people page
remDr$findElement(using = "link text", value = "People")$clickElement()
Sys.sleep(5)

# click the account
# now all click the first account by default
remDr$findElement(using = "xpath", value = paste0("//a[@href='/", search_item, "']"))$clickE

# scrape the account information
l = list()
o = {}

resp <- remDr$getPageSource()
html_info <- read_html(resp[[1]])

# profile name
tryCatch({
  profile_name <- html_info %>% html_node(".r-1vr29t4") %>% html_text()
  o[["profile_name"]] <- profile_name
}, error = function(e) {
  o[["profile_name"]] <- NA
})


# profile bio
tryCatch({
  profile_bio <- html_info %>% html_node("[data-testid='UserDescription']") %>% html_text()
  o[["profile_bio"]] <- profile_bio
}, error = function(e) {
  o[["profile_bio"]] <- NA
})

# information under the profile header
profile_header <- html_info %>% html_node("[data-testid='UserProfileHeader_Items']")

# profile category
tryCatch({
  profile_category <- profile_header %>% html_node("[data-testid='UserProfessionalCategory']
  o[["profile_category"]] <- profile_category
}, error = function(e) {
  o[["profile_category"]] <- NA
})

# website
tryCatch({
  profile_website <- profile_header %>% html_node("a") %>% html_attr("href")
```

```
    o[["profile_website"]] <- profile_website
  }, error = function(e) {
    o[["profile_website"]] <- NA
  })

  # join_date
  tryCatch({
    profile_joining_date <- profile_header %>% html_node("span[data-testid='UserJoinDate']") %
    o[["profile_joining_date"]] <- profile_joining_date
  }, error = function(e) {
    o[["profile_joining_date"]] <- NA
  })

  # number of following
  tryCatch({
    profile_following <- remDr$findElement(using = "xpath", "//span[contains(text(), 'Followin
    following <- profile_following$getElementText()
    o[["profile_following"]] <- following[[1]]
  }, error = function(e) {
    o[["profile_following"]] <- NA
  })

  # number of followers
  tryCatch({
    profile_followers <- remDr$findElement(using = "xpath", "//span[contains(text(), 'Follower
    followers <- profile_followers$getElementText()
    o[["profile_followers"]] <- followers[[1]]
  }, error = function(e) {
    o[["profile_followers"]] <- NA
  })


  # add search name
  o[["user_name"]] <- paste0("@", search_item)

  l <- append(l, o)
  cat(paste0("--------profile collection finish for ", accounts$conm[j], "-------- \n"))
  com_info <- rbind(com_info, l)


  remDr$goBack()
  remDr$goBack()

}

# if twitter run into something wrong, the codes will also run into error
# close the webpage and repeat the login and continue the scraping
# it is better if we can scrape at one time, but due to the restriction or some reason,
# it is acceptable if we need to separate the scraping into many times
remDr$close()

# save the file
write.csv(com_info, "profile_info.csv", row.names = FALSE)
```

```
# terminate the selenium server
system("taskkill /im java.exe /f")
remote_driver$server$stop()
```

3.4 : Processing stock price data

Set up environment and load in data

```
options(scipen=999, digits=4) # avoid scientific display, keep 4 digits in display
rm(list=ls()) # clear Environment
# Set the working directory
dir_data <- setwd ("D:/Desktop/forecast and forensic analysis/group_project/merge/")


library(tidyverse)
library(dplyr)
library(lubridate)
library(tidyr)
library(readxl)

#load the stock price data
stock_price <- read.csv("stock_price.csv")

#load the followers_train data and followers_test data
f_train <- read.csv("Followers_train.csv")
f_test <- read.csv("Followers_test.csv")
```

We retrieve unique identifiers (gvkey) from f_train dataset, filters stock_data based on gvkey, and left joins f_train with stock_data. It then fills missing values with the most recent non-NA value in stock_data. Finally, we using summary( ) to check any missing values in the stock_data dataframe.

```
#get the unique gvkey as save it as an vector
unique_gvkey <- unique(f_train$gvkey)

#filter the stock_price data to only keep the relevent company
stock_price <- stock_price[stock_price$gvkey %in% unique_gvkey,]

# Create a sequence of date for the period
date_df <- data.frame(date = seq(from = ymd("2016-12-29"), to = ymd("2017-10-01"), by = "1 day

# crossjoin the date_df and unique_gvkey to get the full combination of gvkey and date
unique_gvkey <- as.data.frame(unique_gvkey)
full_gvkey_date <- cross_join(unique_gvkey, date_df)
colnames(full_gvkey_date) <- c("gvkey", "datadate")

# left join stock_price to full_gvkey_date
stock_data <- merge(x = full_gvkey_date, y = stock_price, by = c("gvkey", "datadate"), all = T

# fill in the missing value with most recent non na value
stock_data <- stock_data %>%
              group_by(gvkey) %>%
```

```
            fill(iid, conm, prccd, prchd, prcld, prcod,cik,sic, .direction = 'down') %>%
            ungroup()
```

Next, we calculate the percentage of daily change (p_daily_change) and the daily volatility (volatile) based on the stock data. It fills any missing values with zeros for certain columns and creates lag variables for both daily change and volatility. Finally, it provides a summary of the updated stock_data. We notice that there is 95 NA values for both daily change and volatility after the lag operation.

```
#calculate the percentage of daily change and mutate as p_daily_change.
stock_data <- stock_data%>%mutate(p_daily_change=(prccd-prcod)/prcod)

#calculate the daily volatile and mutate as volatile.
stock_data <- stock_data%>%mutate(volatile=(prchd-prcld)/prcld)

# fill in with 0 for NA value
stock_data$prccd <- ifelse(is.na(stock_data$prccd), 0, stock_data$prccd)
stock_data$prchd <- ifelse(is.na(stock_data$prchd), 0, stock_data$prchd)
stock_data$prcld <- ifelse(is.na(stock_data$prcld), 0, stock_data$prcld)
stock_data$prcod <- ifelse(is.na(stock_data$prcod), 0, stock_data$prcod)
stock_data$p_daily_change <- ifelse(is.na(stock_data$p_daily_change), 0, stock_data$p_daily_ch
stock_data$volatile <- ifelse(is.na(stock_data$volatile), 0, stock_data$volatile)

# create lag variables for daily change and volatile
stock_data <- stock_data %>% group_by(gvkey) %>% mutate(p_daily_change_lag = lag(p_daily_chang
stock_data <- stock_data %>% group_by(gvkey) %>% mutate(volatile_lag = lag(volatile)) %>% ungr
```

3.5 : Processing company announcements data

The “key_developments.csv” data is loaded and filtered to retain only relevant companies based on CIK. Then, we convert the character column to the date, ensuring it’s in the appropriate format.

```
#load the key_developments data
key_dev <- read.csv("key_developments.csv")

#filter the key_development data to only keep the relevent company
key_dev <- key_dev[key_dev$cik %in% stock_data$cik,]

# change character to date and number
key_dev$date <- mdy(key_dev$kd_date)
key_dev$date <- as.Date(key_dev$date, "%y/%m/%d")
key_dev$cik = as.numeric(key_dev$cik)
```

Next, we creates a data frame (full_cik_date) for the full combination of CIK and dates from stock_data. Then, we merge key_dev with full_cik_date to get final key_dev_data. Lastly, we merge the resulting dataset (key_dev_data) with another dataset named stock_data based on CIKs and dates. This action integrates various financial datasets while handling potential issues such as duplicated records arising from multiple announcements by companies on the same day.

```r
#get the unique cik as save it as an vector
unique_cik <- unique(stock_data$cik)
unique_cik <- as.data.frame(unique_cik)
full_cik_date <- cross_join(unique_cik, date_df)

# change character to date and number
# full_cik_date$date <- mdy(full_cik_date$date)
# full_cik_date$unique_cik = as.numeric(full_cik_date$unique_cik)
summary(full_cik_date)
```

```
   unique_cik            date
 Min.   :   2488   Min.   :2016-12-29
 1st Qu.: 101829   1st Qu.:2017-03-08
 Median : 885306   Median :2017-05-16
 Mean   : 754029   Mean   :2017-05-16
 3rd Qu.:1090872   3rd Qu.:2017-07-24
 Max.   :1707925   Max.   :2017-10-01
 NA's   :277
```

```r
#left join key_dev to full_cik_date
key_dev_data <- merge(x=full_cik_date,y=key_dev,
                      by.x=c("unique_cik","date"),
                      by.y=c("cik","date"),
                      all.x=TRUE)

# merge into the full dataset
# will have duplicated record if on one day a company has multiple announcements
stock_announce <- merge(x = stock_data, y = key_dev_data,
                        by.x = c("cik", "datadate"), by.y = c("unique_cik", "date"),
                        all.x = TRUE, all.y = TRUE)
```

We generate dummy variables indicating the presence of announcements, specifically identifying whether there is an announcement (is_announce) and whether it falls under predefined red flag event categories (red_announce), using predetermined types of red flag announcements from S&P Capital IQ. Subsequently, we count the number of announcements per gvkey and kd_date, merging this information back into the original dataset while handling missing values. This ensures data integrity by retaining only unique combinations of gvkey and datadate, facilitating clearer analysis of announcement data.

```r
# create announcement dummy variable
stock_announce <- stock_announce %>% mutate(is_annouce = ifelse(is.na(kd_type), 0, 1))

# red flag announcement
red_kd_type <- c("Auditor Changes", "Auditor Going Concern Doubts", "Bankruptcy - Filing", "Bu

stock_announce <- stock_announce %>% mutate(red_annouce = ifelse(kd_type %in% red_kd_type, 1, 


# count announcement
number_annoucement <- stock_announce %>% drop_na(kd_date)
number_annoucement <- number_annoucement %>% group_by(gvkey) %>% count(kd_date) %>% ungroup ()
```

```
# merge number of annoucement and fill in NA with 0 and rename
stock_announce <- merge(x = stock_announce, y = number_annoucement, by = c("gvkey", "kd_date")
stock_announce$n <- ifelse(is.na(stock_announce$n), 0, stock_announce$n)
stock_announce <- stock_announce %>% rename(no_annoucement = `n`)

# distinct by gvkey and datadate
stock_announce <- stock_announce %>% distinct(gvkey, datadate,.keep_all = TRUE)
```

3.6 : Processing financial data

First, it loads financial ratio data and adjusts the dividend yield column to be in percentage form. Next, it reads training and testing datasets for followers. Then, it creates a unique vector of gvkey identifiers from the training set and generates a sequence of dates. Later, it performs a cross join operation to obtain a full combination of gvkey and dates, merges the financial ratios with relevant SIC codes, and calculates industry medians for various financial ratios (dpr, cash_ratio, quick_ratio, curr_ratio, inv_turn, sale_nwc). Finally, it filters the financial ratio data to retain only relevant company records based on gvkey and summarizes the resulting dataset. This sequence of operations prepares the financial data for subsequent analysis or modeling tasks.

```
#load the financial ratios data
df <- read.csv("Financial Ratios.csv")

##Turn divyield into percentage form
df$divyield <- as.numeric(sub("%", "", df$divyield)) / 100

#load the followers training and testing datasets
train <- read.csv("Followers_train.csv")
test <- read.csv("Followers_test.csv")

#get the unique gvkey and save it as a vector
unique_gvkey <- unique(train$gvkey)
print(unique_gvkey)
```

```
 [1]   1161   1447   1487   1659   1878   1919   1920   3170   3226   3336
[11]   3708   3814   4060   4201   4321   4839   4988   5492   5742   6435
[21]   6788   6829   7163   8113   8358   8402   8463   8479   8549   9599
[31]   9667  10005  10247  10983  11115  11264  11304  11584  11669  11770
[41]  12233  12441  13092  14412  17035  21238  23225  23252  24468  24800
[51]  25124  25340  25405  26011  27794  27914  28034  28216  28303  28629
[61]  28924  29011  29241  29612  29710  29751  30138  61494  65772 116504
[71] 118502 121718 122841 126554 136725 137131 141384 145049 147661 148470
[81] 150139 156617 160211 164416 164664 165746 170841 174317 177376 178493
[91] 178548 179666 180402 180646 184500
```

```
# Create a sequence of date for the period
date_df <- data.frame(date = seq(from = ymd("2016-09-30"), to = ymd("2017-11-30"), by = "1 day
date_df <- date_df %>% mutate(mon = month(date), y = year(date))
date_df <- date_df %>% arrange(desc(date)) %>% group_by(y, mon) %>% dplyr::slice(1) %>% ungrou
date_df <- select(date_df, -mon, -y) %>% arrange(date_df, date)
```

```r
# crossjoin the date_df and unique_gvkey to get the full combination of gvkey and date
unique_gvkey <- as.data.frame(unique_gvkey)
full_gvkey_date <- cross_join(unique_gvkey, date_df)
colnames(full_gvkey_date) <- c("gvkey", "public_date")

#load the stock price data to get the sic
sic <- read.csv("sic.csv")
sic <- sic %>% select(gvkey, sic) %>% distinct(gvkey, sic)

df <- merge(full_gvkey_date, df, by = c("gvkey", "public_date"), all = TRUE)
df <- merge(df, sic, by = "gvkey", all.x = TRUE)

# create industry median to fill in na later
df <- df %>%
  group_by(sic, public_date) %>%
  mutate(im_bm = median(bm,na.rm = TRUE)) %>%
  mutate(im_pe_exi = median(pe_exi,na.rm = TRUE)) %>%
  mutate(im_ps = median(ps,na.rm = TRUE)) %>%
  mutate(im_pcf = median(pcf,na.rm = TRUE)) %>%
  mutate(im_dpr = median(dpr,na.rm = TRUE)) %>%
  mutate(im_npm = median(npm,na.rm = TRUE)) %>%
  mutate(im_opmbd = median(opmbd,na.rm = TRUE)) %>%
  mutate(im_opmad = median(opmad,na.rm = TRUE)) %>%
  mutate(im_gpm = median(gpm,na.rm = TRUE)) %>%
  mutate(im_roa = median(roa,na.rm = TRUE)) %>%
  mutate(im_roe = median(roe,na.rm = TRUE)) %>%
  mutate(im_roce = median(roce,na.rm = TRUE)) %>%
  mutate(im_debt_at = median(debt_at,na.rm = TRUE)) %>%
  mutate(im_debt_assets = median(debt_assets,na.rm = TRUE)) %>%
  mutate(im_de_ratio = median(de_ratio,na.rm = TRUE)) %>%
  mutate(im_intcov = median(intcov,na.rm = TRUE)) %>%
  mutate(im_cash_ratio = median(cash_ratio,na.rm = TRUE)) %>%
  mutate(im_quick_ratio = median(quick_ratio,na.rm = TRUE)) %>%
  mutate(im_curr_ratio = median(curr_ratio,na.rm = TRUE)) %>%
  mutate(im_inv_turn = median(inv_turn,na.rm = TRUE)) %>%
  mutate(im_at_turn = median(at_turn,na.rm = TRUE)) %>%
  mutate(im_rect_turn = median(rect_turn,na.rm = TRUE)) %>%
  mutate(im_sale_nwc = median(sale_nwc,na.rm = TRUE)) %>%
  mutate(im_ptb = median(ptb,na.rm = TRUE)) %>%
  mutate(im_divyield = median(divyield,na.rm = TRUE)) %>%
  ungroup()

# create industry median (not group by date) for certain variables
# dpr, cash_ratio, quick_ratio, curr_ratio, inv_turn, sale_nwc
df <- df %>%
  group_by(sic) %>%
  mutate(im_total_bm = median(bm,na.rm = TRUE)) %>%
  mutate(im_total_pe_exi = median(pe_exi,na.rm = TRUE)) %>%
  mutate(im_total_ps = median(ps,na.rm = TRUE)) %>%
  mutate(im_total_pcf = median(pcf,na.rm = TRUE)) %>%
  mutate(im_total_dpr = median(dpr,na.rm = TRUE)) %>%
  mutate(im_total_npm = median(npm,na.rm = TRUE)) %>%
  mutate(im_total_opmbd = median(opmbd,na.rm = TRUE)) %>%
  mutate(im_total_opmad = median(opmad,na.rm = TRUE)) %>%
```

```
  mutate(im_total_gpm = median(gpm,na.rm = TRUE)) %>%
  mutate(im_total_roa = median(roa,na.rm = TRUE)) %>%
  mutate(im_total_roe = median(roe,na.rm = TRUE)) %>%
  mutate(im_total_roce = median(roce,na.rm = TRUE)) %>%
  mutate(im_total_debt_at = median(debt_at,na.rm = TRUE)) %>%
  mutate(im_total_debt_assets = median(debt_assets,na.rm = TRUE)) %>%
  mutate(im_total_de_ratio = median(de_ratio,na.rm = TRUE)) %>%
  mutate(im_total_intcov = median(intcov,na.rm = TRUE)) %>%
  mutate(im_total_cash_ratio = median(cash_ratio,na.rm = TRUE)) %>%
  mutate(im_total_quick_ratio = median(quick_ratio,na.rm = TRUE)) %>%
  mutate(im_total_curr_ratio = median(curr_ratio,na.rm = TRUE)) %>%
  mutate(im_total_inv_turn = median(inv_turn,na.rm = TRUE)) %>%
  mutate(im_total_at_turn = median(at_turn,na.rm = TRUE)) %>%
  mutate(im_total_rect_turn = median(rect_turn,na.rm = TRUE)) %>%
  mutate(im_total_sale_nwc = median(sale_nwc,na.rm = TRUE)) %>%
  mutate(im_total_ptb = median(ptb,na.rm = TRUE)) %>%
  mutate(im_total_divyield = median(divyield,na.rm = TRUE)) %>%
  ungroup()

# create industry median (not group by date) for certain variables
# dpr, cash_ratio, quick_ratio, curr_ratio, inv_turn, sale_nwc
df <- df %>%
  mutate(total_bm = median(bm,na.rm = TRUE)) %>%
  mutate(total_pe_exi = median(pe_exi,na.rm = TRUE)) %>%
  mutate(total_ps = median(ps,na.rm = TRUE)) %>%
  mutate(total_pcf = median(pcf,na.rm = TRUE)) %>%
  mutate(total_dpr = median(dpr,na.rm = TRUE)) %>%
  mutate(total_npm = median(npm,na.rm = TRUE)) %>%
  mutate(total_opmbd = median(opmbd,na.rm = TRUE)) %>%
  mutate(total_opmad = median(opmad,na.rm = TRUE)) %>%
  mutate(total_gpm = median(gpm,na.rm = TRUE)) %>%
  mutate(total_roa = median(roa,na.rm = TRUE)) %>%
  mutate(total_roe = median(roe,na.rm = TRUE)) %>%
  mutate(total_roce = median(roce,na.rm = TRUE)) %>%
  mutate(total_debt_at = median(debt_at,na.rm = TRUE)) %>%
  mutate(total_debt_assets = median(debt_assets,na.rm = TRUE)) %>%
  mutate(total_de_ratio = median(de_ratio,na.rm = TRUE)) %>%
  mutate(total_intcov = median(intcov,na.rm = TRUE)) %>%
  mutate(total_cash_ratio = median(cash_ratio,na.rm = TRUE)) %>%
  mutate(total_quick_ratio = median(quick_ratio,na.rm = TRUE)) %>%
  mutate(total_curr_ratio = median(curr_ratio,na.rm = TRUE)) %>%
  mutate(total_inv_turn = median(inv_turn,na.rm = TRUE)) %>%
  mutate(total_at_turn = median(at_turn,na.rm = TRUE)) %>%
  mutate(total_rect_turn = median(rect_turn,na.rm = TRUE)) %>%
  mutate(total_sale_nwc = median(sale_nwc,na.rm = TRUE)) %>%
  mutate(total_ptb = median(ptb,na.rm = TRUE)) %>%
  mutate(total_divyield = median(divyield,na.rm = TRUE)) %>%
  ungroup()

#filter the financial ratio data to keep only the relevant company
unique_gvkey <- unique(train$gvkey)
fin_ratios <- df[df$gvkey %in% unique_gvkey,]
```

We load financial data to computes various financial ratios to fill in missing values. It calculates ratios such as book-to-market ratio (computed_bm), price-to-earnings excluding extraordinary items (computed_pe_exi), price-to-sales (computed_ps), price-to-cash flow (computed_pcf), dividend payout ratio (computed_dpr), net profit margin (computed_npm), operating margin (computed_opmbd), and others. After merging the financials with relevant SIC codes, it computes industry medians for assets (atq). Then, it fills missing values using nearby non-NA values and replaces infinite values with NA. Finally, it summarizes the resulting dataset (`fin_ratios`), revealing the presence of NA values. This process prepares the financial data for further analysis or modeling tasks while addressing missing or erroneous values.

```
#load financials obtained from WDS compustat to compute the financial ratios to fill in missin
financials <- read.csv("Financials.csv")



#compute financial ratios using Financials data and keep asset variables for merging
financials <- financials %>%
  mutate(computed_bm = teqq/mkvaltq,
         computed_pe_exi = epsfxq,
         computed_ps = mkvaltq/saleq,
         computed_pcf = mkvaltq/((fincfy + ivncfy + oancfy)/ cshoq),
         computed_dpr = dvpq/niq,
         computed_npm = niq/saleq,
         computed_opmbd = oibdpq/saleq,
         computed_opmad = oiadpq/saleq,
         computed_gpm = ugiq/saleq,
         computed_roa = niq/atq,
         computed_roe = niq/teqq,
         computed_roce = ibmiiq/icaptq,
         computed_debt_asset = (dlttq+dlcq)/atq,
         computed_de_ratio = ltq/teqq,
         computed_int_cov = niq/tieq,
         computed_cash_ratio = cheq/lctq,
         computed_quick_ratio = (actq-invfgq)/lctq,
         computed_curr_ratio = actq/lctq,
         computed_inv_turn = cogsq/invfgq,
         computed_at_turn = saleq/atq,
         computed_rect_turn = saleq/rectq,
         computed_sale_nwc = saleq/wcapq,
         computed_ptb = mkvaltq/teqq,) %>%
  mutate(public_date = as.Date(datadate))

financials <- merge(financials, sic, by = "gvkey", all.x = TRUE)

financials <- financials %>%
  group_by(sic, public_date) %>%
  mutate(im_atq = median(atq,na.rm = TRUE)) %>%
  ungroup()

median_atq <- financials %>%
  select(sic, public_date, im_atq) %>% distinct(sic, public_date, im_atq)

financials <- financials %>%
  select(gvkey, public_date, atq, computed_bm,computed_pe_exi,computed_ps,computed_pcf,compute
```

```
        computed_npm,computed_opmbd,computed_opmad,computed_gpm,computed_roa,
        computed_roe,computed_roce,computed_debt_asset,computed_de_ratio,
        computed_int_cov,computed_cash_ratio,computed_quick_ratio,
        computed_curr_ratio,computed_inv_turn,computed_at_turn,computed_rect_turn,
        computed_sale_nwc,computed_ptb)

#filter the financial ratio data to only keep the relevant company gvkey
financials <- financials[financials$gvkey %in% unique_gvkey,]

#left_join financials to fin_ratios to fill in the NA values for gvkey1161 with computed dpr a
fin_ratios <- left_join(fin_ratios, financials)
fin_ratios <- left_join(fin_ratios, median_atq)

#fill in the missing value with most recent non na value
fin_ratios <- fin_ratios %>%
  arrange(gvkey, public_date) %>%
  group_by(gvkey) %>%
  fill(atq, computed_bm,computed_pe_exi,computed_ps,computed_pcf,computed_dpr,
       computed_npm,computed_opmbd,computed_opmad,computed_gpm,computed_roa,
       computed_roe,computed_roce,computed_debt_asset,computed_de_ratio,
       computed_int_cov,computed_cash_ratio,computed_quick_ratio,
       computed_curr_ratio,computed_inv_turn,computed_at_turn,computed_rect_turn,
       computed_sale_nwc,computed_ptb,im_atq,.direction = 'down') %>%
  ungroup()

# fill atq from up direction
fin_ratios <- fin_ratios %>%
  arrange(gvkey, public_date) %>%
  group_by(gvkey) %>%
  fill(atq,.direction = 'up') %>%
  ungroup()


# replace inf with NA
fin_ratios <- fin_ratios %>%
  mutate_if(is.numeric, list(~replace(., !is.finite(.), NA)))
summary(fin_ratios) #NA values
```

```
     gvkey          public_date           adate              qdate
 Min.   :  1161   Min.   :2016-09-30   Length:1425        Length:1425
 1st Qu.:  8113   1st Qu.:2016-12-31   Class :character   Class :character
 Median : 23252   Median :2017-04-30   Mode  :character   Mode  :character
 Mean   : 53462   Mean   :2017-04-30
 3rd Qu.:121718   3rd Qu.:2017-08-31
 Max.   :184500   Max.   :2017-11-30

       bm            pe_exi               ps              pcf                dpr
 Min.   :0.0   Min.   :-2325.0   Min.   : 0.1   Min.   :-239.8   Min.   : 0.0
 1st Qu.:0.2   1st Qu.:   12.2   1st Qu.: 0.9   1st Qu.:   8.3   1st Qu.: 0.0
 Median :0.3   Median :   21.4   Median : 2.1   Median :  14.0   Median : 0.3
 Mean   :0.4   Mean   :  -17.1   Mean   : 2.7   Mean   :  17.2   Mean   : 0.7
 3rd Qu.:0.6   3rd Qu.:   32.2   3rd Qu.: 3.4   3rd Qu.:  22.0   3rd Qu.: 0.6
 Max.   :2.3   Max.   :  437.6   Max.   :17.7   Max.   : 243.8   Max.   :31.9
```

```
NA's   :449   NA's   :412       NA's   :389    NA's   :389      NA's   :525
     npm           opmbd          opmad           gpm             roa
Min.   :-0.5   Min.   :-0.2   Min.   :-0.5   Min.   :-0.1   Min.   :-0.1
1st Qu.: 0.0   1st Qu.: 0.1   1st Qu.: 0.0   1st Qu.: 0.3   1st Qu.: 0.1
Median : 0.1   Median : 0.2   Median : 0.1   Median : 0.4   Median : 0.1
Mean   : 0.1   Mean   : 0.2   Mean   : 0.1   Mean   : 0.4   Mean   : 0.1
3rd Qu.: 0.1   3rd Qu.: 0.3   3rd Qu.: 0.2   3rd Qu.: 0.6   3rd Qu.: 0.2
Max.   : 0.4   Max.   : 0.6   Max.   : 0.5   Max.   : 0.9   Max.   : 0.9
NA's   :389    NA's   :389    NA's   :389    NA's   :389    NA's   :389
     roe            roce         debt_at     debt_assets     de_ratio
Min.   :-0.7   Min.   :-0.2   Min.   :0.0   Min.   :0.1   Min.   :-141.4
1st Qu.: 0.0   1st Qu.: 0.1   1st Qu.:0.2   1st Qu.:0.5   1st Qu.:   0.9
Median : 0.1   Median : 0.1   Median :0.3   Median :0.7   Median :   1.7
Mean   : 0.2   Mean   : 0.2   Mean   :0.3   Mean   :0.7   Mean   :   2.7
3rd Qu.: 0.2   3rd Qu.: 0.2   3rd Qu.:0.4   3rd Qu.:0.8   3rd Qu.:   3.8
Max.   : 5.0   Max.   : 1.6   Max.   :3.2   Max.   :3.7   Max.   : 146.7
NA's   :455    NA's   :396    NA's   :389   NA's   :389   NA's   :389
    intcov       cash_ratio   quick_ratio    curr_ratio     inv_turn
Min.   :  -9   Min.   :0.0   Min.   :0.1   Min.   :0.4   Min.   :  1.1
1st Qu.:   3   1st Qu.:0.2   1st Qu.:0.8   1st Qu.:1.2   1st Qu.:  3.7
Median :   6   Median :0.4   Median :1.1   Median :1.4   Median :  5.3
Mean   :  36   Mean   :0.8   Mean   :1.4   Mean   :1.8   Mean   : 22.0
3rd Qu.:   9   3rd Qu.:1.0   3rd Qu.:1.7   3rd Qu.:2.0   3rd Qu.: 11.5
Max.   :7083   Max.   :6.3   Max.   :7.1   Max.   :7.3   Max.   :444.9
NA's   :470    NA's   :493   NA's   :493   NA's   :493   NA's   :622
   at_turn      rect_turn        sale_nwc           ptb          divyield
Min.   :0.1   Min.   :  0.1   Min.   :   0.6   Min.   : 0.3   Min.   :0.0
1st Qu.:0.5   1st Qu.:  5.4   1st Qu.:   3.5   1st Qu.: 1.6   1st Qu.:0.0
Median :0.7   Median :  7.7   Median :   6.6   Median : 3.2   Median :0.0
Mean   :1.0   Mean   : 15.5   Mean   :  25.2   Mean   : 6.8   Mean   :0.0
3rd Qu.:1.4   3rd Qu.: 12.9   3rd Qu.:  13.8   3rd Qu.: 6.3   3rd Qu.:0.0
Max.   :3.7   Max.   :118.9   Max.   :2500.7   Max.   :78.0   Max.   :0.1
NA's   :389   NA's   :402     NA's   :623      NA's   :449    NA's   :750
   TICKER                sic           im_bm        im_pe_exi
Length:1425        Min.   :1311   Min.   :0.08   Min.   :-79.8
Class :character   1st Qu.:3561   1st Qu.:0.29   1st Qu.: 10.1
Mode  :character   Median :4911   Median :0.40   Median : 18.0
                   Mean   :4838   Mean   :0.47   Mean   : 15.1
                   3rd Qu.:6211   3rd Qu.:0.60   3rd Qu.: 22.6
                   Max.   :8742   Max.   :2.25   Max.   : 59.1
                                  NA's   :190    NA's   :190
    im_ps           im_pcf           im_dpr         im_npm
Min.   : 0.12   Min.   : -4.82   Min.   :0.00   Min.   :-5.76
1st Qu.: 0.93   1st Qu.:  7.52   1st Qu.:0.00   1st Qu.: 0.01
Median : 1.58   Median :  9.84   Median :0.10   Median : 0.04
Mean   : 2.61   Mean   : 10.92   Mean   :0.23   Mean   :-0.14
3rd Qu.: 2.72   3rd Qu.: 14.70   3rd Qu.:0.36   3rd Qu.: 0.07
Max.   :38.39   Max.   :123.00   Max.   :2.34   Max.   : 0.16
NA's   :190     NA's   :190      NA's   :194    NA's   :190
   im_opmbd        im_opmad         im_gpm          im_roa
Min.   :-5.36   Min.   :-5.58   Min.   :-4.62   Min.   :-0.51
1st Qu.: 0.06   1st Qu.: 0.03   1st Qu.: 0.21   1st Qu.: 0.06
Median : 0.12   Median : 0.07   Median : 0.35   Median : 0.10
Mean   :-0.03   Mean   :-0.09   Mean   : 0.24   Mean   : 0.07
```

```
3rd Qu.: 0.17   3rd Qu.: 0.12   3rd Qu.: 0.51   3rd Qu.: 0.13
Max.   : 0.43   Max.   : 0.29   Max.   : 0.78   Max.   : 0.21
NA's   :190     NA's   :190     NA's   :190     NA's   :190
    im_roe         im_roce        im_debt_at   im_debt_assets  im_de_ratio
Min.   :-0.71   Min.   :-0.63   Min.   :0.00   Min.   :0.19   Min.   :0.23
1st Qu.: 0.02   1st Qu.: 0.06   1st Qu.:0.09   1st Qu.:0.46   1st Qu.:0.76
Median : 0.07   Median : 0.09   Median :0.22   Median :0.55   Median :1.08
Mean   : 0.04   Mean   : 0.07   Mean   :0.21   Mean   :0.55   Mean   :1.45
3rd Qu.: 0.11   3rd Qu.: 0.13   3rd Qu.:0.29   3rd Qu.:0.64   3rd Qu.:1.69
Max.   : 0.41   Max.   : 0.58   Max.   :0.66   Max.   :0.89   Max.   :8.42
NA's   :190     NA's   :190     NA's   :190    NA's   :190    NA's   :190
  im_intcov      im_cash_ratio  im_quick_ratio im_curr_ratio   im_inv_turn
Min.   :-43.05   Min.   :0.03   Min.   :0.23   Min.   :0.66   Min.   :  0.6
1st Qu.:  1.87   1st Qu.:0.28   1st Qu.:1.02   1st Qu.:1.41   1st Qu.:  3.5
Median :  3.76   Median :0.49   Median :1.45   Median :1.80   Median :  5.3
Mean   :  3.66   Mean   :0.90   Mean   :1.67   Mean   :2.18   Mean   : 22.3
3rd Qu.:  5.78   3rd Qu.:1.19   3rd Qu.:2.10   3rd Qu.:2.64   3rd Qu.: 17.0
Max.   : 61.36   Max.   :6.26   Max.   :6.62   Max.   :6.72   Max.   :446.3
NA's   :190      NA's   :255    NA's   :255    NA's   :255    NA's   :252
  im_at_turn    im_rect_turn    im_sale_nwc         im_ptb       im_divyield
Min.   :0.07   Min.   : 0.28   Min.   :-84.96   Min.   : 0.49   Min.   :0.00
1st Qu.:0.58   1st Qu.: 5.50   1st Qu.:  2.37   1st Qu.: 1.65   1st Qu.:0.01
Median :0.78   Median : 6.62   Median :  4.79   Median : 2.41   Median :0.02
Mean   :0.92   Mean   :13.37   Mean   :  6.63   Mean   : 2.77   Mean   :0.02
3rd Qu.:1.27   3rd Qu.: 9.92   3rd Qu.:  8.64   3rd Qu.: 3.49   3rd Qu.:0.03
Max.   :2.81   Max.   :80.63   Max.   : 29.27   Max.   :14.46   Max.   :0.12
NA's   :190    NA's   :190     NA's   :255      NA's   :190     NA's   :203
 im_total_bm    im_total_pe_exi  im_total_ps      im_total_pcf
Min.   :0.113   Min.   :-9.13   Min.   : 0.174   Min.   :-4.34
1st Qu.:0.284   1st Qu.:10.71   1st Qu.: 0.986   1st Qu.: 7.49
Median :0.411   Median :17.79   Median : 1.589   Median : 9.63
Mean   :0.472   Mean   :15.07   Mean   : 2.554   Mean   :10.67
3rd Qu.:0.615   3rd Qu.:23.16   3rd Qu.: 2.704   3rd Qu.:14.46
Max.   :1.236   Max.   :41.58   Max.   :25.767   Max.   :24.12

 im_total_dpr    im_total_npm    im_total_opmbd   im_total_opmad
Min.   :0.000   Min.   :-5.391   Min.   :-4.978   Min.   :-5.125
1st Qu.:0.000   1st Qu.: 0.008   1st Qu.: 0.062   1st Qu.: 0.032
Median :0.053   Median : 0.038   Median : 0.115   Median : 0.074
Mean   :0.225   Mean   :-0.143   Mean   :-0.026   Mean   :-0.086
3rd Qu.:0.356   3rd Qu.: 0.065   3rd Qu.: 0.169   3rd Qu.: 0.126
Max.   :2.092   Max.   : 0.130   Max.   : 0.366   Max.   : 0.273

 im_total_gpm     im_total_roa      im_total_roe     im_total_roce
Min.   :-4.007   Min.   :-0.4730   Min.   :-0.6250   Min.   :-0.5710
1st Qu.: 0.212   1st Qu.: 0.0540   1st Qu.: 0.0250   1st Qu.: 0.0560
Median : 0.352   Median : 0.1020   Median : 0.0790   Median : 0.0920
Mean   : 0.250   Mean   : 0.0763   Mean   : 0.0422   Mean   : 0.0714
3rd Qu.: 0.543   3rd Qu.: 0.1280   3rd Qu.: 0.1170   3rd Qu.: 0.1340
Max.   : 0.764   Max.   : 0.2080   Max.   : 0.3070   Max.   : 0.3575

im_total_debt_at im_total_debt_assets im_total_de_ratio im_total_intcov
Min.   :0.000    Min.   :0.203        Min.   :0.254     Min.   :-25.55
1st Qu.:0.093    1st Qu.:0.452        1st Qu.:0.790     1st Qu.:  2.02
```

```
Median :0.214   Median :0.541      Median :1.058    Median :  3.50  
Mean   :0.214   Mean   :0.544      Mean   :1.441    Mean   :  4.09  
3rd Qu.:0.293   3rd Qu.:0.638      3rd Qu.:1.628    3rd Qu.:  5.74  
Max.   :0.643   Max.   :0.893      Max.   :8.311    Max.   : 53.75  

im_total_cash_ratio im_total_quick_ratio im_total_curr_ratio im_total_inv_turn
Min.   :0.04        Min.   :0.26         Min.   :0.75        Min.   :  0.59   
1st Qu.:0.26        1st Qu.:1.02         1st Qu.:1.40        1st Qu.:  3.47   
Median :0.49        Median :1.45         Median :1.80        Median :  5.31   
Mean   :0.90        Mean   :1.66         Mean   :2.17        Mean   : 18.47   
3rd Qu.:1.20        3rd Qu.:2.06         3rd Qu.:2.65        3rd Qu.: 17.02   
Max.   :5.09        Max.   :5.40         Max.   :5.56        Max.   :211.60   
NA's   :75          NA's   :75           NA's   :75          NA's   :60       
im_total_at_turn im_total_rect_turn im_total_sale_nwc  im_total_ptb  
Min.   :0.076    Min.   : 0.29      Min.   : 0.14     Min.   :0.875  
1st Qu.:0.565    1st Qu.: 5.56      1st Qu.: 2.30     1st Qu.:1.624  
Median :0.779    Median : 6.41      Median : 4.77     Median :2.406  
Mean   :0.905    Mean   :13.40      Mean   : 6.74     Mean   :2.730  
3rd Qu.:1.240    3rd Qu.: 9.89      3rd Qu.: 8.31     3rd Qu.:3.542  
Max.   :2.745    Max.   :76.28      Max.   :23.57     Max.   :9.021  
                                    NA's   :75                       
im_total_divyield    total_bm      total_pe_exi     total_ps      total_pcf   
Min.   :0.006     Min.   :0.479   Min.   :15.2   Min.   :1.95   Min.   :9.66  
1st Qu.:0.014     1st Qu.:0.479   1st Qu.:15.2   1st Qu.:1.95   1st Qu.:9.66  
Median :0.018     Median :0.479   Median :15.2   Median :1.95   Median :9.66  
Mean   :0.021     Mean   :0.479   Mean   :15.2   Mean   :1.95   Mean   :9.66  
3rd Qu.:0.025     3rd Qu.:0.479   3rd Qu.:15.2   3rd Qu.:1.95   3rd Qu.:9.66  
Max.   :0.069     Max.   :0.479   Max.   :15.2   Max.   :1.95   Max.   :9.66  
NA's   :15                                                                    
  total_dpr       total_npm      total_opmbd     total_opmad   
Min.   :0.199   Min.   :0.037   Min.   :0.131   Min.   :0.079  
1st Qu.:0.199   1st Qu.:0.037   1st Qu.:0.131   1st Qu.:0.079  
Median :0.199   Median :0.037   Median :0.131   Median :0.079  
Mean   :0.199   Mean   :0.037   Mean   :0.131   Mean   :0.079  
3rd Qu.:0.199   3rd Qu.:0.037   3rd Qu.:0.131   3rd Qu.:0.079  
Max.   :0.199   Max.   :0.037   Max.   :0.131   Max.   :0.079  

  total_gpm       total_roa       total_roe       total_roce   
Min.   :0.386   Min.   :0.068   Min.   :0.059   Min.   :0.081  
1st Qu.:0.386   1st Qu.:0.068   1st Qu.:0.059   1st Qu.:0.081  
Median :0.386   Median :0.068   Median :0.059   Median :0.081  
Mean   :0.386   Mean   :0.068   Mean   :0.059   Mean   :0.081  
3rd Qu.:0.386   3rd Qu.:0.068   3rd Qu.:0.059   3rd Qu.:0.081  
Max.   :0.386   Max.   :0.068   Max.   :0.059   Max.   :0.081  

total_debt_at   total_debt_assets total_de_ratio  total_intcov
Min.   :0.182   Min.   :0.592     Min.   :1.22   Min.   :2.3  
1st Qu.:0.182   1st Qu.:0.592     1st Qu.:1.22   1st Qu.:2.3  
Median :0.182   Median :0.592     Median :1.22   Median :2.3  
Mean   :0.182   Mean   :0.592     Mean   :1.22   Mean   :2.3  
3rd Qu.:0.182   3rd Qu.:0.592     3rd Qu.:1.22   3rd Qu.:2.3  
Max.   :0.182   Max.   :0.592     Max.   :1.22   Max.   :2.3  

total_cash_ratio total_quick_ratio total_curr_ratio total_inv_turn
```

```
Min.   :0.654    Min.   :1.51      Min.   :2.01     Min.   :4.66
1st Qu.:0.654    1st Qu.:1.51      1st Qu.:2.01     1st Qu.:4.66
Median :0.654    Median :1.51      Median :2.01     Median :4.66
Mean   :0.654    Mean   :1.51      Mean   :2.01     Mean   :4.66
3rd Qu.:0.654    3rd Qu.:1.51      3rd Qu.:2.01     3rd Qu.:4.66
Max.   :0.654    Max.   :1.51      Max.   :2.01     Max.   :4.66

total_at_turn   total_rect_turn total_sale_nwc   total_ptb    total_divyield
Min.   :0.629   Min.   :6.24    Min.   :3.68   Min.   :2.09   Min.   :0.0187
1st Qu.:0.629   1st Qu.:6.24    1st Qu.:3.68   1st Qu.:2.09   1st Qu.:0.0187
Median :0.629   Median :6.24    Median :3.68   Median :2.09   Median :0.0187
Mean   :0.629   Mean   :6.24    Mean   :3.68   Mean   :2.09   Mean   :0.0187
3rd Qu.:0.629   3rd Qu.:6.24    3rd Qu.:3.68   3rd Qu.:2.09   3rd Qu.:0.0187
Max.   :0.629   Max.   :6.24    Max.   :3.68   Max.   :2.09   Max.   :0.0187

     atq          computed_bm    computed_pe_exi  computed_ps
Min.   :   154   Min.   :-0.34   Min.   :-2.93   Min.   : 0.56
1st Qu.:  2167   1st Qu.: 0.13   1st Qu.: 0.18   1st Qu.: 3.64
Median :  5648   Median : 0.30   Median : 0.54   Median : 7.94
Mean   : 28689   Mean   : 0.36   Mean   : 0.81   Mean   :10.14
3rd Qu.: 18419   3rd Qu.: 0.52   3rd Qu.: 1.07   3rd Qu.:12.99
Max.   :514568   Max.   : 1.83   Max.   : 7.68   Max.   :53.41
NA's   :150      NA's   :181     NA's   :181     NA's   :181
 computed_pcf        computed_dpr   computed_npm    computed_opmbd
Min.   :-48178775   Min.   :0.00   Min.   :-1.14   Min.   :-1.56
1st Qu.:    -9006   1st Qu.:0.00   1st Qu.: 0.02   1st Qu.: 0.08
Median :      937   Median :0.00   Median : 0.07   Median : 0.17
Mean   :    -8371   Mean   :0.01   Mean   : 0.07   Mean   : 0.18
3rd Qu.:    15299   3rd Qu.:0.00   3rd Qu.: 0.13   3rd Qu.: 0.25
Max.   : 33324132   Max.   :1.63   Max.   : 1.29   Max.   : 0.62
NA's   :181         NA's   :166    NA's   :166     NA's   :217
computed_opmad   computed_gpm   computed_roa    computed_roe
Min.   :-1.66   Min.   :-1.1   Min.   :-0.16   Min.   :-114.00
1st Qu.: 0.05   1st Qu.: 0.1   1st Qu.: 0.00   1st Qu.:   0.01
Median : 0.13   Median : 0.1   Median : 0.01   Median :   0.03
Mean   : 0.13   Mean   : 0.1   Mean   : 0.01   Mean   :  -0.10
3rd Qu.: 0.20   3rd Qu.: 0.2   3rd Qu.: 0.02   3rd Qu.:   0.06
Max.   : 0.56   Max.   : 0.3   Max.   : 0.10   Max.   :  35.65
NA's   :166     NA's   :1365   NA's   :166     NA's   :166
computed_roce   computed_debt_asset computed_de_ratio computed_int_cov
Min.   :-0.20   Min.   :0.00        Min.   :-2490.6   Min.   : NA
1st Qu.: 0.01   1st Qu.:0.16        1st Qu.:    0.9   1st Qu.: NA
Median : 0.02   Median :0.29        Median :    1.7   Median : NA
Mean   : 0.02   Mean   :0.33        Mean   :   -1.3   Mean   :NaN
3rd Qu.: 0.04   3rd Qu.:0.42        3rd Qu.:    3.9   3rd Qu.: NA
Max.   : 0.27   Max.   :3.87        Max.   :  712.1   Max.   : NA
NA's   :226     NA's   :199         NA's   :166       NA's   :1425
computed_cash_ratio computed_quick_ratio computed_curr_ratio computed_inv_turn
Min.   :0.00        Min.   :0.1          Min.   :0.40        Min.   : 0.4
1st Qu.:0.15        1st Qu.:1.0          1st Qu.:1.09        1st Qu.: 1.2
Median :0.42        Median :1.4          Median :1.46        Median : 2.1
Mean   :0.74        Mean   :1.7          Mean   :1.81        Mean   : 3.3
3rd Qu.:0.96        3rd Qu.:2.1          3rd Qu.:2.03        3rd Qu.: 3.7
Max.   :6.33        Max.   :7.2          Max.   :7.29        Max.   :29.9
```

```
 NA's   :315         NA's   :659          NA's   :315         NA's   :865
 computed_at_turn computed_rect_turn computed_sale_nwc  computed_ptb
 Min.   :0.02     Min.   : 0.04      Min.   : -45.8     Min.   :-12930
 1st Qu.:0.11     1st Qu.: 1.35      1st Qu.:   0.4     1st Qu.:     2
 Median :0.17     Median : 1.96      Median :   1.1     Median :     3
 Mean   :0.24     Mean   : 3.90      Mean   :   8.6     Mean   :   -14
 3rd Qu.:0.35     3rd Qu.: 3.26      3rd Qu.:   2.6     3rd Qu.:     6
 Max.   :1.14     Max.   :36.93      Max.   :1817.7     Max.   :  3399
 NA's   :166      NA's   :183        NA's   :315        NA's   :181
     im_atq
 Min.   :     0
 1st Qu.:   198
 Median :   842
 Mean   :  6206
 3rd Qu.:  4481
 Max.   :130522
 NA's   :5
```

Next, we aim to fill in missing values in financial ratios (`fin_ratios`). We first replaces the NA values of specific financial ratios (e.g., bm, pe_exi, dpr, roe, roce, intcov, cash_ratio, etc.) with computed values. If the financials of these financial ratios are missing, we will then replaced the remaining NA values with industry medians. If NA values still exist after filling in with computed financial ratios and industry medians, we then continues to fill in remaining NA values using total industry medians. After imputation, it removes variables that might cause multicollinearity issues. The process iteratively handles missing data, ensuring that the dataset is prepared for subsequent analysis.

```r
# create a function for repeated fill in
fill_in <- function(data, var, fillin){
  data[[var]] <- ifelse(is.na(data[[var]]) | data[[var]] == "", data[[fillin]], data[[var]])
  return(data)
}

#fill in with quarterly computed_fin_ratios for NA value
fin_ratios <- fill_in(fin_ratios, "bm", "computed_bm")
fin_ratios <- fill_in(fin_ratios, "pe_exi", "computed_pe_exi")
fin_ratios <- fill_in(fin_ratios, "dpr", "computed_dpr")
fin_ratios <- fill_in(fin_ratios, "roe", "computed_roe")
fin_ratios <- fill_in(fin_ratios, "roce", "computed_roce")
fin_ratios <- fill_in(fin_ratios, "intcov", "computed_int_cov")
fin_ratios <- fill_in(fin_ratios, "cash_ratio", "computed_cash_ratio")
fin_ratios <- fill_in(fin_ratios, "quick_ratio", "computed_quick_ratio")
fin_ratios <- fill_in(fin_ratios, "curr_ratio", "computed_curr_ratio")
fin_ratios <- fill_in(fin_ratios, "inv_turn", "computed_inv_turn")
fin_ratios <- fill_in(fin_ratios, "rect_turn", "computed_rect_turn")
fin_ratios <- fill_in(fin_ratios, "sale_nwc", "computed_sale_nwc")
fin_ratios <- fill_in(fin_ratios, "ptb", "computed_ptb")

# still has NA values bm 181
# fill in with industry median on specific day for remaining NA value
fin_ratios <- fill_in(fin_ratios, "bm", "im_bm")
fin_ratios <- fill_in(fin_ratios, "pe_exi", "im_pe_exi")
fin_ratios <- fill_in(fin_ratios, "ps", "im_ps")
fin_ratios <- fill_in(fin_ratios, "pcf", "im_pcf")
```

```r
fin_ratios <- fill_in(fin_ratios, "dpr", "im_dpr")
fin_ratios <- fill_in(fin_ratios, "npm", "im_npm")
fin_ratios <- fill_in(fin_ratios, "opmbd", "im_opmbd")
fin_ratios <- fill_in(fin_ratios, "opmad", "im_opmad")
fin_ratios <- fill_in(fin_ratios, "gpm", "im_gpm")
fin_ratios <- fill_in(fin_ratios, "roa", "im_roa")
fin_ratios <- fill_in(fin_ratios, "roe", "im_roe")
fin_ratios <- fill_in(fin_ratios, "roce", "im_roce")
fin_ratios <- fill_in(fin_ratios, "debt_at", "im_debt_at")
fin_ratios <- fill_in(fin_ratios, "debt_assets", "im_debt_assets")
fin_ratios <- fill_in(fin_ratios, "de_ratio", "im_de_ratio")
fin_ratios <- fill_in(fin_ratios, "intcov", "im_intcov")
fin_ratios <- fill_in(fin_ratios, "cash_ratio", "im_cash_ratio")
fin_ratios <- fill_in(fin_ratios, "quick_ratio", "im_quick_ratio")
fin_ratios <- fill_in(fin_ratios, "curr_ratio", "im_curr_ratio")
fin_ratios <- fill_in(fin_ratios, "inv_turn", "im_inv_turn")
fin_ratios <- fill_in(fin_ratios, "at_turn", "im_at_turn")
fin_ratios <- fill_in(fin_ratios, "rect_turn", "im_rect_turn")
fin_ratios <- fill_in(fin_ratios, "sale_nwc", "im_sale_nwc")
fin_ratios <- fill_in(fin_ratios, "ptb", "im_ptb")
fin_ratios <- fill_in(fin_ratios, "divyield", "im_divyield")


# still has NA values bm 38
# fill in with industry median for remaining NA value
fin_ratios <- fill_in(fin_ratios, "bm", "im_total_bm")
fin_ratios <- fill_in(fin_ratios, "pe_exi", "im_total_pe_exi")
fin_ratios <- fill_in(fin_ratios, "ps", "im_total_ps")
fin_ratios <- fill_in(fin_ratios, "pcf", "im_total_pcf")
fin_ratios <- fill_in(fin_ratios, "dpr", "im_total_dpr")
fin_ratios <- fill_in(fin_ratios, "npm", "im_total_npm")
fin_ratios <- fill_in(fin_ratios, "opmbd", "im_total_opmbd")
fin_ratios <- fill_in(fin_ratios, "opmad", "im_total_opmad")
fin_ratios <- fill_in(fin_ratios, "gpm", "im_total_gpm")
fin_ratios <- fill_in(fin_ratios, "roa", "im_total_roa")
fin_ratios <- fill_in(fin_ratios, "roe", "im_total_roe")
fin_ratios <- fill_in(fin_ratios, "roce", "im_total_roce")
fin_ratios <- fill_in(fin_ratios, "debt_at", "im_total_debt_at")
fin_ratios <- fill_in(fin_ratios, "debt_assets", "im_total_debt_assets")
fin_ratios <- fill_in(fin_ratios, "de_ratio", "im_total_de_ratio")
fin_ratios <- fill_in(fin_ratios, "intcov", "im_total_intcov")
fin_ratios <- fill_in(fin_ratios, "cash_ratio", "im_total_cash_ratio")
fin_ratios <- fill_in(fin_ratios, "quick_ratio", "im_total_quick_ratio")
fin_ratios <- fill_in(fin_ratios, "curr_ratio", "im_total_curr_ratio")
fin_ratios <- fill_in(fin_ratios, "inv_turn", "im_total_inv_turn")
fin_ratios <- fill_in(fin_ratios, "at_turn", "im_total_at_turn")
fin_ratios <- fill_in(fin_ratios, "rect_turn", "im_total_rect_turn")
fin_ratios <- fill_in(fin_ratios, "sale_nwc", "im_total_sale_nwc")
fin_ratios <- fill_in(fin_ratios, "ptb", "im_total_ptb")
fin_ratios <- fill_in(fin_ratios, "divyield", "im_total_divyield")


# still has NA values bm 0， atq 150
# fill in with total median for remaining NA value
```

```r
fin_ratios <- fill_in(fin_ratios, "cash_ratio", "total_cash_ratio")
fin_ratios <- fill_in(fin_ratios, "quick_ratio", "total_quick_ratio")
fin_ratios <- fill_in(fin_ratios, "curr_ratio", "total_curr_ratio")
fin_ratios <- fill_in(fin_ratios, "inv_turn", "total_inv_turn")
fin_ratios <- fill_in(fin_ratios, "sale_nwc", "total_sale_nwc")
fin_ratios <- fill_in(fin_ratios, "divyield", "total_divyield")
fin_ratios <- fill_in(fin_ratios, "atq", "im_atq")

# no NA values for financial ratio, atq 3
# remove variables for imputation
fin_ratios <- fin_ratios %>%
  select (gvkey, public_date, bm, pe_exi, ps, pcf, dpr, npm, opmbd, opmad, gpm, roa, roe, roce
          de_ratio, intcov, cash_ratio, quick_ratio, curr_ratio, inv_turn, at_turn, debt_asset
          rect_turn, sale_nwc, divyield, TICKER, atq)

summary(fin_ratios)
```

```
     gvkey         public_date                bm             pe_exi
 Min.   :  1161   Min.   :2016-09-30   Min.   :-0.339   Min.   :-2325.0
 1st Qu.:  8113   1st Qu.:2016-12-31   1st Qu.: 0.182   1st Qu.:    0.6
 Median : 23252   Median :2017-04-30   Median : 0.332   Median :   18.2
 Mean   : 53462   Mean   :2017-04-30   Mean   : 0.411   Mean   :  -10.6
 3rd Qu.:121718   3rd Qu.:2017-08-31   3rd Qu.: 0.600   3rd Qu.:   26.1
 Max.   :184500   Max.   :2017-11-30   Max.   : 2.346   Max.   :  437.6

       ps              pcf                dpr              npm
 Min.   : 0.07   Min.   :-239.75   Min.   : 0.00   Min.   :-5.758
 1st Qu.: 0.98   1st Qu.:    7.98   1st Qu.: 0.00   1st Qu.: 0.014
 Median : 2.05   Median :   13.05   Median : 0.11   Median : 0.057
 Mean   : 2.84   Mean   :   15.45   Mean   : 0.48   Mean   :-0.012
 3rd Qu.: 3.21   3rd Qu.:   19.84   3rd Qu.: 0.45   3rd Qu.: 0.107
 Max.   :38.39   Max.   : 243.78   Max.   :31.92   Max.   : 0.388

     opmbd             opmad              gpm               roa
 Min.   :-5.363   Min.   :-5.577   Min.   :-4.616   Min.   :-0.509
 1st Qu.: 0.076   1st Qu.: 0.044   1st Qu.: 0.268   1st Qu.: 0.071
 Median : 0.140   Median : 0.100   Median : 0.376   Median : 0.112
 Mean   : 0.103   Mean   : 0.047   Mean   : 0.363   Mean   : 0.124
 3rd Qu.: 0.239   3rd Qu.: 0.186   3rd Qu.: 0.585   3rd Qu.: 0.166
 Max.   : 0.556   Max.   : 0.501   Max.   : 0.912   Max.   : 0.895

      roe                roce             debt_at          de_ratio
 Min.   :-114.00   Min.   :-0.627   Min.   :0.000   Min.   :-141.44
 1st Qu.:    0.01   1st Qu.: 0.032   1st Qu.:0.112   1st Qu.:    0.83
 Median :    0.10   Median : 0.111   Median :0.256   Median :    1.48
 Mean   :   -0.02   Mean   : 0.146   Mean   :0.286   Mean   :    2.30
 3rd Qu.:    0.18   3rd Qu.: 0.188   3rd Qu.:0.385   3rd Qu.:    2.70
 Max.   :  35.65   Max.   : 1.570   Max.   :3.176   Max.   : 146.68

     intcov         cash_ratio       quick_ratio       curr_ratio        inv_turn
 Min.   : -33   Min.   :0.000   Min.   :0.129   Min.   :0.401   Min.   :  0.5
 1st Qu.:   2   1st Qu.:0.212   1st Qu.:0.886   1st Qu.:1.244   1st Qu.:  3.7
 Median :   5   Median :0.494   Median :1.306   Median :1.590   Median :  5.3
```

```
Mean   :  25   Mean   :0.840   Mean   :1.581   Mean   :1.979   Mean   : 26.0
3rd Qu.:   8   3rd Qu.:1.102   3rd Qu.:1.925   3rd Qu.:2.192   3rd Qu.: 17.0
Max.   :7083   Max.   :6.328   Max.   :7.076   Max.   :7.286   Max.   :446.3

    at_turn       debt_assets         ptb            rect_turn
Min.   :0.076   Min.   :0.107   Min.   :-12930   Min.   :  0.04
1st Qu.:0.475   1st Qu.:0.481   1st Qu.:      1   1st Qu.:  4.21
Median :0.739   Median :0.614   Median :      3   Median :  6.41
Mean   :0.967   Mean   :0.642   Mean   :    -13   Mean   : 13.72
3rd Qu.:1.394   3rd Qu.:0.752   3rd Qu.:      5   3rd Qu.: 12.20
Max.   :3.670   Max.   :3.722   Max.   :   3399   Max.   :118.88

    sale_nwc         divyield           TICKER               atq
Min.   : -85.0   Min.   :0.00227   Length:1425        Min.   :     6
1st Qu.:   1.6   1st Qu.:0.01460   Class :character   1st Qu.:  1588
Median :   3.7   Median :0.01920   Mode  :character   Median :  5228
Mean   :  14.8   Mean   :0.02250                      Mean   : 26684
3rd Qu.:   9.7   3rd Qu.:0.02745                      3rd Qu.: 17630
Max.   :2500.7   Max.   :0.12100                      Max.   :514568
                                                      NA's   :3
```

We create monthly lag variables for each financial ratio in the `fin_ratios` dataset. It groups the data by `gvkey` and computes lagged values for each financial ratio. These lag variables enable the analysis of how changes in financial ratios over time correlate with subsequent outcomes, providing insights into financial performance trends and dynamics.

```
# create monthly lag variables
fin_ratios <- fin_ratios %>%
  group_by(gvkey) %>%
  mutate(bm_lag = lag(bm)) %>%
  mutate(pe_exi_lag = lag(pe_exi)) %>%
  mutate(ps_lag = lag(ps)) %>%
  mutate(pcf_lag = lag(pcf)) %>%
  mutate(dpr_lag = lag(dpr)) %>%
  mutate(npm_lag = lag(npm)) %>%
  mutate(opmbd_lag = lag(opmbd)) %>%
  mutate(opmad_lag = lag(opmad)) %>%
  mutate(gpm_lag = lag(gpm)) %>%
  mutate(roa_lag = lag(roa)) %>%
  mutate(roe_lag = lag(roe)) %>%
  mutate(roce_lag = lag(roce)) %>%
  mutate(debt_at_lag = lag(debt_at)) %>%
  mutate(de_ratio_lag = lag(de_ratio)) %>%
  mutate(intcov_lag = lag(intcov)) %>%
  mutate(cash_ratio_lag = lag(cash_ratio)) %>%
  mutate(quick_ratio_lag = lag(quick_ratio)) %>%
  mutate(curr_ratio_lag = lag(curr_ratio)) %>%
  mutate(inv_turn_lag = lag(inv_turn)) %>%
  mutate(at_turn_lag = lag(at_turn)) %>%
  mutate(rect_turn_lag = lag(rect_turn)) %>%
  mutate(sale_nwc_lag = lag(sale_nwc)) %>%
  mutate(divyield_lag = lag(divyield)) %>%
  mutate(debt_assets_lag = lag(debt_assets)) %>%
```

```
  mutate(ptb_lag = lag(ptb)) %>%
  ungroup()
```

We creating a `date_df` dataframe with a sequence of dates from 31 December 2016 to 30 November 2017. Then, it performs a cross-join operation between the unique `gvkey` values and `date_df` to obtain a full combination of `gvkey` and dates, stored in `full_gvkey_date`. Next, it left-joins `fin_ratios` with `full_gvkey_date`. After that, it fills in missing values in `fin_ratios` with the most recent non-NA value. This ensures that each `gvkey` has complete data for all dates in the specified period. Finally, the `summary` function is used to verify that there are no missing values left in `fin_ratios`.

```
#create a sequence of date for the period
date_df <- data.frame(date = seq(from = ymd("2016-12-31"), to = ymd("2017-11-30"), by = "1 day

#crossjoin date_df and unique_gvkey to obtain full combination of gvkey and date
unique_gvkey <- as.data.frame(unique_gvkey)
full_gvkey_date <- cross_join(unique_gvkey, date_df)
colnames(full_gvkey_date) <- c("gvkey", "public_date")

#left join fin_ratios to full_gvkey_date
fin_ratios <- merge(x = full_gvkey_date, y = fin_ratios, by = c("gvkey", "public_date"), all.x


#fill in the missing value with most recent non na value
fin_ratios <- fin_ratios %>%
  arrange(gvkey, public_date) %>%
  group_by(gvkey) %>%
  fill(bm, pe_exi, ps, pcf, dpr, npm, opmbd, opmad, gpm, roa, roe, roce, debt_at, de_ratio, in
       cash_ratio, quick_ratio, curr_ratio, inv_turn, at_turn, rect_turn, debt_assets, ptb,
       sale_nwc, divyield, atq, bm_lag, pe_exi_lag, ps_lag, pcf_lag, dpr_lag, npm_lag, opmbd_l
       opmad_lag, gpm_lag, roa_lag, roe_lag, roce_lag, debt_at_lag, de_ratio_lag, intcov_lag,
       cash_ratio_lag, quick_ratio_lag, curr_ratio_lag, inv_turn_lag, at_turn_lag, rect_turn_l
       debt_assets_lag, ptb_lag, sale_nwc_lag, divyield_lag,.direction = 'down') %>%
  ungroup()


summary (fin_ratios) # no missing value
```

```
     gvkey           public_date               bm             pe_exi
 Min.   :  1161   Min.   :2016-12-31   Min.   :-0.339   Min.   :-2325.0
 1st Qu.:  8113   1st Qu.:2017-03-24   1st Qu.: 0.182   1st Qu.:    5.5
 Median : 23252   Median :2017-06-16   Median : 0.328   Median :   19.8
 Mean   : 53462   Mean   :2017-06-16   Mean   : 0.412   Mean   :  -13.7
 3rd Qu.:121718   3rd Qu.:2017-09-08   3rd Qu.: 0.603   3rd Qu.:   27.6
 Max.   :184500   Max.   :2017-11-30   Max.   : 2.346   Max.   :  437.6
       ps              pcf                dpr               npm
 Min.   : 0.07   Min.   :-239.75   Min.   : 0.00   Min.   :-5.758
 1st Qu.: 0.98   1st Qu.:   8.05   1st Qu.: 0.00   1st Qu.: 0.015
 Median : 2.10   Median :  13.62   Median : 0.23   Median : 0.064
 Mean   : 2.89   Mean   :  16.86   Mean   : 0.57   Mean   : 0.008
 3rd Qu.: 3.38   3rd Qu.:  20.89   3rd Qu.: 0.51   3rd Qu.: 0.108
 Max.   :38.39   Max.   : 243.78   Max.   :31.92   Max.   : 0.388
```

```
    opmbd            opmad             gpm              roa
 Min.   :-5.363   Min.   :-5.577   Min.   :-4.616   Min.   :-0.509
 1st Qu.: 0.078   1st Qu.: 0.044   1st Qu.: 0.270   1st Qu.: 0.072
 Median : 0.146   Median : 0.111   Median : 0.392   Median : 0.116
 Mean   : 0.123   Mean   : 0.067   Mean   : 0.380   Mean   : 0.132
 3rd Qu.: 0.243   3rd Qu.: 0.190   3rd Qu.: 0.587   3rd Qu.: 0.175
 Max.   : 0.556   Max.   : 0.501   Max.   : 0.912   Max.   : 0.895
      roe                roce            debt_at         de_ratio
 Min.   :-114.00   Min.   :-0.627   Min.   :0.000   Min.   :-141.44
 1st Qu.:   0.02   1st Qu.: 0.052   1st Qu.:0.117   1st Qu.:   0.84
 Median :   0.11   Median : 0.132   Median :0.267   Median :   1.54
 Mean   :  -0.07   Mean   : 0.163   Mean   :0.297   Mean   :   2.39
 3rd Qu.:   0.20   3rd Qu.: 0.207   3rd Qu.:0.396   3rd Qu.:   2.78
 Max.   :  35.65   Max.   : 1.570   Max.   :3.176   Max.   : 146.68
     intcov       cash_ratio      quick_ratio      curr_ratio       inv_turn
 Min.   : -33   Min.   :0.000   Min.   :0.129   Min.   :0.401   Min.   :  0.6
 1st Qu.:   2   1st Qu.:0.216   1st Qu.:0.880   1st Qu.:1.246   1st Qu.:  3.9
 Median :   5   Median :0.498   Median :1.291   Median :1.588   Median :  5.4
 Mean   :  24   Mean   :0.843   Mean   :1.564   Mean   :1.973   Mean   : 26.2
 3rd Qu.:   8   3rd Qu.:1.107   3rd Qu.:1.954   3rd Qu.:2.192   3rd Qu.: 17.1
 Max.   :7083   Max.   :6.328   Max.   :7.076   Max.   :7.286   Max.   :446.3
    at_turn       debt_assets          ptb            rect_turn
 Min.   :0.087   Min.   :0.107   Min.   :-12930   Min.   :  0.14
 1st Qu.:0.473   1st Qu.:0.484   1st Qu.:     1   1st Qu.:  5.32
 Median :0.733   Median :0.637   Median :     3   Median :  7.18
 Mean   :0.976   Mean   :0.657   Mean   :   -18   Mean   : 14.74
 3rd Qu.:1.400   3rd Qu.:0.794   3rd Qu.:     5   3rd Qu.: 12.69
 Max.   :3.670   Max.   :3.722   Max.   :  3399   Max.   :118.88
    sale_nwc          divyield          TICKER               atq
 Min.   : -85.0   Min.   :0.00233   Length:31825       Min.   :     6
 1st Qu.:   1.8   1st Qu.:0.01460   Class :character   1st Qu.:  1698
 Median :   4.0   Median :0.01930   Mode  :character   Median :  5228
 Mean   :  17.6   Mean   :0.02286                      Mean   : 26608
 3rd Qu.:  10.6   3rd Qu.:0.02780                      3rd Qu.: 17630
 Max.   :2500.7   Max.   :0.12100                      Max.   :503073
     bm_lag          pe_exi_lag          ps_lag          pcf_lag
 Min.   :-0.339   Min.   :-2325.0   Min.   : 0.07   Min.   :-226.65
 1st Qu.: 0.182   1st Qu.:     5.8   1st Qu.: 0.98   1st Qu.:   8.11
 Median : 0.328   Median :    19.8   Median : 2.08   Median :  13.66
 Mean   : 0.412   Mean   :   -10.4   Mean   : 2.84   Mean   :  16.65
 3rd Qu.: 0.615   3rd Qu.:    27.8   3rd Qu.: 3.38   3rd Qu.:  20.60
 Max.   : 2.346   Max.   :  432.6   Max.   :38.39   Max.   : 243.78
    dpr_lag          npm_lag          opmbd_lag         opmad_lag
 Min.   : 0.00   Min.   :-5.758   Min.   :-5.363   Min.   :-5.577
 1st Qu.: 0.00   1st Qu.: 0.015   1st Qu.: 0.078   1st Qu.: 0.044
 Median : 0.23   Median : 0.064   Median : 0.145   Median : 0.111
 Mean   : 0.57   Mean   : 0.009   Mean   : 0.123   Mean   : 0.068
 3rd Qu.: 0.50   3rd Qu.: 0.107   3rd Qu.: 0.242   3rd Qu.: 0.190
 Max.   :31.92   Max.   : 0.388   Max.   : 0.556   Max.   : 0.501
    gpm_lag           roa_lag           roe_lag             roce_lag
 Min.   :-4.616   Min.   :-0.509   Min.   :-114.00   Min.   :-0.627
 1st Qu.: 0.270   1st Qu.: 0.071   1st Qu.:   0.02   1st Qu.: 0.058
 Median : 0.392   Median : 0.116   Median :   0.11   Median : 0.132
 Mean   : 0.381   Mean   : 0.133   Mean   :  -0.07   Mean   : 0.164
```

```
 3rd Qu.: 0.587   3rd Qu.: 0.176   3rd Qu.:   0.20   3rd Qu.: 0.207  
 Max.   : 0.912   Max.   : 0.895   Max.   :  35.65   Max.   : 1.570  
  debt_at_lag      de_ratio_lag       intcov_lag    cash_ratio_lag 
 Min.   :0.000   Min.   :-141.44   Min.   : -33   Min.   :0.000  
 1st Qu.:0.112   1st Qu.:   0.83   1st Qu.:   2   1st Qu.:0.212  
 Median :0.268   Median :   1.53   Median :   5   Median :0.498  
 Mean   :0.296   Mean   :   2.30   Mean   :  31   Mean   :0.843  
 3rd Qu.:0.394   3rd Qu.:   2.78   3rd Qu.:   8   3rd Qu.:1.107  
 Max.   :3.161   Max.   : 134.26   Max.   :7083   Max.   :6.328  
 quick_ratio_lag curr_ratio_lag   inv_turn_lag    at_turn_lag   
 Min.   :0.129   Min.   :0.401   Min.   :  0.6   Min.   :0.087  
 1st Qu.:0.886   1st Qu.:1.248   1st Qu.:  3.9   1st Qu.:0.473  
 Median :1.291   Median :1.606   Median :  5.4   Median :0.733  
 Mean   :1.569   Mean   :1.977   Mean   : 27.4   Mean   :0.979  
 3rd Qu.:1.954   3rd Qu.:2.192   3rd Qu.: 17.1   3rd Qu.:1.409  
 Max.   :7.076   Max.   :7.286   Max.   :446.3   Max.   :3.670  
 rect_turn_lag     sale_nwc_lag      divyield_lag      debt_assets_lag
 Min.   :  0.14   Min.   : -85.0   Min.   :0.00227   Min.   :0.107  
 1st Qu.:  5.32   1st Qu.:   1.8   1st Qu.:0.01480   1st Qu.:0.484  
 Median :  7.18   Median :   4.0   Median :0.01920   Median :0.638  
 Mean   : 14.74   Mean   :  17.3   Mean   :0.02275   Mean   :0.657  
 3rd Qu.: 12.72   3rd Qu.:  10.6   3rd Qu.:0.02770   3rd Qu.:0.794  
 Max.   :114.23   Max.   :2500.7   Max.   :0.12100   Max.   :3.722  
    ptb_lag      
 Min.   :-12930  
 1st Qu.:     1  
 Median :     3  
 Mean   :   -20  
 3rd Qu.:     5  
 Max.   :  3399  
```

Finally, we left join between two datasets: `stock_announce` and `fin_ratios`, and we named the merged dataset as `stock_announce_financials`. The `summary` fuction reveals 190 NA values for financial data for 30 December and 31 December 2016.

```
# merge into the full dataset
# no financial data on 29/12/2016 and 30/12/2016
stock_announce_financials <- left_join(x = stock_announce, y = fin_ratios,
                                       by = c("gvkey" = "gvkey", "datadate" = "public_date"))
summary(stock_announce_financials) # na 190 for financial data on 2016/12/30 and 2016/12/31
```

```
     gvkey          kd_date               cik             datadate         
 Min.   :  1161   Length:26315       Min.   :   2488   Min.   :2016-12-29  
 1st Qu.:  8113   Class :character   1st Qu.: 101829   1st Qu.:2017-03-08  
 Median : 23252   Mode  :character   Median : 885306   Median :2017-05-16  
 Mean   : 53462                      Mean   : 754029   Mean   :2017-05-16  
 3rd Qu.:121718                      3rd Qu.:1090872   3rd Qu.:2017-07-24  
 Max.   :184500                      Max.   :1707925   Max.   :2017-10-01  
                                     NA's   :2770                          
     iid                conm               prccd            prchd      
 Length:26315       Length:26315       Min.   :   0.0   Min.   :   0.0  
 Class :character   Class :character   1st Qu.:  22.3   1st Qu.:  22.7  
 Mode  :character   Mode  :character   Median :  52.1   Median :  52.5  
                                       Mean   :  82.7   Mean   :  83.4  
```

```
                                      3rd Qu.:  94.9   3rd Qu.:  95.8  
                                      Max.   :1046.4   Max.   :1054.8  
                                                                       
     prcld            prcod              sic       p_daily_change   
 Min.   :   0.0   Min.   :   0.0   Min.   :1311   Min.   :-0.3924  
 1st Qu.:  21.9   1st Qu.:  22.3   1st Qu.:3576   1st Qu.:-0.0048  
 Median :  51.4   Median :  51.8   Median :4841   Median : 0.0000  
 Mean   :  82.0   Mean   :  82.6   Mean   :4821   Mean   : 0.0016  
 3rd Qu.:  93.8   3rd Qu.:  94.9   3rd Qu.:6141   3rd Qu.: 0.0060  
 Max.   :1041.8   Max.   :1048.2   Max.   :8742   Max.   : 0.6198  
                                   NA's   :2770                    
    volatile      p_daily_change_lag  volatile_lag    kd_type         
 Min.   :0.0000   Min.   :-0.39      Min.   :0.00   Length:26315      
 1st Qu.:0.0093   1st Qu.: 0.00      1st Qu.:0.01   Class :character  
 Median :0.0145   Median : 0.00      Median :0.01   Mode  :character  
 Mean   :0.0200   Mean   : 0.00      Mean   :0.02                     
 3rd Qu.:0.0232   3rd Qu.: 0.01      3rd Qu.:0.02                     
 Max.   :0.7460   Max.   : 0.62      Max.   :0.75                     
                  NA's   :92         NA's   :92                       
  kd_company         kd_headline          is_annouce      red_annouce    
 Length:26315       Length:26315       Min.   :0.0000   Min.   :0.0000  
 Class :character   Class :character   1st Qu.:0.0000   1st Qu.:0.0000  
 Mode  :character   Mode  :character   Median :0.0000   Median :0.0000  
                                       Mean   :0.0568   Mean   :0.0043  
                                       3rd Qu.:0.0000   3rd Qu.:0.0000  
                                       Max.   :1.0000   Max.   :1.0000  
                                                                        
 no_annoucement        bm            pe_exi              ps        
 Min.   : 0.00   Min.   :-0.34   Min.   :-2325.0   Min.   : 0.07  
 1st Qu.: 0.00   1st Qu.: 0.18   1st Qu.:    6.1   1st Qu.: 0.98  
 Median : 0.00   Median : 0.32   Median :   20.1   Median : 2.08  
 Mean   : 0.11   Mean   : 0.41   Mean   :   -4.0   Mean   : 2.84  
 3rd Qu.: 0.00   3rd Qu.: 0.62   3rd Qu.:   27.9   3rd Qu.: 3.39  
 Max.   :46.00   Max.   : 2.35   Max.   :  432.6   Max.   :38.39  
                 NA's   :190     NA's   :190       NA's   :190    
      pcf              dpr              npm             opmbd      
 Min.   :-226.65   Min.   : 0.00   Min.   :-5.76   Min.   :-5.36  
 1st Qu.:   8.12   1st Qu.: 0.00   1st Qu.: 0.01   1st Qu.: 0.08  
 Median :  13.79   Median : 0.23   Median : 0.06   Median : 0.15  
 Mean   :  17.54   Mean   : 0.60   Mean   : 0.01   Mean   : 0.12  
 3rd Qu.:  20.71   3rd Qu.: 0.50   3rd Qu.: 0.11   3rd Qu.: 0.24  
 Max.   : 243.78   Max.   :31.92   Max.   : 0.39   Max.   : 0.56  
 NA's   :190       NA's   :190     NA's   :190     NA's   :190    
     opmad            gpm             roa             roe         
 Min.   :-5.58   Min.   :-4.62   Min.   :-0.51   Min.   :-114.00  
 1st Qu.: 0.04   1st Qu.: 0.27   1st Qu.: 0.07   1st Qu.:   0.02  
 Median : 0.11   Median : 0.39   Median : 0.12   Median :   0.11  
 Mean   : 0.07   Mean   : 0.38   Mean   : 0.13   Mean   :  -0.12  
 3rd Qu.: 0.19   3rd Qu.: 0.59   3rd Qu.: 0.18   3rd Qu.:   0.20  
 Max.   : 0.50   Max.   : 0.91   Max.   : 0.90   Max.   :  35.65  
 NA's   :190     NA's   :190     NA's   :190     NA's   :190      
      roce           debt_at        de_ratio            intcov        cash_ratio   
 Min.   :-0.63   Min.   :0.00   Min.   :-141.44   Min.   : -33   Min.   :0.00  
 1st Qu.: 0.06   1st Qu.:0.11   1st Qu.:   0.83   1st Qu.:   2   1st Qu.:0.22  
```

```
 Median : 0.13   Median :0.27   Median :   1.54   Median :   5   Median :0.51
 Mean   : 0.16   Mean   :0.30   Mean   :   2.23   Mean   :  27   Mean   :0.84
 3rd Qu.: 0.21   3rd Qu.:0.39   3rd Qu.:   2.78   3rd Qu.:   8   3rd Qu.:1.11
 Max.   : 1.57   Max.   :3.16   Max.   : 134.26   Max.   :7083   Max.   :6.33
 NA's   :190     NA's   :190    NA's   :190       NA's   :190    NA's   :190
  quick_ratio     curr_ratio      inv_turn        at_turn      debt_assets
 Min.   :0.13   Min.   :0.40   Min.   :  0.6   Min.   :0.09   Min.   :0.11
 1st Qu.:0.89   1st Qu.:1.25   1st Qu.:  3.8   1st Qu.:0.48   1st Qu.:0.48
 Median :1.29   Median :1.61   Median :  5.4   Median :0.73   Median :0.64
 Mean   :1.57   Mean   :1.98   Mean   : 26.6   Mean   :0.98   Mean   :0.66
 3rd Qu.:1.95   3rd Qu.:2.19   3rd Qu.: 17.1   3rd Qu.:1.41   3rd Qu.:0.80
 Max.   :7.08   Max.   :7.29   Max.   :446.3   Max.   :3.67   Max.   :3.72
 NA's   :190    NA's   :190    NA's   :190     NA's   :190    NA's   :190
      ptb           rect_turn         sale_nwc         divyield
 Min.   :-12930   Min.   :  0.14   Min.   : -85.0   Min.   :0.00
 1st Qu.:     1   1st Qu.:  5.32   1st Qu.:   1.8   1st Qu.:0.01
 Median :     3   Median :  7.17   Median :   4.0   Median :0.02
 Mean   :   -24   Mean   : 14.69   Mean   :  18.6   Mean   :0.02
 3rd Qu.:     5   3rd Qu.: 12.76   3rd Qu.:  10.7   3rd Qu.:0.03
 Max.   :  3399   Max.   :114.23   Max.   :2500.7   Max.   :0.12
 NA's   :190      NA's   :190      NA's   :190      NA's   :190
    TICKER               atq             bm_lag        pe_exi_lag
 Length:26315       Min.   :     6   Min.   :-0.27   Min.   :-2260.0
 Class :character   1st Qu.:  1551   1st Qu.: 0.18   1st Qu.:    6.6
 Mode  :character   Median :  4976   Median : 0.33   Median :   20.0
                    Mean   : 26197   Mean   : 0.41   Mean   :   -0.6
                    3rd Qu.: 17471   3rd Qu.: 0.62   3rd Qu.:   28.1
                    Max.   :503073   Max.   : 2.35   Max.   :  432.6
                    NA's   :190      NA's   :190     NA's   :190
     ps_lag         pcf_lag          dpr_lag         npm_lag
 Min.   : 0.07   Min.   :-166.46   Min.   : 0.00   Min.   :-5.76
 1st Qu.: 0.98   1st Qu.:   8.27   1st Qu.: 0.00   1st Qu.: 0.02
 Median : 2.08   Median :  13.79   Median : 0.23   Median : 0.06
 Mean   : 2.80   Mean   :  17.00   Mean   : 0.61   Mean   : 0.01
 3rd Qu.: 3.39   3rd Qu.:  20.38   3rd Qu.: 0.49   3rd Qu.: 0.11
 Max.   :32.19   Max.   : 243.78   Max.   :31.92   Max.   : 0.39
 NA's   :190     NA's   :190       NA's   :190     NA's   :190
   opmbd_lag       opmad_lag        gpm_lag         roa_lag
 Min.   :-5.32   Min.   :-5.56   Min.   :-4.62   Min.   :-0.51
 1st Qu.: 0.08   1st Qu.: 0.04   1st Qu.: 0.27   1st Qu.: 0.07
 Median : 0.14   Median : 0.11   Median : 0.40   Median : 0.12
 Mean   : 0.12   Mean   : 0.07   Mean   : 0.38   Mean   : 0.13
 3rd Qu.: 0.24   3rd Qu.: 0.19   3rd Qu.: 0.59   3rd Qu.: 0.18
 Max.   : 0.56   Max.   : 0.50   Max.   : 0.91   Max.   : 0.90
 NA's   :190     NA's   :190     NA's   :190     NA's   :190
    roe_lag           roce_lag      debt_at_lag    de_ratio_lag
 Min.   :-114.00   Min.   :-0.63   Min.   :0.00   Min.   :-141.44
 1st Qu.:   0.02   1st Qu.: 0.06   1st Qu.:0.11   1st Qu.:   0.83
 Median :   0.11   Median : 0.13   Median :0.27   Median :   1.53
 Mean   :  -0.14   Mean   : 0.17   Mean   :0.30   Mean   :   2.13
 3rd Qu.:   0.20   3rd Qu.: 0.21   3rd Qu.:0.39   3rd Qu.:   2.78
 Max.   :  35.65   Max.   : 1.57   Max.   :3.16   Max.   : 134.26
 NA's   :190       NA's   :190     NA's   :190    NA's   :190
   intcov_lag   cash_ratio_lag quick_ratio_lag curr_ratio_lag  inv_turn_lag
```

```
 Min.   : -33   Min.   :0.00   Min.   :0.13    Min.   :0.40   Min.   :  0.6  
 1st Qu.:   2   1st Qu.:0.21   1st Qu.:0.90    1st Qu.:1.25   1st Qu.:  3.9  
 Median :   5   Median :0.51   Median :1.29    Median :1.61   Median :  5.4  
 Mean   :  36   Mean   :0.84   Mean   :1.57    Mean   :1.98   Mean   : 28.1  
 3rd Qu.:   9   3rd Qu.:1.10   3rd Qu.:1.95    3rd Qu.:2.19   3rd Qu.: 17.1  
 Max.   :7083   Max.   :6.33   Max.   :7.08    Max.   :7.29   Max.   :446.3  
 NA's   :190    NA's   :190    NA's   :190     NA's   :190    NA's   :190    
  at_turn_lag   rect_turn_lag     sale_nwc_lag     divyield_lag  
 Min.   :0.09   Min.   :  0.14   Min.   : -85.0   Min.   :0.00  
 1st Qu.:0.48   1st Qu.:  5.30   1st Qu.:   1.8   1st Qu.:0.01  
 Median :0.73   Median :  7.01   Median :   4.0   Median :0.02  
 Mean   :0.98   Mean   : 14.72   Mean   :  18.3   Mean   :0.02  
 3rd Qu.:1.41   3rd Qu.: 12.76   3rd Qu.:  10.6   3rd Qu.:0.03  
 Max.   :3.67   Max.   :114.23   Max.   :2500.7   Max.   :0.12  
 NA's   :190    NA's   :190      NA's   :190      NA's   :190   
 debt_assets_lag    ptb_lag      
 Min.   :0.11     Min.   :-12930  
 1st Qu.:0.49     1st Qu.:     1  
 Median :0.64     Median :     3  
 Mean   :0.66     Mean   :   -26  
 3rd Qu.:0.79     3rd Qu.:     5  
 Max.   :3.72     Max.   :  3399  
 NA's   :190      NA's   :190     
```

3.7 : Processing Twitter profile data

We load two datasets which are profile and com_info respectively. After selecting specific columns from com_info, we left join profile and com_info based on the matching columns “user_name” and “username” and named it as “profile”.

```
library(stringr)
#load the company information data
profile <- read.csv("profile_info.csv")
com_info <- read_excel("company information.xlsx")
com_info <- select(com_info, gvkey, conm, username)

# merge profile to gvkey with username
profile <- left_join(profile, com_info, by = c("user_name" = "username"))
```

We create a function named `convert_to_numeric` designed to transform string representations of numbers into numeric format, converting abbreviations like “K” for thousands and “M” for millions while handling commas. The function is then applied to the “profile_followers” and “profile_following” columns in the `profile` dataset to compute the Twitter Follower to Following Ratio (TFF). The summary of TFF reveals occurrences of infinite (INF) values, which denote companies with either no followers and zero following or those that have been acquired or are bankrupt.

```
#function to convert string representation of numbers into numeric format
convert_to_numeric <- function(x) {
  # Convert K (thousands) and M (millions) to numeric values
  x_numeric <- gsub("K", "e3", x)  # Convert K to 10^3
  x_numeric <- gsub("M", "e6", x_numeric)  # Convert M to 10^6
```

```
  # Remove any commas and convert to numeric
  as.numeric(gsub(",", "", x_numeric))
}


#apply the function to the profile_followers/profile_following column
profile$profile_followers_num <- convert_to_numeric(profile$profile_followers)
profile$profile_following_num <- convert_to_numeric(profile$profile_following)

#compute twitter follower to following ratio (TFF)
profile <- profile %>% mutate (TFF = profile_followers_num/ profile_following_num)
summary (profile$TFF) #TFF has INF values
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    0.1     7.0    36.9     Inf   335.9     Inf 
```

Next, we adjust the Twitter Follower to Following Ratio (TFF) for companies with INF values by modifying the profile_following_num column. It replaces instances where profile_following_num equals zero with one to prevent division by zero, thereby ensuring numeric stability. Following this adjustment, the TFF is recalculated. The summary of TFF is examined to verify that there are no more occurrences of INF values, indicating successful handling of the issue.

```
#handle TFF=INF companies by changing profile_following_num = 0 to profile_following_num = 1
profile <- profile%>%
  mutate(profile_following_num = ifelse(profile_following_num == 0, 1, profile_following_num))
  mutate (TFF = profile_followers_num/ profile_following_num)

#examine TFF ratio should have no more INF
summary (profile$TFF)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
      0       5      31    1690     218   77778 
```

We calculate readability indices (Flesch score, Coleman-Liau, and FOG) for the profile biographies of individuals or entities. Using functions from the quanteda.textstats package, the code computes these indices and assigns them to corresponding columns in the profile dataframe. Additionally, the code counts the number of words in each biography using regular expressions and stores the counts in the num_words column. NA values in the word count column are replaced with 0 to indicate profiles with no biography.

```
pacman :: p_load(quanteda.textstats)
coleman_info <- textstat_readability(profile$profile_bio, "Coleman.Liau")
flesch_info <- textstat_readability(profile$profile_bio, "Flesch")
FOG_info <- textstat_readability(profile$profile_bio, "FOG")

profile$coleman_liau <- coleman_info$Coleman.Liau.ECP
profile$flesch <- flesch_info$Flesch
profile$fog_info <- FOG_info$FOG

#load the stock price data to get the sic
sic <- read.csv("sic.csv")
sic <- sic %>% select(gvkey, sic) %>% distinct(gvkey, sic)
```

```r
#left join to profile df to get profile by sic
profile <- left_join(profile, sic)

#compute industry median for readability to fill in NA later
profile <- profile %>% group_by(sic) %>%
  mutate(im_flesch = median(flesch,na.rm = TRUE)) %>%
  mutate(im_coleman = median(coleman_liau,na.rm = TRUE)) %>%
  mutate(im_FOG = median(fog_info,na.rm = TRUE)) %>%
  ungroup()

#compute median for all firms to fill in NA for those firm which is the only firm present in s
profile <- profile %>%
  mutate(im_total_flesch = median(flesch,na.rm = TRUE)) %>%
  mutate(im_total_coleman = median(coleman_liau,na.rm = TRUE)) %>%
  mutate(im_total_FOG = median(fog_info,na.rm = TRUE))

#fill in NA for readability using industry median.
profile <- profile %>%
  mutate(flesch = ifelse(is.na(flesch), im_flesch, flesch)) %>%
  mutate(coleman_liau = ifelse(is.na(coleman_liau), im_coleman, coleman_liau)) %>%
  mutate(fog_info = ifelse(is.na(fog_info), im_FOG, fog_info))

#if the only firm present in sic is the NA firm, replace NA with median of all firm
profile <- profile %>%
  mutate(flesch = ifelse(is.na(flesch), im_total_flesch, flesch)) %>%
  mutate(coleman_liau = ifelse(is.na(coleman_liau), im_total_coleman, coleman_liau)) %>%
  mutate(fog_info = ifelse(is.na(fog_info), im_total_FOG, fog_info))

summary(profile$coleman_liau)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-104.9    17.4    31.6    31.0    47.1   138.0 
```

```r
summary(profile$flesch)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
-122.6    11.1    33.3    34.4    56.5    98.8 
```

```r
summary(profile$fog_info)
```

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3.7    10.0    14.0    14.8    18.1    34.8 
```

```r
str(profile$profile_bio) #character vector containing bio sentence
```

```
chr [1:89] "together we advance_" ...
```

```r
# Use str_count to count the number of words in each sentence
#"\S+" matches one or more non-whitespace characters in the string.
profile$num_words <- str_count(profile$profile_bio, "\\S+")
```

```
#NA means no profile bio, hence replace with 0 since no word count
profile$num_words <- ifelse(is.na(profile$num_words),0,profile$num_words)
str(profile$num_words)
```

```
num [1:89] 3 15 19 0 8 6 4 9 21 8 ...
```

```
summary (profile$num_words)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    0.0    11.0    18.0    15.5    21.0    28.0
```

We evaluate the completeness of user profiles, aiming to enhance search visibility. It assesses whether essential profile fields like name, category, website, bio, and username are populated or not. For each profile, it assigns a binary indicator, with '1' indicating a complete profile and '0' indicating an incomplete one. Additionally, it identifies profiles with URLs by checking if the website field is empty, assigning '1' for profiles with URLs and '0' otherwise.

```
#profile completeness increase the chance of people searching for profile.
#check if profile data is complete
profile$bio_complete <- ifelse(is.na(profile$profile_name) |is.na(profile$profile_category) |i
                                 is.na(profile$profile_bio) |is.na(profile$user_name), 0, 1)

profile$has_url <- ifelse(is.na(profile$profile_website), 0, 1)
```

We create two variables ("hashtags_count", "has_hashtags")

hashtags_count: The frequency of hashtags in the profile bio of the company's account

has_hashtags: The profile bio of the company's account contains hashtags or not

```
#compute the no of hashtags the profile bio has
#function to count hashtags in each profile bio
count_hashtags <- function(text) {
  # Use str_count from stringr package to count occurrences of '#'
  hashtags_count <- stringr::str_count(text, "#")
  return(hashtags_count)
}

#apply the function to each profile bio to compute the no of hashtags
profile$hashtags_count <- sapply(profile$profile_bio, count_hashtags)

#NA means no profile bio, hence replace with 0 since no hashtag
profile$hashtags_count <- ifelse(is.na(profile$hashtags_count),0,profile$hashtags_count)
summary (profile$hashtags_count)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.000   0.000   0.000   0.393   0.000  11.000
```

```
#compute has hashtags variable (has hastag = 1, no hashtag = 0)
profile$has_hashtags <- ifelse(profile$hashtags_count == 0, 0, 1)
```

We create three dummy variables (“Contains Company Name”, “Contains Words”, “Custom Content”) based on the ARXIV article’s classification of user names.

Contains Name: This group contains users that have a given name in their name field that we could match to the Behind the Name data.

Contains Words: This group contains all users that are not in the first group, but have at least one English word from the SIL list in their name field.

Custom Content: This group contains all users that are neither in the first nor in the second group.

It defines a function to categorize user names according to their similarity to company names or the presence of English words. Then, it applies this function to the user_name column and further adjusts certain specific user names to ensure accurate classification. Finally, it generates dummy variables to indicate each user’s category, allowing for further analysis of profile completeness.

```
#load the qdapDictionaries package
pacman :: p_load(qdapDictionaries)

#load the English words list
data("words")

#assign the words vector to english_words
english_words <- words

#function to categorize user_name based on conm
categorize_user_name <- function(name, conm) {
  if (grepl(tolower(conm), tolower(name))) {
    return("Contains Company Name")
  } else if (any(grepl(english_words, name))) {
    return("Contains Words")
  } else {
    return("Custom Content")
  }
}

#apply function to user_name column
profile$user_name_category <- mapply(categorize_user_name, profile$user_name, profile$conm)

profile <- profile %>% mutate(user_name_category = ifelse(user_name=="@AmericanExpress"|user_n
                                                          |user_name == "@autodesk"|user_name 
                                                          |user_name == "@AvonInsider"|user_na
                                                          |user_name == "@DXCTechnology"|user_
                                                          |user_name == "@DPL_INC"|user_name =
                                                          |user_name == "@Ford"|user_name == "
                                                          |user_name == "@L3HarrisTech"|user_n
                                                          |user_name == "@Range_Resources"|use
                                                          |user_name == "@SPGlobal"|user_name 
                                                          |user_name == "@ParkerHannifin"|user
                                                          |user_name == "@Pentair"|user_name =
                                                          |user_name == "@SemtechCorp"|user_na
```

```
                                                    |user_name == "@Sysco"|user_name ==
                                                    |user_name == "@footlocker"|user_nam
                                                    |user_name == "@Gartner_inc"|user_na
                                                    |user_name == "@mercury"|user_name =
                                                    |user_name == "@biogen"|user_name ==
                                                    |user_name == "@Lindeplc"|user_name
                                                    |user_name == "@Wolfspeed"|user_name
                                                    |user_name == "@Chubb"|user_name ==
                                                    |user_name == "@byodgaming"|user_nam
                                                    |user_name == "@celandongroup"|user_
                                                    |user_name == "@JosAbank"|user_name
                                                    |user_name == "@mettlertoledo"|user_
                                                    |user_name == "@cebinc"|user_name ==
                                                    |user_name == "@RedHat"|user_name ==
                                                    |user_name == "@dominos"|user_name =
                                                    |user_name == "@cynosureinc"|user_na
                                                    |user_name == "@ConstantContact"|use
                                                    |user_name == "@MYRGroupInc",
                                                   "Contains Company Name", user_name_c
#examine user_name category
user_name_category_examine <- profile %>% select(user_name, user_name_category, conm)


#create 3 dummy variables
profile <- profile %>% mutate(contains_company_name = ifelse(user_name_category == "Contains C
  mutate(contains_words = ifelse(user_name_category == "Contains Words",1,0)) %>%
  mutate(custom_content = ifelse(user_name_category == "Custom Content",1,0))

summary(profile)
```

```
profile_name       profile_bio        profile_category   profile_website
Length:89          Length:89          Length:89          Length:89
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character



profile_joining_date profile_following  profile_followers   user_name
Length:89            Length:89          Length:89          Length:89
Class :character     Class :character   Class :character   Class :character
Mode  :character     Mode  :character   Mode  :character   Mode  :character



    gvkey            conm           profile_followers_num
Min.   :  1161   Length:89          Min.   :      1
1st Qu.:  7163   Class :character   1st Qu.:   4398
Median : 23252   Mode  :character   Median :  22000
Mean   : 52627                      Mean   : 188033
3rd Qu.:118502                      3rd Qu.: 118600
Max.   :184500                      Max.   :3800000
```

```
 profile_following_num      TFF          coleman_liau       flesch
 Min.   :    1          Min.   :    0   Min.   :-104.9   Min.   :-122.6
 1st Qu.:   99          1st Qu.:    5   1st Qu.:  17.4   1st Qu.:  11.1
 Median :  510          Median :   31   Median :  31.6   Median :  33.3
 Mean   : 1863          Mean   : 1690   Mean   :  31.0   Mean   :  34.4
 3rd Qu.: 1091          3rd Qu.:  218   3rd Qu.:  47.1   3rd Qu.:  56.5
 Max.   :29500          Max.   :77778   Max.   : 138.0   Max.   :  98.8

    fog_info         sic          im_flesch         im_coleman         im_FOG
 Min.   : 3.7   Min.   :1311   Min.   :-122.6   Min.   :-104.9   Min.   : 3.7
 1st Qu.:10.0   1st Qu.:3576   1st Qu.:  12.9   1st Qu.:  22.5   1st Qu.:11.1
 Median :14.0   Median :4931   Median :  33.3   Median :  31.6   Median :14.0
 Mean   :14.8   Mean   :4877   Mean   :  34.2   Mean   :  32.0   Mean   :15.0
 3rd Qu.:18.1   3rd Qu.:6211   3rd Qu.:  55.3   3rd Qu.:  48.4   3rd Qu.:18.6
 Max.   :34.8   Max.   :8742   Max.   :  98.8   Max.   : 138.0   Max.   :34.8
                               NA's   :4        NA's   :4        NA's   :4
 im_total_flesch im_total_coleman  im_total_FOG   num_words     bio_complete
 Min.   :33.3    Min.   :31.6     Min.   :14    Min.   : 0.0   Min.   :0.000
 1st Qu.:33.3    1st Qu.:31.6     1st Qu.:14    1st Qu.:11.0   1st Qu.:0.000
 Median :33.3    Median :31.6     Median :14    Median :18.0   Median :0.000
 Mean   :33.3    Mean   :31.6     Mean   :14    Mean   :15.5   Mean   :0.169
 3rd Qu.:33.3    3rd Qu.:31.6     3rd Qu.:14    3rd Qu.:21.0   3rd Qu.:0.000
 Max.   :33.3    Max.   :31.6     Max.   :14    Max.   :28.0   Max.   :1.000

    has_url      hashtags_count    has_hashtags   user_name_category
 Min.   :0.000   Min.   : 0.000   Min.   :0.000   Length:89
 1st Qu.:1.000   1st Qu.: 0.000   1st Qu.:0.000   Class :character
 Median :1.000   Median : 0.000   Median :0.000   Mode  :character
 Mean   :0.899   Mean   : 0.393   Mean   :0.169
 3rd Qu.:1.000   3rd Qu.: 0.000   3rd Qu.:0.000
 Max.   :1.000   Max.   :11.000   Max.   :1.000

 contains_company_name contains_words   custom_content
 Min.   :0.000         Min.   :0.000   Min.   :0.00
 1st Qu.:0.000         1st Qu.:0.000   1st Qu.:0.00
 Median :1.000         Median :0.000   Median :0.00
 Mean   :0.618         Mean   :0.202   Mean   :0.18
 3rd Qu.:1.000         3rd Qu.:0.000   3rd Qu.:0.00
 Max.   :1.000         Max.   :1.000   Max.   :1.00
```

We generate a sequence of dates from 29 December 2016 to 1 October 2017. It then performs a cross join between the date sequence and unique gvkeys to create a full combination of gvkeys and dates. Afterward, it merges the profile information with the full combination based on gvkey using a left join, resulting in a dataset that includes profile data for each date within the specified period.

```r
# select need variables
profile <- profile %>% select(-im_flesch, -im_coleman, -im_FOG, -im_total_coleman, -im_total_f
summary(profile)
```

```
 profile_name        profile_bio         profile_category    profile_website    
 Length:89           Length:89           Length:89           Length:89          
 Class :character    Class :character    Class :character    Class :character   
 Mode  :character    Mode  :character    Mode  :character    Mode  :character   
                                                                                
                                                                                
                                                                                
 profile_joining_date profile_following  profile_followers    user_name        
 Length:89            Length:89          Length:89          Length:89          
 Class :character     Class :character   Class :character   Class :character   
 Mode  :character     Mode  :character   Mode  :character   Mode  :character   
                                                                               
                                                                               
                                                                               
     gvkey            conm           profile_followers_num
 Min.   :  1161   Length:89          Min.   :      1      
 1st Qu.:  7163   Class :character   1st Qu.:   4398      
 Median : 23252   Mode  :character   Median :  22000      
 Mean   : 52627                      Mean   : 188033      
 3rd Qu.:118502                      3rd Qu.: 118600      
 Max.   :184500                      Max.   :3800000      
 profile_following_num      TFF          coleman_liau        flesch      
 Min.   :    1         Min.   :    0   Min.   :-104.9   Min.   :-122.6  
 1st Qu.:   99         1st Qu.:    5   1st Qu.:  17.4   1st Qu.:  11.1  
 Median :  510         Median :   31   Median :  31.6   Median :  33.3  
 Mean   : 1863         Mean   : 1690   Mean   :  31.0   Mean   :  34.4  
 3rd Qu.: 1091         3rd Qu.:  218   3rd Qu.:  47.1   3rd Qu.:  56.5  
 Max.   :29500         Max.   :77778   Max.   : 138.0   Max.   :  98.8  
    fog_info          sic          num_words      bio_complete      has_url     
 Min.   : 3.7   Min.   :1311   Min.   : 0.0   Min.   :0.000   Min.   :0.000  
 1st Qu.:10.0   1st Qu.:3576   1st Qu.:11.0   1st Qu.:0.000   1st Qu.:1.000  
 Median :14.0   Median :4931   Median :18.0   Median :0.000   Median :1.000  
 Mean   :14.8   Mean   :4877   Mean   :15.5   Mean   :0.169   Mean   :0.899  
 3rd Qu.:18.1   3rd Qu.:6211   3rd Qu.:21.0   3rd Qu.:0.000   3rd Qu.:1.000  
 Max.   :34.8   Max.   :8742   Max.   :28.0   Max.   :1.000   Max.   :1.000  
 hashtags_count     has_hashtags    user_name_category contains_company_name
 Min.   : 0.000   Min.   :0.000   Length:89          Min.   :0.000        
 1st Qu.: 0.000   1st Qu.:0.000   Class :character   1st Qu.:0.000        
 Median : 0.000   Median :0.000   Mode  :character   Median :1.000        
 Mean   : 0.393   Mean   :0.169                      Mean   :0.618        
 3rd Qu.: 0.000   3rd Qu.:0.000                      3rd Qu.:1.000        
 Max.   :11.000   Max.   :1.000                      Max.   :1.000        
 contains_words  custom_content
 Min.   :0.000   Min.   :0.00  
 1st Qu.:0.000   1st Qu.:0.00  
 Median :0.000   Median :0.00  
 Mean   :0.202   Mean   :0.18  
 3rd Qu.:0.000   3rd Qu.:0.00  
 Max.   :1.000   Max.   :1.00  
```

```
# Create a sequence of date for the period
date_df <- data.frame(date = seq(from = ymd("2016-12-29"), to = ymd("2017-10-01"), by = "1 day

# crossjoin the date_df and unique_gvkey to get the full combination of gvkey and date
```

```
unique_gvkey <- as.data.frame(unique_gvkey)
full_gvkey_date <- cross_join(unique_gvkey, date_df)
colnames(full_gvkey_date) <- c("gvkey", "datadate")

# left join profile info to full_gvkey_date
profile_data <- left_join(x = full_gvkey_date, y = profile, by = "gvkey")
```

We calculates the number of days, years, and months a company has been active on Twitter from its joining date. It first converts the joining date into a standard format and then computes the number of days and years by calculating the difference between the joining date and the date of the data record. The number of months joined is calculated using the interval function. The analysis reveals 1662 NA values for the six companies that don't have Twitter accounts.

```
str(profile_data$profile_joining_date)
```

```
chr [1:26315] "Joined May 2008" "Joined May 2008" "Joined May 2008" ...
```

```
#extract month and year and remove "Joined"
month_year <- gsub("Joined ", "", profile_data$profile_joining_date)

#convert month_year to date format with default date = "01"
profile_data$joining_date <- as.Date(paste(month_year, "01"), format = "%B %Y %d")

#format the date as YYYY-MM-DD
profile_data$joining_date <- format(profile_data$joining_date, "%Y-%m-%d")
class(profile_data$joining_date) #character
```

```
[1] "character"
```

```
#convert joining_date to Date format
profile_data$joining_date <- as.Date(profile_data$joining_date)
class(profile_data$joining_date) #date
```

```
[1] "Date"
```

```
#compute the number of days between joining_date and datadate
profile_data$days_joined <- as.numeric(difftime(profile_data$datadate, profile_data$joining_da
class(profile_data$days_joined) #numeric
```

```
[1] "numeric"
```

```
#compute the number of years between joining_date and datadate
#dividing by 365.25 instead of 365 accounts for leap years.
#a year has approximately 365.25 days on average due to the occurrence of leap years,
profile_data$years_joined <- as.numeric(difftime(profile_data$datadate, profile_data$joining_d
str(profile_data$years_joined)
```

```
num [1:26315] 8.66 8.67 8.67 8.67 8.67 ...
```

```
#compute no of months joined
profile_data$months_joined <- as.numeric(interval(profile_data$joining_date, profile_data$data
class(profile_data$months_joined) #numeric
```

```
[1] "numeric"
```

```
# 1662 NA for the six companies that don't have accounts
summary(profile_data)
```

```
     gvkey           datadate           profile_name       profile_bio       
 Min.   :  1161   Min.   :2016-12-29   Length:26315       Length:26315      
 1st Qu.:  8113   1st Qu.:2017-03-08   Class :character   Class :character  
 Median : 23252   Median :2017-05-16   Mode  :character   Mode  :character  
 Mean   : 53462   Mean   :2017-05-16                                        
 3rd Qu.:121718   3rd Qu.:2017-07-24                                        
 Max.   :184500   Max.   :2017-10-01                                        

 profile_category   profile_website    profile_joining_date profile_following 
 Length:26315       Length:26315       Length:26315         Length:26315      
 Class :character   Class :character   Class :character     Class :character  
 Mode  :character   Mode  :character   Mode  :character     Mode  :character  




 profile_followers   user_name             conm           profile_followers_num
 Length:26315       Length:26315       Length:26315       Min.   :      1      
 Class :character   Class :character   Class :character   1st Qu.:   4398      
 Mode  :character   Mode  :character   Mode  :character   Median :  22000      
                                                          Mean   : 188033      
                                                          3rd Qu.: 118600      
                                                          Max.   :3800000      
                                                          NA's   :1662         
 profile_following_num      TFF           coleman_liau         flesch      
 Min.   :    1         Min.   :    0   Min.   :-104.9   Min.   :-122.6  
 1st Qu.:   99         1st Qu.:    5   1st Qu.:  17.4   1st Qu.:  11.1  
 Median :  510         Median :   31   Median :  31.6   Median :  33.3  
 Mean   : 1863         Mean   : 1690   Mean   :  31.0   Mean   :  34.3  
 3rd Qu.: 1091         3rd Qu.:  218   3rd Qu.:  47.1   3rd Qu.:  56.5  
 Max.   :29500         Max.   :77778   Max.   : 138.0   Max.   :  98.8  
 NA's   :1662          NA's   :1662    NA's   :1662     NA's   :1662    
    fog_info          sic          num_words      bio_complete      has_url     
 Min.   : 3.7   Min.   :1311   Min.   : 0.0   Min.   :0.0    Min.   :0.0    
 1st Qu.:10.0   1st Qu.:3576   1st Qu.:11.0   1st Qu.:0.0    1st Qu.:1.0    
 Median :14.0   Median :4931   Median :18.0   Median :0.0    Median :1.0    
 Mean   :14.8   Mean   :4877   Mean   :15.5   Mean   :0.2    Mean   :0.9    
 3rd Qu.:18.1   3rd Qu.:6211   3rd Qu.:21.0   3rd Qu.:0.0    3rd Qu.:1.0    
 Max.   :34.8   Max.   :8742   Max.   :28.0   Max.   :1.0    Max.   :1.0    
 NA's   :1662   NA's   :1662   NA's   :1662   NA's   :1662   NA's   :1662   
 hashtags_count  has_hashtags  user_name_category contains_company_name
 Min.   : 0.0   Min.   :0.0    Length:26315       Min.   :0.0          
 1st Qu.: 0.0   1st Qu.:0.0    Class :character   1st Qu.:0.0          
```

```
 Median : 0.0    Median :0.0    Mode  :character   Median :1.0
 Mean   : 0.4    Mean   :0.2                       Mean   :0.6
 3rd Qu.: 0.0    3rd Qu.:0.0                       3rd Qu.:1.0
 Max.   :11.0    Max.   :1.0                       Max.   :1.0
 NA's   :1662    NA's   :1662                      NA's   :1662
 contains_words custom_content  joining_date         days_joined
 Min.   :0.0     Min.   :0.0    Min.   :2008-04-01   Min.   :-2102
 1st Qu.:0.0     1st Qu.:0.0    1st Qu.:2009-03-01   1st Qu.: 1992
 Median :0.0     Median :0.0    Median :2009-11-01   Median : 2750
 Mean   :0.2     Mean   :0.2    Mean   :2011-02-03   Mean   : 2294
 3rd Qu.:0.0     3rd Qu.:0.0    3rd Qu.:2012-01-01   3rd Qu.: 3011
 Max.   :1.0     Max.   :1.0    Max.   :2022-10-01   Max.   : 3470
 NA's   :1662    NA's   :1662   NA's   :1662         NA's   :1662
  years_joined   months_joined
 Min.   :-5.8    Min.   :-69
 1st Qu.: 5.5    1st Qu.: 65
 Median : 7.5    Median : 90
 Mean   : 6.3    Mean   : 75
 3rd Qu.: 8.2    3rd Qu.: 98
 Max.   : 9.5    Max.   :114
 NA's   :1662    NA's   :1662
```

```r
unique(profile_data[is.na(profile_data$months_joined), ]$gvkey)
```

```
[1]  10005  12233  21238  25405 147661 178548
```

We left join operation to merge financial and profile data into a full dataset based on the matching gvkey and datadate. After the merge, it removes redundant columns and renames the remaining columns for clarity. The summary function indicates the presence of NA values, specifically 190 for financial data and 1663 for profile data.

```r
# merge into the full dataset
full_data <- left_join(x = stock_announce_financials, y = profile_data,
                      by = c("gvkey", "datadate"))
full_data <- select(full_data, -conm.y, -sic.y)
full_data <- full_data %>% rename(`conm` = conm.x, `sic` = sic.x)
```

We creates industry dummy variables and week dummy variables in the dataset named `full_data`. Industry dummy variables are generated based on the Standard Industrial Classification (SIC) codes, with a value of 1 assigned if the SIC code corresponds to a popular industry and 0 otherwise. Weekday dummy variables are created based on the day of the week for each observation, assigning a value of 1 if the observation falls on that particular day and 0 otherwise.

```r
# create industry dummy variable
popular_ind <- c("2844", "3576", "3674", "3679", "3845", "5047", "5734",
                 "5812", "6147", "6200", "6211", "6282", "6311", "6798", "7370", "7372")
full_data <- full_data %>% mutate(pop_ind = ifelse(sic %in% popular_ind, 1, 0))

# create week dummy variable
full_data <- full_data %>% mutate(weekday = weekdays(datadate))
full_data <- full_data %>% mutate(is_mon = ifelse(weekday == "Monday", 1, 0))
```

```r
full_data <- full_data %>% mutate(is_tue = ifelse(weekday == "Tuesday", 1, 0))
full_data <- full_data %>% mutate(is_wed = ifelse(weekday == "Wednesday", 1, 0))
full_data <- full_data %>% mutate(is_thu = ifelse(weekday == "Thursday", 1, 0))
full_data <- full_data %>% mutate(is_fri = ifelse(weekday == "Friday", 1, 0))
full_data <- full_data %>% mutate(is_sat = ifelse(weekday == "Saturday", 1, 0))

# save the full data
summary(full_data) # NA 190 for financial data, NA 1663 for profile data
```

```
    gvkey          kd_date                cik              datadate         
 Min.   :  1161   Length:26315       Min.   :   2488   Min.   :2016-12-29  
 1st Qu.:  8113   Class :character   1st Qu.: 101829   1st Qu.:2017-03-08  
 Median : 23252   Mode  :character   Median : 885306   Median :2017-05-16  
 Mean   : 53462                      Mean   : 754029   Mean   :2017-05-16  
 3rd Qu.:121718                      3rd Qu.:1090872   3rd Qu.:2017-07-24  
 Max.   :184500                      Max.   :1707925   Max.   :2017-10-01  
                                     NA's   :2770                           
    iid                conm               prccd             prchd        
 Length:26315       Length:26315       Min.   :   0.0   Min.   :   0.0  
 Class :character   Class :character   1st Qu.:  22.3   1st Qu.:  22.7  
 Mode  :character   Mode  :character   Median :  52.1   Median :  52.5  
                                       Mean   :  82.7   Mean   :  83.4  
                                       3rd Qu.:  94.9   3rd Qu.:  95.8  
                                       Max.   :1046.4   Max.   :1054.8  
                                                                        
    prcld             prcod              sic        p_daily_change    
 Min.   :   0.0   Min.   :   0.0   Min.   :1311   Min.   :-0.3924  
 1st Qu.:  21.9   1st Qu.:  22.3   1st Qu.:3576   1st Qu.:-0.0048  
 Median :  51.4   Median :  51.8   Median :4841   Median : 0.0000  
 Mean   :  82.0   Mean   :  82.6   Mean   :4821   Mean   : 0.0016  
 3rd Qu.:  93.8   3rd Qu.:  94.9   3rd Qu.:6141   3rd Qu.: 0.0060  
 Max.   :1041.8   Max.   :1048.2   Max.   :8742   Max.   : 0.6198  
                                   NA's   :2770                      
   volatile       p_daily_change_lag  volatile_lag     kd_type         
 Min.   :0.0000   Min.   :-0.39      Min.   :0.00   Length:26315      
 1st Qu.:0.0093   1st Qu.: 0.00      1st Qu.:0.01   Class :character  
 Median :0.0145   Median : 0.00      Median :0.01   Mode  :character  
 Mean   :0.0200   Mean   : 0.00      Mean   :0.02                     
 3rd Qu.:0.0232   3rd Qu.: 0.01      3rd Qu.:0.02                     
 Max.   :0.7460   Max.   : 0.62      Max.   :0.75                     
                  NA's   :92         NA's   :92                       
  kd_company         kd_headline          is_annouce       red_annouce    
 Length:26315       Length:26315       Min.   :0.0000   Min.   :0.0000  
 Class :character   Class :character   1st Qu.:0.0000   1st Qu.:0.0000  
 Mode  :character   Mode  :character   Median :0.0000   Median :0.0000  
                                       Mean   :0.0568   Mean   :0.0043  
                                       3rd Qu.:0.0000   3rd Qu.:0.0000  
                                       Max.   :1.0000   Max.   :1.0000  
                                                                        
 no_annoucement        bm              pe_exi              ps       
 Min.   : 0.00   Min.   :-0.34   Min.   :-2325.0   Min.   : 0.07  
 1st Qu.: 0.00   1st Qu.: 0.18   1st Qu.:     6.1   1st Qu.: 0.98  
 Median : 0.00   Median : 0.32   Median :    20.1   Median : 2.08  
 Mean   : 0.11   Mean   : 0.41   Mean   :    -4.0   Mean   : 2.84  
```

```
 3rd Qu.: 0.00   3rd Qu.: 0.62   3rd Qu.:   27.9   3rd Qu.: 3.39  
 Max.   :46.00   Max.   : 2.35   Max.   :  432.6   Max.   :38.39  
                 NA's   :190     NA's   :190       NA's   :190    
      pcf               dpr             npm            opmbd       
 Min.   :-226.65   Min.   : 0.00   Min.   :-5.76   Min.   :-5.36  
 1st Qu.:   8.12   1st Qu.: 0.00   1st Qu.: 0.01   1st Qu.: 0.08  
 Median :  13.79   Median : 0.23   Median : 0.06   Median : 0.15  
 Mean   :  17.54   Mean   : 0.60   Mean   : 0.01   Mean   : 0.12  
 3rd Qu.:  20.71   3rd Qu.: 0.50   3rd Qu.: 0.11   3rd Qu.: 0.24  
 Max.   : 243.78   Max.   :31.92   Max.   : 0.39   Max.   : 0.56  
 NA's   :190       NA's   :190     NA's   :190     NA's   :190    
     opmad            gpm             roa             roe         
 Min.   :-5.58   Min.   :-4.62   Min.   :-0.51   Min.   :-114.00  
 1st Qu.: 0.04   1st Qu.: 0.27   1st Qu.: 0.07   1st Qu.:   0.02  
 Median : 0.11   Median : 0.39   Median : 0.12   Median :   0.11  
 Mean   : 0.07   Mean   : 0.38   Mean   : 0.13   Mean   :  -0.12  
 3rd Qu.: 0.19   3rd Qu.: 0.59   3rd Qu.: 0.18   3rd Qu.:   0.20  
 Max.   : 0.50   Max.   : 0.91   Max.   : 0.90   Max.   :  35.65  
 NA's   :190     NA's   :190     NA's   :190     NA's   :190      
      roce           debt_at        de_ratio           intcov       cash_ratio  
 Min.   :-0.63   Min.   :0.00   Min.   :-141.44   Min.   : -33   Min.   :0.00  
 1st Qu.: 0.06   1st Qu.:0.11   1st Qu.:   0.83   1st Qu.:   2   1st Qu.:0.22  
 Median : 0.13   Median :0.27   Median :   1.54   Median :   5   Median :0.51  
 Mean   : 0.16   Mean   :0.30   Mean   :   2.23   Mean   :  27   Mean   :0.84  
 3rd Qu.: 0.21   3rd Qu.:0.39   3rd Qu.:   2.78   3rd Qu.:   8   3rd Qu.:1.11  
 Max.   : 1.57   Max.   :3.16   Max.   : 134.26   Max.   :7083   Max.   :6.33  
 NA's   :190     NA's   :190    NA's   :190       NA's   :190    NA's   :190   
  quick_ratio     curr_ratio      inv_turn        at_turn      debt_assets  
 Min.   :0.13   Min.   :0.40   Min.   :  0.6   Min.   :0.09   Min.   :0.11  
 1st Qu.:0.89   1st Qu.:1.25   1st Qu.:  3.8   1st Qu.:0.48   1st Qu.:0.48  
 Median :1.29   Median :1.61   Median :  5.4   Median :0.73   Median :0.64  
 Mean   :1.57   Mean   :1.98   Mean   : 26.6   Mean   :0.98   Mean   :0.66  
 3rd Qu.:1.95   3rd Qu.:2.19   3rd Qu.: 17.1   3rd Qu.:1.41   3rd Qu.:0.80  
 Max.   :7.08   Max.   :7.29   Max.   :446.3   Max.   :3.67   Max.   :3.72  
 NA's   :190    NA's   :190    NA's   :190     NA's   :190    NA's   :190   
      ptb           rect_turn         sale_nwc         divyield    
 Min.   :-12930   Min.   :  0.14   Min.   : -85.0   Min.   :0.00  
 1st Qu.:     1   1st Qu.:  5.32   1st Qu.:   1.8   1st Qu.:0.01  
 Median :     3   Median :  7.17   Median :   4.0   Median :0.02  
 Mean   :   -24   Mean   : 14.69   Mean   :  18.6   Mean   :0.02  
 3rd Qu.:     5   3rd Qu.: 12.76   3rd Qu.:  10.7   3rd Qu.:0.03  
 Max.   :  3399   Max.   :114.23   Max.   :2500.7   Max.   :0.12  
 NA's   :190      NA's   :190      NA's   :190      NA's   :190   
    TICKER               atq             bm_lag        pe_exi_lag    
 Length:26315       Min.   :     6   Min.   :-0.27   Min.   :-2260.0  
 Class :character   1st Qu.:  1551   1st Qu.: 0.18   1st Qu.:    6.6  
 Mode  :character   Median :  4976   Median : 0.33   Median :   20.0  
                    Mean   : 26197   Mean   : 0.41   Mean   :   -0.6  
                    3rd Qu.: 17471   3rd Qu.: 0.62   3rd Qu.:   28.1  
                    Max.   :503073   Max.   : 2.35   Max.   :  432.6  
                    NA's   :190      NA's   :190     NA's   :190      
     ps_lag         pcf_lag            dpr_lag         npm_lag     
 Min.   : 0.07   Min.   :-166.46   Min.   : 0.00   Min.   :-5.76  
 1st Qu.: 0.98   1st Qu.:   8.27   1st Qu.: 0.00   1st Qu.: 0.02  
```

```
Median : 2.08   Median :  13.79   Median : 0.23   Median : 0.06
Mean   : 2.80   Mean   :  17.00   Mean   : 0.61   Mean   : 0.01
3rd Qu.: 3.39   3rd Qu.:  20.38   3rd Qu.: 0.49   3rd Qu.: 0.11
Max.   :32.19   Max.   : 243.78   Max.   :31.92   Max.   : 0.39
NA's   :190     NA's   :190       NA's   :190     NA's   :190
  opmbd_lag       opmad_lag        gpm_lag         roa_lag
Min.   :-5.32   Min.   :-5.56   Min.   :-4.62   Min.   :-0.51
1st Qu.: 0.08   1st Qu.: 0.04   1st Qu.: 0.27   1st Qu.: 0.07
Median : 0.14   Median : 0.11   Median : 0.40   Median : 0.12
Mean   : 0.12   Mean   : 0.07   Mean   : 0.38   Mean   : 0.13
3rd Qu.: 0.24   3rd Qu.: 0.19   3rd Qu.: 0.59   3rd Qu.: 0.18
Max.   : 0.56   Max.   : 0.50   Max.   : 0.91   Max.   : 0.90
NA's   :190     NA's   :190     NA's   :190     NA's   :190
   roe_lag           roce_lag      debt_at_lag    de_ratio_lag
Min.   :-114.00   Min.   :-0.63   Min.   :0.00   Min.   :-141.44
1st Qu.:   0.02   1st Qu.: 0.06   1st Qu.:0.11   1st Qu.:   0.83
Median :   0.11   Median : 0.13   Median :0.27   Median :   1.53
Mean   :  -0.14   Mean   : 0.17   Mean   :0.30   Mean   :   2.13
3rd Qu.:   0.20   3rd Qu.: 0.21   3rd Qu.:0.39   3rd Qu.:   2.78
Max.   :  35.65   Max.   : 1.57   Max.   :3.16   Max.   : 134.26
NA's   :190       NA's   :190     NA's   :190    NA's   :190
  intcov_lag   cash_ratio_lag quick_ratio_lag curr_ratio_lag  inv_turn_lag
Min.   : -33   Min.   :0.00   Min.   :0.13    Min.   :0.40   Min.   :  0.6
1st Qu.:   2   1st Qu.:0.21   1st Qu.:0.90    1st Qu.:1.25   1st Qu.:  3.9
Median :   5   Median :0.51   Median :1.29    Median :1.61   Median :  5.4
Mean   :  36   Mean   :0.84   Mean   :1.57    Mean   :1.98   Mean   : 28.1
3rd Qu.:   9   3rd Qu.:1.10   3rd Qu.:1.95    3rd Qu.:2.19   3rd Qu.: 17.1
Max.   :7083   Max.   :6.33   Max.   :7.08    Max.   :7.29   Max.   :446.3
NA's   :190    NA's   :190    NA's   :190     NA's   :190    NA's   :190
 at_turn_lag   rect_turn_lag     sale_nwc_lag     divyield_lag
Min.   :0.09   Min.   :  0.14   Min.   : -85.0   Min.   :0.00
1st Qu.:0.48   1st Qu.:  5.30   1st Qu.:   1.8   1st Qu.:0.01
Median :0.73   Median :  7.01   Median :   4.0   Median :0.02
Mean   :0.98   Mean   : 14.72   Mean   :  18.3   Mean   :0.02
3rd Qu.:1.41   3rd Qu.: 12.76   3rd Qu.:  10.6   3rd Qu.:0.03
Max.   :3.67   Max.   :114.23   Max.   :2500.7   Max.   :0.12
NA's   :190    NA's   :190      NA's   :190      NA's   :190
debt_assets_lag    ptb_lag       profile_name       profile_bio
Min.   :0.11    Min.   :-12930   Length:26315       Length:26315
1st Qu.:0.49    1st Qu.:     1   Class :character   Class :character
Median :0.64    Median :     3   Mode  :character   Mode  :character
Mean   :0.66    Mean   :   -26
3rd Qu.:0.79    3rd Qu.:     5
Max.   :3.72    Max.   :  3399
NA's   :190     NA's   :190
profile_category   profile_website    profile_joining_date profile_following
Length:26315       Length:26315       Length:26315         Length:26315
Class :character   Class :character   Class :character     Class :character
Mode  :character   Mode  :character   Mode  :character     Mode  :character


profile_followers   user_name         profile_followers_num
```

```
 Length:26315       Length:26315       Min.   :      1  
 Class :character   Class :character   1st Qu.:   4398  
 Mode  :character   Mode  :character   Median :  22000  
                                       Mean   : 188033  
                                       3rd Qu.: 118600  
                                       Max.   :3800000  
                                       NA's   :1662     
 profile_following_num      TFF         coleman_liau        flesch      
 Min.   :    1         Min.   :    0   Min.   :-104.9   Min.   :-122.6  
 1st Qu.:   99         1st Qu.:    5   1st Qu.:  17.4   1st Qu.:  11.1  
 Median :  510         Median :   31   Median :  31.6   Median :  33.3  
 Mean   : 1863         Mean   : 1690   Mean   :  31.0   Mean   :  34.3  
 3rd Qu.: 1091         3rd Qu.:  218   3rd Qu.:  47.1   3rd Qu.:  56.5  
 Max.   :29500         Max.   :77778   Max.   : 138.0   Max.   :  98.8  
 NA's   :1662          NA's   :1662    NA's   :1662     NA's   :1662    
    fog_info      num_words     bio_complete     has_url     hashtags_count
 Min.   : 3.7   Min.   : 0.0   Min.   :0.0    Min.   :0.0    Min.   : 0.0  
 1st Qu.:10.0   1st Qu.:11.0   1st Qu.:0.0    1st Qu.:1.0    1st Qu.: 0.0  
 Median :14.0   Median :18.0   Median :0.0    Median :1.0    Median : 0.0  
 Mean   :14.8   Mean   :15.5   Mean   :0.2    Mean   :0.9    Mean   : 0.4  
 3rd Qu.:18.1   3rd Qu.:21.0   3rd Qu.:0.0    3rd Qu.:1.0    3rd Qu.: 0.0  
 Max.   :34.8   Max.   :28.0   Max.   :1.0    Max.   :1.0    Max.   :11.0  
 NA's   :1662   NA's   :1662   NA's   :1662   NA's   :1662   NA's   :1662  
  has_hashtags  user_name_category contains_company_name contains_words
 Min.   :0.0    Length:26315       Min.   :0.0           Min.   :0.0   
 1st Qu.:0.0    Class :character   1st Qu.:0.0           1st Qu.:0.0   
 Median :0.0    Mode  :character   Median :1.0           Median :0.0   
 Mean   :0.2                       Mean   :0.6           Mean   :0.2   
 3rd Qu.:0.0                       3rd Qu.:1.0           3rd Qu.:0.0   
 Max.   :1.0                       Max.   :1.0           Max.   :1.0   
 NA's   :1662                      NA's   :1662          NA's   :1662  
 custom_content  joining_date         days_joined     years_joined
 Min.   :0.0    Min.   :2008-04-01   Min.   :-2102   Min.   :-5.8  
 1st Qu.:0.0    1st Qu.:2009-03-01   1st Qu.: 1992   1st Qu.: 5.5  
 Median :0.0    Median :2009-11-01   Median : 2750   Median : 7.5  
 Mean   :0.2    Mean   :2011-02-03   Mean   : 2294   Mean   : 6.3  
 3rd Qu.:0.0    3rd Qu.:2012-01-01   3rd Qu.: 3011   3rd Qu.: 8.2  
 Max.   :1.0    Max.   :2022-10-01   Max.   : 3470   Max.   : 9.5  
 NA's   :1662   NA's   :1662         NA's   :1662    NA's   :1662  
 months_joined     pop_ind        weekday              is_mon     
 Min.   :-69    Min.   :0.000   Length:26315       Min.   :0.000  
 1st Qu.: 65    1st Qu.:0.000   Class :character   1st Qu.:0.000  
 Median : 90    Median :0.000   Mode  :character   Median :0.000  
 Mean   : 75    Mean   :0.295                      Mean   :0.141  
 3rd Qu.: 98    3rd Qu.:1.000                      3rd Qu.:0.000  
 Max.   :114    Max.   :1.000                      Max.   :1.000  
 NA's   :1662                                                     
     is_tue          is_wed          is_thu          is_fri     
 Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000  
 Median :0.000   Median :0.000   Median :0.000   Median :0.000  
 Mean   :0.141   Mean   :0.141   Mean   :0.144   Mean   :0.144  
 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000  
 Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000  
```

```
    is_sat
Min.   :0.000
1st Qu.:0.000
Median :0.000
Mean   :0.144
3rd Qu.:0.000
Max.   :1.000
```

```
write.csv(full_data, "full_data.csv", row.names = FALSE)
```

4. Explanatory Data Analysis

4.1 Prepare Train data for EDA and Modelling

The data preparation process involves importing df (cleaned full dataset) created from the various data scrape and extracted. We then select all relevant financial and non financial variables alongside fixed-effect variables for model construction.

First, we load "full_data.csv" as `df`, convert "sic" into a factor, selects specific financial and non financial variables from `df`, converts the "datadate" variable into an integer format suitable for merging.

Subsequently, we load "Followers_train.csv" as `train`, load SIC data from "sic.csv" and merges it with the `train` data frame based on "gvkey". Finally, we filter and merge data from `combined_var` and `train` from 1 January 2017 to 30 June 2017.

```
# avoid scientific display, keep 4 digits in display
options(scipen=999, digits=4)
# clear environment
rm(list=ls())

# store full_data as df.
df<- read.csv("full_data.csv")

#convert sic into factor
df$sic <- factor(df$sic)

#select the stock price, financial and profile variables from full data
combined_vars <- df %>% select(sic, datadate, gvkey, atq, p_daily_change, volatile, p_daily_ch

#convert datadate into integer for merging
combined_vars <- rename(combined_vars,date="datadate")
combined_vars$date <- as.Date(combined_vars$date)
combined_vars$date <- as.integer(format(combined_vars$date, "%Y%m%d"))

#merge to train data
train <- read.csv("Followers_train.csv")

#load the sic data to get the sic
```

```r
sic <- read.csv("sic.csv")
sic <- sic %>% select(gvkey, sic) %>% distinct(gvkey, sic)

#left join to train df to get sic
#no missing sic
train <- left_join(train, sic)

combined_train <- combined_vars %>% filter(date>=20170101 & date<=20170630)
Train <- left_join(combined_train,train,by=c("gvkey","date"))
Train <- Train %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y)
```

4.2 Outlier Management

When performing data extraction of company's announcement, we noted 5 companies undergoing Merger and Acquistion (M&A) and 1 company undergoing liqudation.

STANDARD REGISTER CO, GENZYME CORP, GENERAL CABLE CORP/DE and COMPELLENT TECHNOLOGIES INC was acquired by TAYLOR COMMUNICATIONS, SANOFI, PRYSMIAN and DELL respectively. VICOR CORP'S Twitter account was suspended and ARO LIQUIDATION INC was undering liquidation during the period of study.

Hence, as part of outlier management, we will remove these 6 companies from our Train dataset in order to improve the accuracy of our regression models. We later fill in the followers of these 6 companies on our Test data separately based on their last day of followers in the Train dataset.

```r
# outlier management - filter/remove the 6 M&A co. who have no financial ratios/profile variab
Train <- Train %>% filter(!gvkey %in% c("10005", "12233", "21238", "25405", "147661", "178548"
```

As the dates in our Train dataset is not consecutive (i.e. there is missing dates in between lines of dataset.), we filled in the followers of the missing dates with the followers of the latest available dates for the respective companies. By doing this, we assume that there is no increase in followers for the missing dates.

```r
# fill in the number of followers with the latest available in early days
Train <- Train %>%
  arrange(gvkey, date) %>%
  group_by(gvkey) %>%
  fill(followers,.direction = 'down') %>%
  ungroup()

Train <- Train %>%
  arrange(gvkey, date) %>%
  group_by(gvkey) %>%
  fill(sic,.direction = 'down') %>%
  ungroup()

#fill in the number of followers with the latest available in later days
Train <- Train %>%
```

```
  arrange(gvkey, date) %>%
  group_by(gvkey) %>%
  fill(followers,.direction = 'up') %>%
  ungroup()

Train <- Train %>%
  arrange(gvkey, date) %>%
  group_by(gvkey) %>%
  fill(sic,.direction = 'up') %>%
  ungroup()

summary(Train) #no NA for followers and sic
```

```
      date               gvkey             atq          p_daily_change
 Min.   :20170101   Min.   :  1161   Min.   :    29   Min.   :-0.392
 1st Qu.:20170215   1st Qu.:  7163   1st Qu.:  2141   1st Qu.:-0.005
 Median :20170401   Median : 23252   Median :  5786   Median : 0.000
 Mean   :20170366   Mean   : 52627   Mean   : 27585   Mean   : 0.001
 3rd Qu.:20170516   3rd Qu.:118502   3rd Qu.: 17630   3rd Qu.: 0.006
 Max.   :20170630   Max.   :184500   Max.   :500162   Max.   : 0.480
    volatile       p_daily_change_lag  volatile_lag          bm
 Min.   :0.0000   Min.   :-0.3921    Min.   :0.0000   Min.   :-0.246
 1st Qu.:0.0098   1st Qu.:-0.0051    1st Qu.:0.0098   1st Qu.: 0.163
 Median :0.0147   Median : 0.0000    Median :0.0147   Median : 0.332
 Mean   :0.0189   Mean   : 0.0005    Mean   :0.0187   Mean   : 0.409
 3rd Qu.:0.0229   3rd Qu.: 0.0059    3rd Qu.:0.0228   3rd Qu.: 0.619
 Max.   :0.6923   Max.   : 0.4802    Max.   :0.6923   Max.   : 2.346
     pe_exi             ps              pcf               dpr
 Min.   :-2135.0   Min.   : 0.069   Min.   :-154.72   Min.   : 0.00
 1st Qu.:   10.8   1st Qu.: 1.011   1st Qu.:   8.85   1st Qu.: 0.00
 Median :   20.8   Median : 2.129   Median :  14.10   Median : 0.24
 Mean   :    5.2   Mean   : 2.668   Mean   :  16.80   Mean   : 0.59
 3rd Qu.:   28.9   3rd Qu.: 3.443   3rd Qu.:  20.96   3rd Qu.: 0.49
 Max.   :  432.6   Max.   :17.695   Max.   : 184.75   Max.   :31.92
      npm              opmbd            opmad             gpm
 Min.   :-0.5430   Min.   :-0.150   Min.   :-0.535   Min.   :-0.112
 1st Qu.: 0.0190   1st Qu.: 0.084   1st Qu.: 0.046   1st Qu.: 0.276
 Median : 0.0700   Median : 0.158   Median : 0.116   Median : 0.372
 Mean   : 0.0659   Mean   : 0.181   Mean   : 0.125   Mean   : 0.429
 3rd Qu.: 0.1090   3rd Qu.: 0.252   3rd Qu.: 0.191   3rd Qu.: 0.613
 Max.   : 0.3830   Max.   : 0.556   Max.   : 0.501   Max.   : 0.912
      roa               roe                roce            debt_at
 Min.   :-0.0755   Min.   :-114.00   Min.   :-0.164   Min.   :0.000
 1st Qu.: 0.0690   1st Qu.:   0.03   1st Qu.: 0.060   1st Qu.:0.155
 Median : 0.1190   Median :   0.12   Median : 0.137   Median :0.280
 Mean   : 0.1422   Mean   :  -0.34   Mean   : 0.179   Mean   :0.307
 3rd Qu.: 0.1790   3rd Qu.:   0.22   3rd Qu.: 0.212   3rd Qu.:0.394
 Max.   : 0.8950   Max.   :  35.65   Max.   : 1.363   Max.   :3.161
    de_ratio            intcov         cash_ratio      quick_ratio
 Min.   :-141.44   Min.   : -26    Min.   :0.000   Min.   :0.129
 1st Qu.:   0.88   1st Qu.:   2    1st Qu.:0.210   1st Qu.:0.886
 Median :   1.57   Median :   5    Median :0.475   Median :1.286
 Mean   :   1.49   Mean   :  38    Mean   :0.774   Mean   :1.491
 3rd Qu.:   2.78   3rd Qu.:   8    3rd Qu.:1.072   3rd Qu.:1.784
```

```
 Max.   : 134.26   Max.   :7083   Max.   :4.674   Max.   :5.503  
   curr_ratio       inv_turn        at_turn       debt_assets   
 Min.   :0.401   Min.   :  0.6   Min.   :0.106   Min.   :0.107  
 1st Qu.:1.228   1st Qu.:  3.8   1st Qu.:0.475   1st Qu.:0.514  
 Median :1.549   Median :  5.5   Median :0.703   Median :0.650  
 Mean   :1.875   Mean   : 29.2   Mean   :0.970   Mean   :0.674  
 3rd Qu.:2.122   3rd Qu.: 17.3   3rd Qu.:1.400   3rd Qu.:0.796  
 Max.   :6.200   Max.   :446.3   Max.   :3.670   Max.   :3.722  
      ptb           rect_turn         sale_nwc         divyield      
 Min.   :-12930   Min.   :  0.14   Min.   : -85.0   Min.   :0.00233  
 1st Qu.:     1   1st Qu.:  5.23   1st Qu.:   1.7   1st Qu.:0.01480  
 Median :     3   Median :  6.97   Median :   3.9   Median :0.01870  
 Mean   :   -47   Mean   : 14.65   Mean   :  22.4   Mean   :0.02189  
 3rd Qu.:     5   3rd Qu.: 12.76   3rd Qu.:  10.8   3rd Qu.:0.02570  
 Max.   :  3399   Max.   :109.77   Max.   :2500.7   Max.   :0.12100  
     bm_lag         pe_exi_lag          ps_lag          pcf_lag      
 Min.   :-0.265   Min.   :-2260.0   Min.   : 0.069   Min.   :-166.5  
 1st Qu.: 0.163   1st Qu.:   10.9   1st Qu.: 1.011   1st Qu.:   9.2  
 Median : 0.332   Median :   20.7   Median : 2.098   Median :  14.2  
 Mean   : 0.412   Mean   :    1.8   Mean   : 2.612   Mean   :  16.1  
 3rd Qu.: 0.619   3rd Qu.:   29.2   3rd Qu.: 3.351   3rd Qu.:  20.6  
 Max.   : 2.346   Max.   :  432.6   Max.   :17.695   Max.   : 179.6  
    dpr_lag         npm_lag          opmbd_lag        opmad_lag     
 Min.   : 0.00   Min.   :-0.5430   Min.   :-0.135   Min.   :-0.535  
 1st Qu.: 0.00   1st Qu.: 0.0190   1st Qu.: 0.085   1st Qu.: 0.046  
 Median : 0.24   Median : 0.0680   Median : 0.150   Median : 0.116  
 Mean   : 0.52   Mean   : 0.0653   Mean   : 0.181   Mean   : 0.125  
 3rd Qu.: 0.49   3rd Qu.: 0.1090   3rd Qu.: 0.252   3rd Qu.: 0.191  
 Max.   :31.92   Max.   : 0.3830   Max.   : 0.556   Max.   : 0.501  
    gpm_lag          roa_lag            roe_lag            roce_lag     
 Min.   :-0.112   Min.   :-0.0755   Min.   :-114.00   Min.   :-0.157  
 1st Qu.: 0.276   1st Qu.: 0.0760   1st Qu.:   0.03   1st Qu.: 0.061  
 Median : 0.372   Median : 0.1190   Median :   0.12   Median : 0.136  
 Mean   : 0.428   Mean   : 0.1423   Mean   :  -0.14   Mean   : 0.180  
 3rd Qu.: 0.613   3rd Qu.: 0.1800   3rd Qu.:   0.22   3rd Qu.: 0.212  
 Max.   : 0.912   Max.   : 0.8950   Max.   :  35.65   Max.   : 1.363  
  debt_at_lag     de_ratio_lag       intcov_lag   cash_ratio_lag 
 Min.   :0.000   Min.   :-141.44   Min.   : -26   Min.   :0.000  
 1st Qu.:0.151   1st Qu.:   0.88   1st Qu.:   2   1st Qu.:0.200  
 Median :0.280   Median :   1.57   Median :   5   Median :0.475  
 Mean   :0.307   Mean   :   1.18   Mean   :  51   Mean   :0.773  
 3rd Qu.:0.394   3rd Qu.:   2.78   3rd Qu.:   9   3rd Qu.:1.072  
 Max.   :3.161   Max.   : 134.26   Max.   :7083   Max.   :4.674  
 quick_ratio_lag curr_ratio_lag   inv_turn_lag    at_turn_lag   
 Min.   :0.129   Min.   :0.401   Min.   :  0.6   Min.   :0.106  
 1st Qu.:0.880   1st Qu.:1.229   1st Qu.:  3.8   1st Qu.:0.473  
 Median :1.286   Median :1.529   Median :  5.5   Median :0.698  
 Mean   :1.495   Mean   :1.877   Mean   : 31.8   Mean   :0.975  
 3rd Qu.:1.784   3rd Qu.:2.166   3rd Qu.: 18.1   3rd Qu.:1.400  
 Max.   :5.503   Max.   :6.200   Max.   :446.3   Max.   :3.670  
 debt_assets_lag    ptb_lag       rect_turn_lag     sale_nwc_lag   
 Min.   :0.107   Min.   :-12930   Min.   :  0.14   Min.   : -85.0  
 1st Qu.:0.514   1st Qu.:     1   1st Qu.:  5.25   1st Qu.:   1.7  
 Median :0.650   Median :     3   Median :  6.97   Median :   3.9  
```

```
 Mean   :0.674   Mean   :   -25   Mean   : 14.66   Mean   :  18.0  
 3rd Qu.:0.796   3rd Qu.:     5   3rd Qu.: 12.76   3rd Qu.:  10.6  
 Max.   :3.722   Max.   :  3399   Max.   :109.77   Max.   :2500.7  
  divyield_lag       is_annouce      red_annouce     no_annoucement 
 Min.   :0.00227   Min.   :0.0000   Min.   :0.0000   Min.   : 0.00  
 1st Qu.:0.01480   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.00  
 Median :0.01870   Median :0.0000   Median :0.0000   Median : 0.00  
 Mean   :0.02160   Mean   :0.0634   Mean   :0.0055   Mean   : 0.12  
 3rd Qu.:0.02570   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 0.00  
 Max.   :0.12100   Max.   :1.0000   Max.   :1.0000   Max.   :46.00  
      TFF         coleman_liau       flesch          fog_info   
 Min.   :    0   Min.   :-104.9   Min.   :-122.6   Min.   : 3.7  
 1st Qu.:    5   1st Qu.:  17.4   1st Qu.:  11.1   1st Qu.:10.0  
 Median :   31   Median :  31.6   Median :  33.3   Median :14.0  
 Mean   : 1690   Mean   :  31.0   Mean   :  34.4   Mean   :14.8  
 3rd Qu.:  218   3rd Qu.:  47.1   3rd Qu.:  56.5   3rd Qu.:18.1  
 Max.   :77778   Max.   : 138.0   Max.   :  98.8   Max.   :34.8  
   num_words     bio_complete      has_url      hashtags_count 
 Min.   : 0.0   Min.   :0.000   Min.   :0.000   Min.   : 0.000  
 1st Qu.:11.0   1st Qu.:0.000   1st Qu.:1.000   1st Qu.: 0.000  
 Median :18.0   Median :0.000   Median :1.000   Median : 0.000  
 Mean   :15.5   Mean   :0.169   Mean   :0.899   Mean   : 0.393  
 3rd Qu.:21.0   3rd Qu.:0.000   3rd Qu.:1.000   3rd Qu.: 0.000  
 Max.   :28.0   Max.   :1.000   Max.   :1.000   Max.   :11.000  
  has_hashtags   contains_company_name contains_words  custom_content 
 Min.   :0.000   Min.   :0.000         Min.   :0.000   Min.   :0.00  
 1st Qu.:0.000   1st Qu.:0.000         1st Qu.:0.000   1st Qu.:0.00  
 Median :0.000   Median :1.000         Median :0.000   Median :0.00  
 Mean   :0.169   Mean   :0.618         Mean   :0.202   Mean   :0.18  
 3rd Qu.:0.000   3rd Qu.:1.000         3rd Qu.:0.000   3rd Qu.:0.00  
 Max.   :1.000   Max.   :1.000         Max.   :1.000   Max.   :1.00  
  days_joined     years_joined   months_joined      pop_ind     
 Min.   :-2099   Min.   :-5.75   Min.   :-69.0   Min.   :0.000  
 1st Qu.: 1932   1st Qu.: 5.29   1st Qu.: 63.0   1st Qu.:0.000  
 Median : 2694   Median : 7.38   Median : 88.0   Median :0.000  
 Mean   : 2249   Mean   : 6.16   Mean   : 73.5   Mean   :0.303  
 3rd Qu.: 2957   3rd Qu.: 8.10   3rd Qu.: 97.0   3rd Qu.:1.000  
 Max.   : 3377   Max.   : 9.25   Max.   :110.0   Max.   :1.000  
     is_mon          is_tue          is_wed          is_thu     
 Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000  
 Median :0.000   Median :0.000   Median :0.000   Median :0.000  
 Mean   :0.144   Mean   :0.144   Mean   :0.144   Mean   :0.144  
 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000  
 Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000  
     is_fri          is_sat        followers             ID           
 Min.   :0.000   Min.   :0.000   Min.   :    246   Length:16109      
 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:   6117   Class :character  
 Median :0.000   Median :0.000   Median :  20412   Mode  :character  
 Mean   :0.144   Mean   :0.138   Mean   : 157951                     
 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.: 110593                     
 Max.   :1.000   Max.   :1.000   Max.   :1929880                     
      sic      
 Min.   :1311  
```

```
1st Qu.:3576
Median :4931
Mean   :4877
3rd Qu.:6211
Max.   :8742
```

4.3 Prepare Test data for Out of Sample prediction & Kaggle submission

We first load "Followers_test.csv" before performing a left join on `test` with `sic`. This ensures that the `test` dataframe consist of SIC with no missing values. Consequently, `test` is merged with `combined_vars`. Also, we filter out specific "gvkey" (companies) lacking profile data from the final `Test` dataset. Duplicated sic.y columns generated from the merge are removed and the remaining sic.x column is renamed sic.

```
# load followers test data
test <- read.csv("Followers_test.csv")

# left join sic to test df and then left join combined variables into test df to obtain Test
test <- left_join(test, sic)
```

```
Joining with `by = join_by(gvkey)`
```

```
Test<- left_join(test,combined_vars,by=c("gvkey","date"))

# outlier management - filter/remove the 6 M&A co. who have no financial ratios/profile variab
Test <- Test %>%
  filter(!gvkey %in% c("10005", "12233", "21238", "25405", "147661", "178548")) %>%
  select(-sic.y) %>%
  rename(`sic` = sic.x)
```

4.4 Creation of Twitter Followers' growth rate

As the companies in our dataset varies in size, so we recognise the need to use daily followers' growth rate instead of the actual number of followers as our dependent variable. By using the daily followers' growth rate, we normalize the follower counts across companies. Instead of dealing with absolute numbers, which may vary widely, we focus on the relative change in followers over time. This helps to remove the influence of company size and makes the model more robust.

To achieve this, we compute the daily growth rate of followers for each company (gvkey) in the Train dataset. This involves creating a lagged variable, followers_lag, and calculating the growth rate based on current and lagged followers. Subsequently, we replace any infinite values with NA to ensure data integrity in both the Train and Test datasets. Lastly, we remove NA values from the Train dataset, with the expectation that only 89 NA values should remain.

```
# create followers growth rate in Train data
Train<- Train %>%
  group_by(gvkey) %>%
  mutate(followers_lag=lag(followers,1))%>%
```

```r
  mutate(growth_rate = (followers / lag(followers)-1)) %>%
  ungroup()

# convert infinite value to NA
Train <- Train %>%
  mutate_if(is.numeric, list(~replace(., !is.finite(.), NA)))
Test <- Test %>%
  mutate_if(is.numeric, list(~replace(., !is.finite(.), NA)))

# remove NA Value
summary(Train) #89 NA values
```

```
     date               gvkey             atq           p_daily_change
Min.   :20170101   Min.   :  1161   Min.   :    29   Min.   :-0.392
1st Qu.:20170215   1st Qu.:  7163   1st Qu.:  2141   1st Qu.:-0.005
Median :20170401   Median : 23252   Median :  5786   Median : 0.000
Mean   :20170366   Mean   : 52627   Mean   : 27585   Mean   : 0.001
3rd Qu.:20170516   3rd Qu.:118502   3rd Qu.: 17630   3rd Qu.: 0.006
Max.   :20170630   Max.   :184500   Max.   :500162   Max.   : 0.480

    volatile       p_daily_change_lag  volatile_lag          bm
Min.   :0.0000   Min.   :-0.3921    Min.   :0.0000   Min.   :-0.246
1st Qu.:0.0098   1st Qu.:-0.0051    1st Qu.:0.0098   1st Qu.: 0.163
Median :0.0147   Median : 0.0000    Median :0.0147   Median : 0.332
Mean   :0.0189   Mean   : 0.0005    Mean   :0.0187   Mean   : 0.409
3rd Qu.:0.0229   3rd Qu.: 0.0059    3rd Qu.:0.0228   3rd Qu.: 0.619
Max.   :0.6923   Max.   : 0.4802    Max.   :0.6923   Max.   : 2.346

     pe_exi              ps               pcf               dpr
Min.   :-2135.0   Min.   : 0.069   Min.   :-154.72   Min.   : 0.00
1st Qu.:   10.8   1st Qu.: 1.011   1st Qu.:   8.85   1st Qu.: 0.00
Median :   20.8   Median : 2.129   Median :  14.10   Median : 0.24
Mean   :    5.2   Mean   : 2.668   Mean   :  16.80   Mean   : 0.59
3rd Qu.:   28.9   3rd Qu.: 3.443   3rd Qu.:  20.96   3rd Qu.: 0.49
Max.   :  432.6   Max.   :17.695   Max.   : 184.75   Max.   :31.92

      npm              opmbd            opmad             gpm
Min.   :-0.5430   Min.   :-0.150   Min.   :-0.535   Min.   :-0.112
1st Qu.: 0.0190   1st Qu.: 0.084   1st Qu.: 0.046   1st Qu.: 0.276
Median : 0.0700   Median : 0.158   Median : 0.116   Median : 0.372
Mean   : 0.0659   Mean   : 0.181   Mean   : 0.125   Mean   : 0.429
3rd Qu.: 0.1090   3rd Qu.: 0.252   3rd Qu.: 0.191   3rd Qu.: 0.613
Max.   : 0.3830   Max.   : 0.556   Max.   : 0.501   Max.   : 0.912

      roa               roe                roce            debt_at
Min.   :-0.0755   Min.   :-114.00   Min.   :-0.164   Min.   :0.000
1st Qu.: 0.0690   1st Qu.:   0.03   1st Qu.: 0.060   1st Qu.:0.155
Median : 0.1190   Median :   0.12   Median : 0.137   Median :0.280
Mean   : 0.1422   Mean   :  -0.34   Mean   : 0.179   Mean   :0.307
3rd Qu.: 0.1790   3rd Qu.:   0.22   3rd Qu.: 0.212   3rd Qu.:0.394
Max.   : 0.8950   Max.   :  35.65   Max.   : 1.363   Max.   :3.161

    de_ratio            intcov        cash_ratio       quick_ratio
Min.   :-141.44   Min.   : -26   Min.   :0.000   Min.   :0.129
```

```
 1st Qu.:   0.88   1st Qu.:   2   1st Qu.:0.210   1st Qu.:0.886  
 Median :   1.57   Median :   5   Median :0.475   Median :1.286  
 Mean   :   1.49   Mean   :  38   Mean   :0.774   Mean   :1.491  
 3rd Qu.:   2.78   3rd Qu.:   8   3rd Qu.:1.072   3rd Qu.:1.784  
 Max.   : 134.26   Max.   :7083   Max.   :4.674   Max.   :5.503  

   curr_ratio       inv_turn        at_turn       debt_assets   
 Min.   :0.401   Min.   :  0.6   Min.   :0.106   Min.   :0.107  
 1st Qu.:1.228   1st Qu.:  3.8   1st Qu.:0.475   1st Qu.:0.514  
 Median :1.549   Median :  5.5   Median :0.703   Median :0.650  
 Mean   :1.875   Mean   : 29.2   Mean   :0.970   Mean   :0.674  
 3rd Qu.:2.122   3rd Qu.: 17.3   3rd Qu.:1.400   3rd Qu.:0.796  
 Max.   :6.200   Max.   :446.3   Max.   :3.670   Max.   :3.722  

      ptb           rect_turn         sale_nwc         divyield      
 Min.   :-12930   Min.   :  0.14   Min.   : -85.0   Min.   :0.00233  
 1st Qu.:     1   1st Qu.:  5.23   1st Qu.:   1.7   1st Qu.:0.01480  
 Median :     3   Median :  6.97   Median :   3.9   Median :0.01870  
 Mean   :   -47   Mean   : 14.65   Mean   :  22.4   Mean   :0.02189  
 3rd Qu.:     5   3rd Qu.: 12.76   3rd Qu.:  10.8   3rd Qu.:0.02570  
 Max.   :  3399   Max.   :109.77   Max.   :2500.7   Max.   :0.12100  

     bm_lag         pe_exi_lag          ps_lag          pcf_lag      
 Min.   :-0.265   Min.   :-2260.0   Min.   : 0.069   Min.   :-166.5  
 1st Qu.: 0.163   1st Qu.:   10.9   1st Qu.: 1.011   1st Qu.:   9.2  
 Median : 0.332   Median :   20.7   Median : 2.098   Median :  14.2  
 Mean   : 0.412   Mean   :    1.8   Mean   : 2.612   Mean   :  16.1  
 3rd Qu.: 0.619   3rd Qu.:   29.2   3rd Qu.: 3.351   3rd Qu.:  20.6  
 Max.   : 2.346   Max.   :  432.6   Max.   :17.695   Max.   : 179.6  

    dpr_lag         npm_lag          opmbd_lag        opmad_lag     
 Min.   : 0.00   Min.   :-0.5430   Min.   :-0.135   Min.   :-0.535  
 1st Qu.: 0.00   1st Qu.: 0.0190   1st Qu.: 0.085   1st Qu.: 0.046  
 Median : 0.24   Median : 0.0680   Median : 0.150   Median : 0.116  
 Mean   : 0.52   Mean   : 0.0653   Mean   : 0.181   Mean   : 0.125  
 3rd Qu.: 0.49   3rd Qu.: 0.1090   3rd Qu.: 0.252   3rd Qu.: 0.191  
 Max.   :31.92   Max.   : 0.3830   Max.   : 0.556   Max.   : 0.501  

    gpm_lag          roa_lag           roe_lag            roce_lag     
 Min.   :-0.112   Min.   :-0.0755   Min.   :-114.00   Min.   :-0.157  
 1st Qu.: 0.276   1st Qu.: 0.0760   1st Qu.:   0.03   1st Qu.: 0.061  
 Median : 0.372   Median : 0.1190   Median :   0.12   Median : 0.136  
 Mean   : 0.428   Mean   : 0.1423   Mean   :  -0.14   Mean   : 0.180  
 3rd Qu.: 0.613   3rd Qu.: 0.1800   3rd Qu.:   0.22   3rd Qu.: 0.212  
 Max.   : 0.912   Max.   : 0.8950   Max.   :  35.65   Max.   : 1.363  

  debt_at_lag     de_ratio_lag       intcov_lag   cash_ratio_lag 
 Min.   :0.000   Min.   :-141.44   Min.   : -26   Min.   :0.000  
 1st Qu.:0.151   1st Qu.:   0.88   1st Qu.:   2   1st Qu.:0.200  
 Median :0.280   Median :   1.57   Median :   5   Median :0.475  
 Mean   :0.307   Mean   :   1.18   Mean   :  51   Mean   :0.773  
 3rd Qu.:0.394   3rd Qu.:   2.78   3rd Qu.:   9   3rd Qu.:1.072  
 Max.   :3.161   Max.   : 134.26   Max.   :7083   Max.   :4.674  
```

```
quick_ratio_lag curr_ratio_lag   inv_turn_lag    at_turn_lag
Min.   :0.129   Min.   :0.401   Min.   :  0.6   Min.   :0.106
1st Qu.:0.880   1st Qu.:1.229   1st Qu.:  3.8   1st Qu.:0.473
Median :1.286   Median :1.529   Median :  5.5   Median :0.698
Mean   :1.495   Mean   :1.877   Mean   : 31.8   Mean   :0.975
3rd Qu.:1.784   3rd Qu.:2.166   3rd Qu.: 18.1   3rd Qu.:1.400
Max.   :5.503   Max.   :6.200   Max.   :446.3   Max.   :3.670

debt_assets_lag    ptb_lag        rect_turn_lag     sale_nwc_lag
Min.   :0.107   Min.   :-12930   Min.   :  0.14   Min.   : -85.0
1st Qu.:0.514   1st Qu.:      1   1st Qu.:  5.25   1st Qu.:   1.7
Median :0.650   Median :      3   Median :  6.97   Median :   3.9
Mean   :0.674   Mean   :    -25   Mean   : 14.66   Mean   :  18.0
3rd Qu.:0.796   3rd Qu.:      5   3rd Qu.: 12.76   3rd Qu.:  10.6
Max.   :3.722   Max.   :  3399   Max.   :109.77   Max.   :2500.7

 divyield_lag       is_annouce       red_annouce      no_annoucement
Min.   :0.00227   Min.   :0.0000   Min.   :0.0000   Min.   : 0.00
1st Qu.:0.01480   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.: 0.00
Median :0.01870   Median :0.0000   Median :0.0000   Median : 0.00
Mean   :0.02160   Mean   :0.0634   Mean   :0.0055   Mean   : 0.12
3rd Qu.:0.02570   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.: 0.00
Max.   :0.12100   Max.   :1.0000   Max.   :1.0000   Max.   :46.00

     TFF          coleman_liau        flesch          fog_info
Min.   :    0   Min.   :-104.9   Min.   :-122.6   Min.   : 3.7
1st Qu.:    5   1st Qu.:  17.4   1st Qu.:  11.1   1st Qu.:10.0
Median :   31   Median :  31.6   Median :  33.3   Median :14.0
Mean   : 1690   Mean   :  31.0   Mean   :  34.4   Mean   :14.8
3rd Qu.:  218   3rd Qu.:  47.1   3rd Qu.:  56.5   3rd Qu.:18.1
Max.   :77778   Max.   : 138.0   Max.   :  98.8   Max.   :34.8

  num_words     bio_complete      has_url       hashtags_count
Min.   : 0.0   Min.   :0.000   Min.   :0.000   Min.   : 0.000
1st Qu.:11.0   1st Qu.:0.000   1st Qu.:1.000   1st Qu.: 0.000
Median :18.0   Median :0.000   Median :1.000   Median : 0.000
Mean   :15.5   Mean   :0.169   Mean   :0.899   Mean   : 0.393
3rd Qu.:21.0   3rd Qu.:0.000   3rd Qu.:1.000   3rd Qu.: 0.000
Max.   :28.0   Max.   :1.000   Max.   :1.000   Max.   :11.000

 has_hashtags   contains_company_name contains_words  custom_content
Min.   :0.000   Min.   :0.000         Min.   :0.000   Min.   :0.00
1st Qu.:0.000   1st Qu.:0.000         1st Qu.:0.000   1st Qu.:0.00
Median :0.000   Median :1.000         Median :0.000   Median :0.00
Mean   :0.169   Mean   :0.618         Mean   :0.202   Mean   :0.18
3rd Qu.:0.000   3rd Qu.:1.000         3rd Qu.:0.000   3rd Qu.:0.00
Max.   :1.000   Max.   :1.000         Max.   :1.000   Max.   :1.00

 days_joined     years_joined    months_joined      pop_ind
Min.   :-2099   Min.   :-5.75   Min.   :-69.0   Min.   :0.000
1st Qu.: 1932   1st Qu.: 5.29   1st Qu.: 63.0   1st Qu.:0.000
Median : 2694   Median : 7.38   Median : 88.0   Median :0.000
Mean   : 2249   Mean   : 6.16   Mean   : 73.5   Mean   :0.303
3rd Qu.: 2957   3rd Qu.: 8.10   3rd Qu.: 97.0   3rd Qu.:1.000
```

```
 Max.   : 3377   Max.   : 9.25   Max.   :110.0   Max.   :1.000  

     is_mon          is_tue          is_wed          is_thu     
 Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000  
 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000  
 Median :0.000   Median :0.000   Median :0.000   Median :0.000  
 Mean   :0.144   Mean   :0.144   Mean   :0.144   Mean   :0.144  
 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000  
 Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000  

     is_fri          is_sat        followers             ID           
 Min.   :0.000   Min.   :0.000   Min.   :    246   Length:16109      
 1st Qu.:0.000   1st Qu.:0.000   1st Qu.:   6117   Class :character  
 Median :0.000   Median :0.000   Median :  20412   Mode  :character  
 Mean   :0.144   Mean   :0.138   Mean   : 157951                     
 3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.: 110593                     
 Max.   :1.000   Max.   :1.000   Max.   :1929880                     

      sic       followers_lag      growth_rate    
 Min.   :1311   Min.   :    246   Min.   :-0.03  
 1st Qu.:3576   1st Qu.:   6117   1st Qu.: 0.00  
 Median :4931   Median :  20407   Median : 0.00  
 Mean   :4877   Mean   : 157915   Mean   : 0.00  
 3rd Qu.:6211   3rd Qu.: 110506   3rd Qu.: 0.00  
 Max.   :8742   Max.   :1925753   Max.   : 0.05  
                NA's   :89        NA's   :89     
```

```
Train <- Train[complete.cases(Train), ]

summary(Test)
```

```
      ID                 gvkey             date                 sic      
 Length:6090        Min.   :  1161   Min.   :20170701   Min.   :1311  
 Class :character   1st Qu.:  6788   1st Qu.:20170724   1st Qu.:3674  
 Mode  :character   Median : 13092   Median :20170816   Median :5140  
                    Mean   : 52126   Mean   :20170816   Mean   :5074  
                    3rd Qu.:121718   3rd Qu.:20170908   3rd Qu.:6282  
                    Max.   :184500   Max.   :20170930   Max.   :8742  
      atq         p_daily_change        volatile        p_daily_change_lag
 Min.   :    39   Min.   :-0.1513   Min.   :0.00000   Min.   :-0.1513   
 1st Qu.:  2167   1st Qu.:-0.0046   1st Qu.:0.00921   1st Qu.:-0.0048   
 Median :  5786   Median : 0.0000   Median :0.01428   Median : 0.0000   
 Mean   : 30743   Mean   : 0.0033   Mean   :0.01863   Mean   : 0.0029   
 3rd Qu.: 20717   3rd Qu.: 0.0064   3rd Qu.:0.02237   3rd Qu.: 0.0063   
 Max.   :503073   Max.   : 0.6198   Max.   :0.26390   Max.   : 0.6198   
  volatile_lag           bm             pe_exi              ps        
 Min.   :0.00000   Min.   :-0.318   Min.   :-2325.0   Min.   : 0.085  
 1st Qu.:0.00921   1st Qu.: 0.159   1st Qu.:     9.2   1st Qu.: 0.982  
 Median :0.01427   Median : 0.307   Median :    20.4   Median : 2.103  
 Mean   :0.01842   Mean   : 0.399   Mean   :   -18.9   Mean   : 2.719  
 3rd Qu.:0.02217   3rd Qu.: 0.586   3rd Qu.:    27.2   3rd Qu.: 3.363  
 Max.   :0.31373   Max.   : 2.346   Max.   :  380.0   Max.   :13.836  
      pcf              dpr              npm              opmbd       
```

```
Min.   :-226.65   Min.   : 0.00   Min.   :-0.3020   Min.   :-0.181
1st Qu.:   7.38   1st Qu.: 0.00   1st Qu.: 0.0190   1st Qu.: 0.083
Median :  12.63   Median : 0.25   Median : 0.0740   Median : 0.159
Mean   :  17.87   Mean   : 0.64   Mean   : 0.0656   Mean   : 0.180
3rd Qu.:  21.68   3rd Qu.: 0.56   3rd Qu.: 0.1100   3rd Qu.: 0.247
Max.   : 184.75   Max.   :31.92   Max.   : 0.3880   Max.   : 0.555
    opmad              gpm             roa              roe
Min.   :-0.2460   Min.   :0.041   Min.   :-0.092   Min.   :-0.727
1st Qu.: 0.0535   1st Qu.:0.271   1st Qu.: 0.071   1st Qu.: 0.025
Median : 0.1210   Median :0.359   Median : 0.117   Median : 0.127
Mean   : 0.1253   Mean   :0.435   Mean   : 0.145   Mean   : 0.281
3rd Qu.: 0.1920   3rd Qu.:0.587   3rd Qu.: 0.177   3rd Qu.: 0.212
Max.   : 0.4690   Max.   :0.899   Max.   : 0.884   Max.   : 5.955
     roce          debt_at         de_ratio          intcov
Min.   :-0.200   Min.   :0.000   Min.   :-53.11   Min.   :-20.8
1st Qu.: 0.063   1st Qu.:0.172   1st Qu.:  0.88   1st Qu.:  2.3
Median : 0.135   Median :0.287   Median :  1.76   Median :  4.7
Mean   : 0.184   Mean   :0.324   Mean   :  4.25   Mean   : 11.4
3rd Qu.: 0.214   3rd Qu.:0.400   3rd Qu.:  3.85   3rd Qu.:  8.3
Max.   : 1.570   Max.   :3.161   Max.   :132.71   Max.   :348.9
  cash_ratio     quick_ratio      curr_ratio       inv_turn
Min.   :0.000   Min.   :0.163   Min.   :0.517   Min.   :  0.6
1st Qu.:0.197   1st Qu.:0.868   1st Qu.:1.202   1st Qu.:  3.6
Median :0.515   Median :1.257   Median :1.556   Median :  5.6
Mean   :0.744   Mean   :1.432   Mean   :1.827   Mean   : 26.0
3rd Qu.:1.069   3rd Qu.:1.762   3rd Qu.:2.084   3rd Qu.: 17.8
Max.   :6.328   Max.   :7.076   Max.   :7.286   Max.   :432.6
   at_turn       debt_assets         ptb          rect_turn
Min.   :0.102   Min.   :0.113   Min.   :-11.2   Min.   :  0.15
1st Qu.:0.458   1st Qu.:0.492   1st Qu.:  1.4   1st Qu.:  4.90
Median :0.714   Median :0.663   Median :  2.8   Median :  8.04
Mean   :0.957   Mean   :0.694   Mean   : 14.1   Mean   : 14.89
3rd Qu.:1.363   3rd Qu.:0.811   3rd Qu.:  5.8   3rd Qu.: 13.04
Max.   :3.670   Max.   :3.722   Max.   :742.0   Max.   :114.23
   sale_nwc         divyield           bm_lag         pe_exi_lag
Min.   : -9.17   Min.   :0.00345   Min.   :-0.231   Min.   :-1854.0
1st Qu.:  1.80   1st Qu.:0.01430   1st Qu.: 0.156   1st Qu.:   10.0
Median :  4.02   Median :0.01880   Median : 0.307   Median :   20.4
Mean   : 15.41   Mean   :0.02304   Mean   : 0.394   Mean   :   -1.7
3rd Qu.: 10.79   3rd Qu.:0.02710   3rd Qu.: 0.586   3rd Qu.:   28.8
Max.   :303.27   Max.   :0.09375   Max.   : 2.346   Max.   :  380.0
    ps_lag          pcf_lag          dpr_lag          npm_lag
Min.   : 0.069   Min.   :-77.16   Min.   : 0.00   Min.   :-0.2870
1st Qu.: 1.004   1st Qu.:  7.53   1st Qu.: 0.00   1st Qu.: 0.0190
Median : 2.116   Median : 12.75   Median : 0.25   Median : 0.0740
Mean   : 2.742   Mean   : 18.74   Mean   : 0.75   Mean   : 0.0654
3rd Qu.: 3.462   3rd Qu.: 21.37   3rd Qu.: 0.56   3rd Qu.: 0.1070
Max.   :13.836   Max.   :184.75   Max.   :31.92   Max.   : 0.3830
  opmbd_lag        opmad_lag         gpm_lag          roa_lag
Min.   :-0.150   Min.   :-0.215   Min.   :0.041   Min.   :-0.0755
1st Qu.: 0.081   1st Qu.: 0.050   1st Qu.:0.271   1st Qu.: 0.0680
Median : 0.161   Median : 0.121   Median :0.358   Median : 0.1170
Mean   : 0.180   Mean   : 0.125   Mean   :0.436   Mean   : 0.1477
3rd Qu.: 0.247   3rd Qu.: 0.192   3rd Qu.:0.590   3rd Qu.: 0.1780
```

```
 Max.   : 0.555   Max.   : 0.469   Max.   :0.899   Max.   : 0.8840  
    roe_lag           roce_lag        debt_at_lag     de_ratio_lag   
 Min.   :-114.00   Min.   :-0.164   Min.   :0.000   Min.   :-50.96  
 1st Qu.:   0.02   1st Qu.: 0.063   1st Qu.:0.171   1st Qu.:  0.93  
 Median :   0.13   Median : 0.135   Median :0.287   Median :  1.76  
 Mean   :  -0.06   Mean   : 0.187   Mean   :0.324   Mean   :  4.58  
 3rd Qu.:   0.21   3rd Qu.: 0.214   3rd Qu.:0.389   3rd Qu.:  3.85  
 Max.   :   5.95   Max.   : 1.570   Max.   :3.161   Max.   :108.38  
   intcov_lag    cash_ratio_lag  quick_ratio_lag curr_ratio_lag 
 Min.   : -9.2   Min.   :0.000   Min.   :0.129   Min.   :0.517  
 1st Qu.:  2.3   1st Qu.:0.197   1st Qu.:0.903   1st Qu.:1.210  
 Median :  4.8   Median :0.515   Median :1.257   Median :1.578  
 Mean   : 11.8   Mean   :0.739   Mean   :1.438   Mean   :1.830  
 3rd Qu.:  8.6   3rd Qu.:1.069   3rd Qu.:1.902   3rd Qu.:2.107  
 Max.   :348.9   Max.   :4.674   Max.   :5.503   Max.   :6.167  
  inv_turn_lag    at_turn_lag    debt_assets_lag    ptb_lag      
 Min.   :  0.6   Min.   :0.102   Min.   :0.113   Min.   :-12930  
 1st Qu.:  3.6   1st Qu.:0.458   1st Qu.:0.495   1st Qu.:     1  
 Median :  5.6   Median :0.714   Median :0.657   Median :     3  
 Mean   : 25.9   Mean   :0.962   Mean   :0.696   Mean   :   -26  
 3rd Qu.: 17.8   3rd Qu.:1.363   3rd Qu.:0.813   3rd Qu.:     6  
 Max.   :432.6   Max.   :3.670   Max.   :3.722   Max.   :   742  
 rect_turn_lag     sale_nwc_lag     divyield_lag       is_annouce   
 Min.   :  0.15   Min.   : -45.8   Min.   :0.00345   Min.   :0.000  
 1st Qu.:  4.88   1st Qu.:   1.9   1st Qu.:0.01450   1st Qu.:0.000  
 Median :  8.07   Median :   4.0   Median :0.01860   Median :0.000  
 Mean   : 15.03   Mean   :  25.3   Mean   :0.02311   Mean   :0.067  
 3rd Qu.: 13.04   3rd Qu.:  11.4   3rd Qu.:0.02720   3rd Qu.:0.000  
 Max.   :114.23   Max.   :2500.7   Max.   :0.12100   Max.   :1.000  
  red_annouce     no_annoucement        TFF         coleman_liau   
 Min.   :0.0000   Min.   : 0.000   Min.   :    0   Min.   :-104.9  
 1st Qu.:0.0000   1st Qu.: 0.000   1st Qu.:    7   1st Qu.:  17.0  
 Median :0.0000   Median : 0.000   Median :   37   Median :  31.6  
 Mean   :0.0028   Mean   : 0.123   Mean   : 2164   Mean   :  30.4  
 3rd Qu.:0.0000   3rd Qu.: 0.000   3rd Qu.:  262   3rd Qu.:  49.1  
 Max.   :1.0000   Max.   :13.000   Max.   :77778   Max.   : 138.0  
     flesch          fog_info       num_words     bio_complete  
 Min.   :-122.6   Min.   : 3.70   Min.   : 0.0   Min.   :0.000  
 1st Qu.:  11.1   1st Qu.: 9.91   1st Qu.:10.0   1st Qu.:0.000  
 Median :  33.3   Median :14.00   Median :18.0   Median :0.000  
 Mean   :  35.0   Mean   :14.59   Mean   :15.1   Mean   :0.172  
 3rd Qu.:  61.2   3rd Qu.:18.00   3rd Qu.:21.0   3rd Qu.:0.000  
 Max.   :  98.8   Max.   :34.80   Max.   :28.0   Max.   :1.000  
    has_url      hashtags_count    has_hashtags   contains_company_name
 Min.   :0.000   Min.   : 0.000   Min.   :0.000   Min.   :0.000        
 1st Qu.:1.000   1st Qu.: 0.000   1st Qu.:0.000   1st Qu.:0.000        
 Median :1.000   Median : 0.000   Median :0.000   Median :1.000        
 Mean   :0.891   Mean   : 0.422   Mean   :0.181   Mean   :0.638        
 3rd Qu.:1.000   3rd Qu.: 0.000   3rd Qu.:0.000   3rd Qu.:1.000        
 Max.   :1.000   Max.   :11.000   Max.   :1.000   Max.   :1.000        
 contains_words custom_content   days_joined     years_joined   months_joined
 Min.   :0.0    Min.   :0.000   Min.   :-1918   Min.   :-5.25   Min.   :-63.0  
 1st Qu.:0.0    1st Qu.:0.000   1st Qu.: 2343   1st Qu.: 6.42   1st Qu.: 76.0  
 Median :0.0    Median :0.000   Median : 2905   Median : 7.95   Median : 95.0  
```

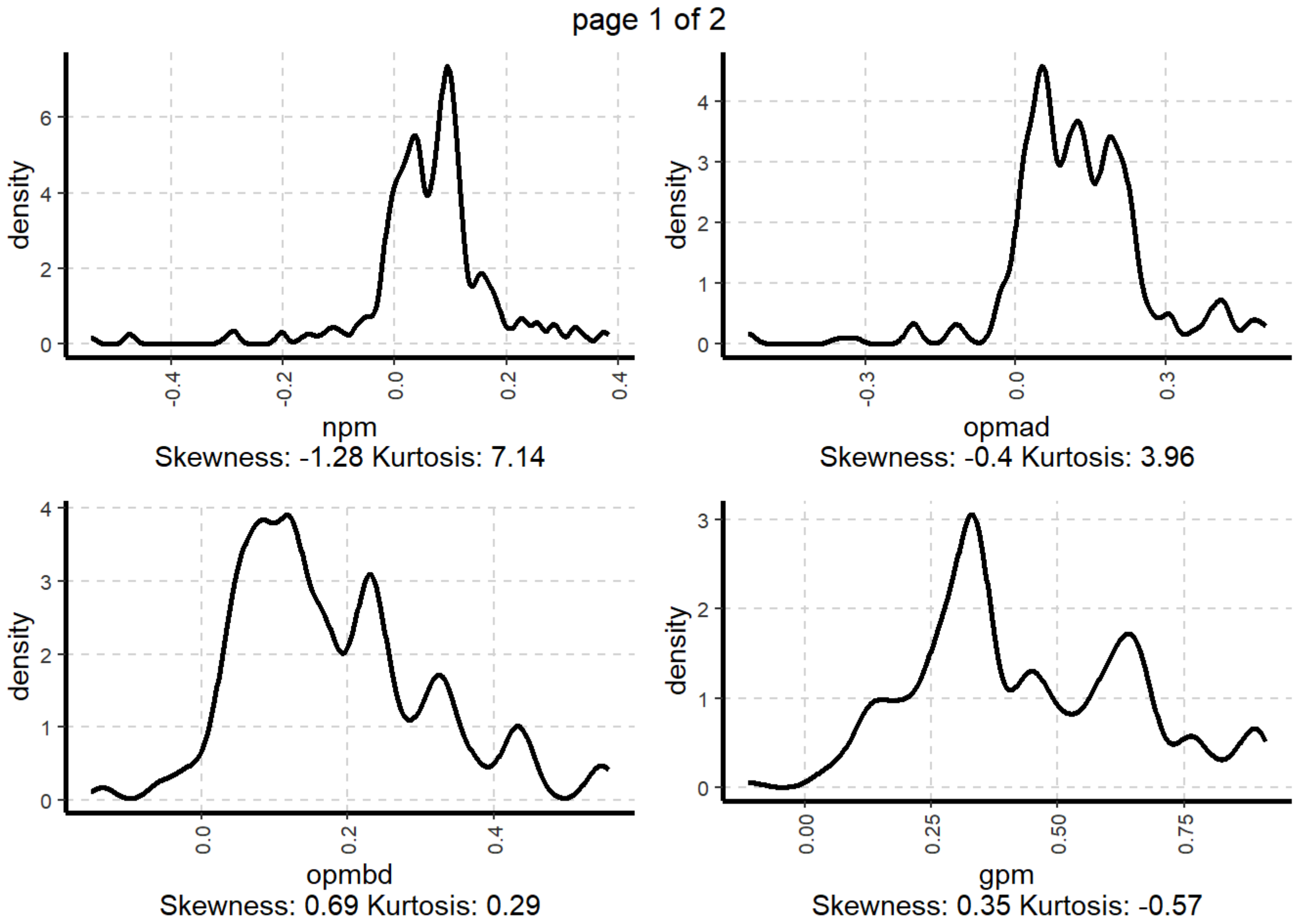
```
Mean   :0.2    Mean   :0.162   Mean   : 2428   Mean   : 6.65   Mean   : 79.4
3rd Qu.:0.0    3rd Qu.:0.000   3rd Qu.: 3098   3rd Qu.: 8.48   3rd Qu.:101.0
Max.   :1.0    Max.   :1.000   Max.   : 3469   Max.   : 9.50   Max.   :113.0
   pop_ind          is_mon          is_tue          is_wed
Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000
Median :0.000   Median :0.000   Median :0.000   Median :0.000
Mean   :0.309   Mean   :0.157   Mean   :0.163   Mean   :0.165
3rd Qu.:1.000   3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000
Max.   :1.000   Max.   :1.000   Max.   :1.000   Max.   :1.000
    is_thu          is_fri          is_sat
Min.   :0.000   Min.   :0.000   Min.   :0.000
1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.000
Median :0.000   Median :0.000   Median :0.000
Mean   :0.163   Mean   :0.162   Mean   :0.102
3rd Qu.:0.000   3rd Qu.:0.000   3rd Qu.:0.000
Max.   :1.000   Max.   :1.000   Max.   :1.000
```

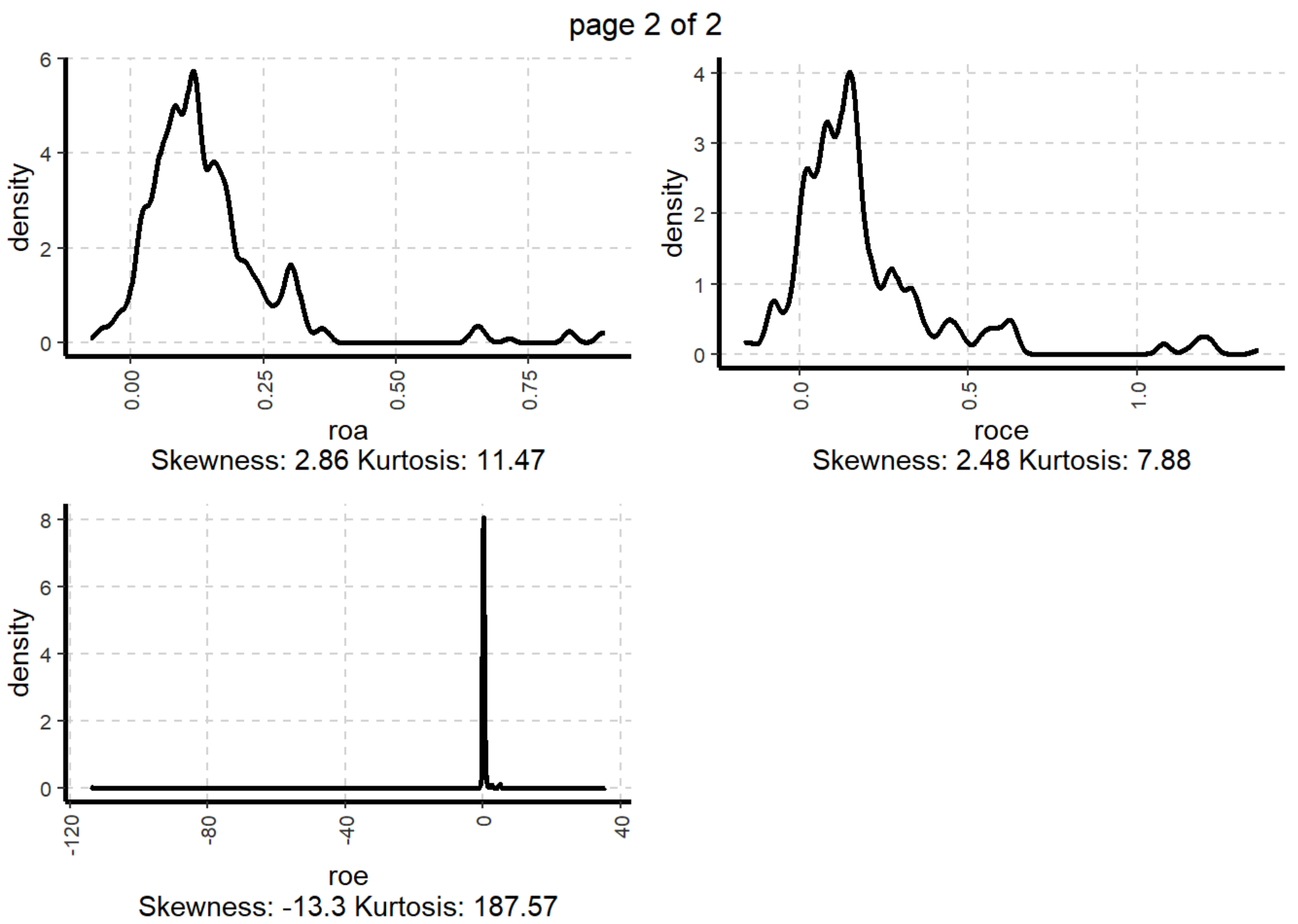
4.5 Analysis of the distribution of Financial ratios

Subsequently, we perform Explanatory Data Analysis to better understand the relationship between dependent variable (Twitter followers growth rate) and some of the independent variables (, stock price, financial ratios and non-financial variables). This in turn help us identify missing values, outliers, categorical variables, distributions, and correlations of the dataset.

```
#examine the distribution of the profitability ratios variables
Train %>%
  select(npm, opmbd, opmad, gpm, roa, roe, roce)%>%
  ExpNumViz(target = NULL,
            nlim = 10,
            Page = c(2,2))
```

```
$`0`
```

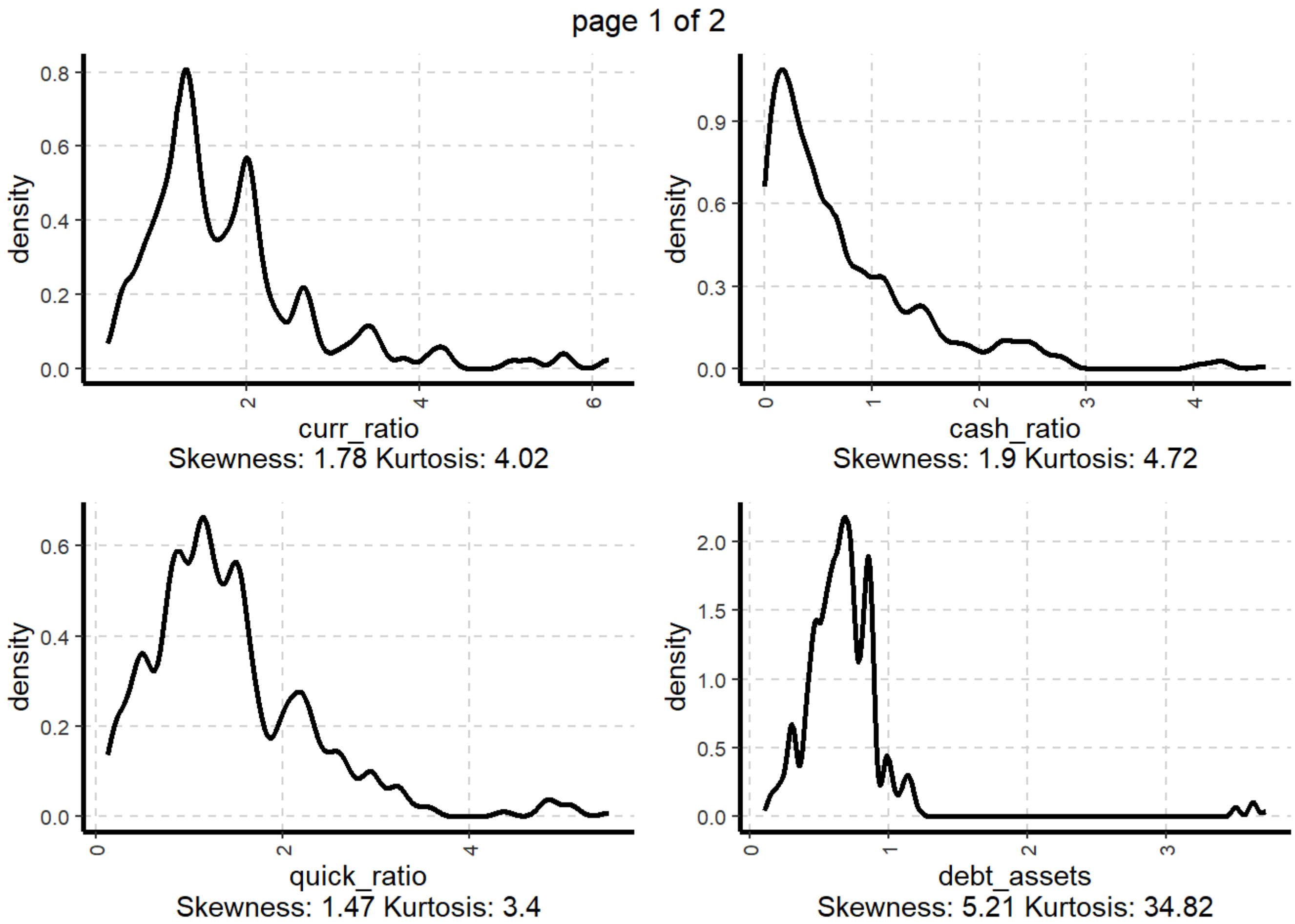


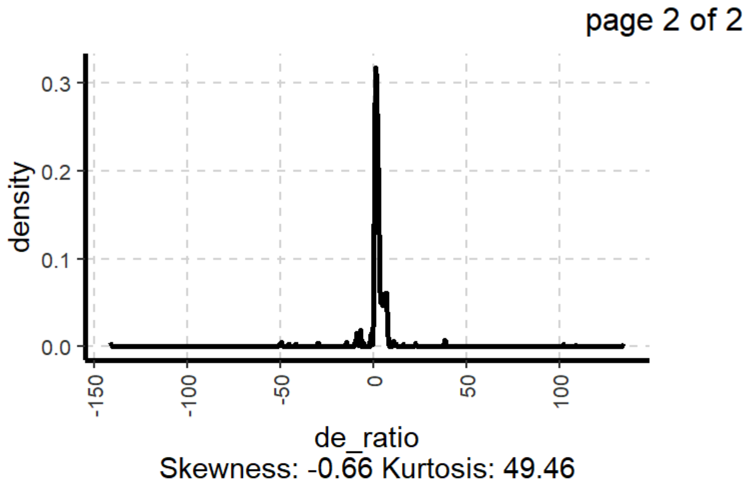


Observation: Majority of the companies profitability ratios are between 0 and 1, where the density peak at 0.2 to 0.3. The skewness in distribution of the profitability ratios are consistent across the different profitability ratios, with the exception of roa and roce which witness a wider spread of values between 0 and 0.5.

```
#examine the distribution of liquidity and solvency ratios
Train %>%
  select(curr_ratio, quick_ratio,cash_ratio,debt_assets,de_ratio)%>%
  ExpNumViz(target = NULL,
            nlim = 10,
            Page = c(2,2))
```

```
$`0`
```



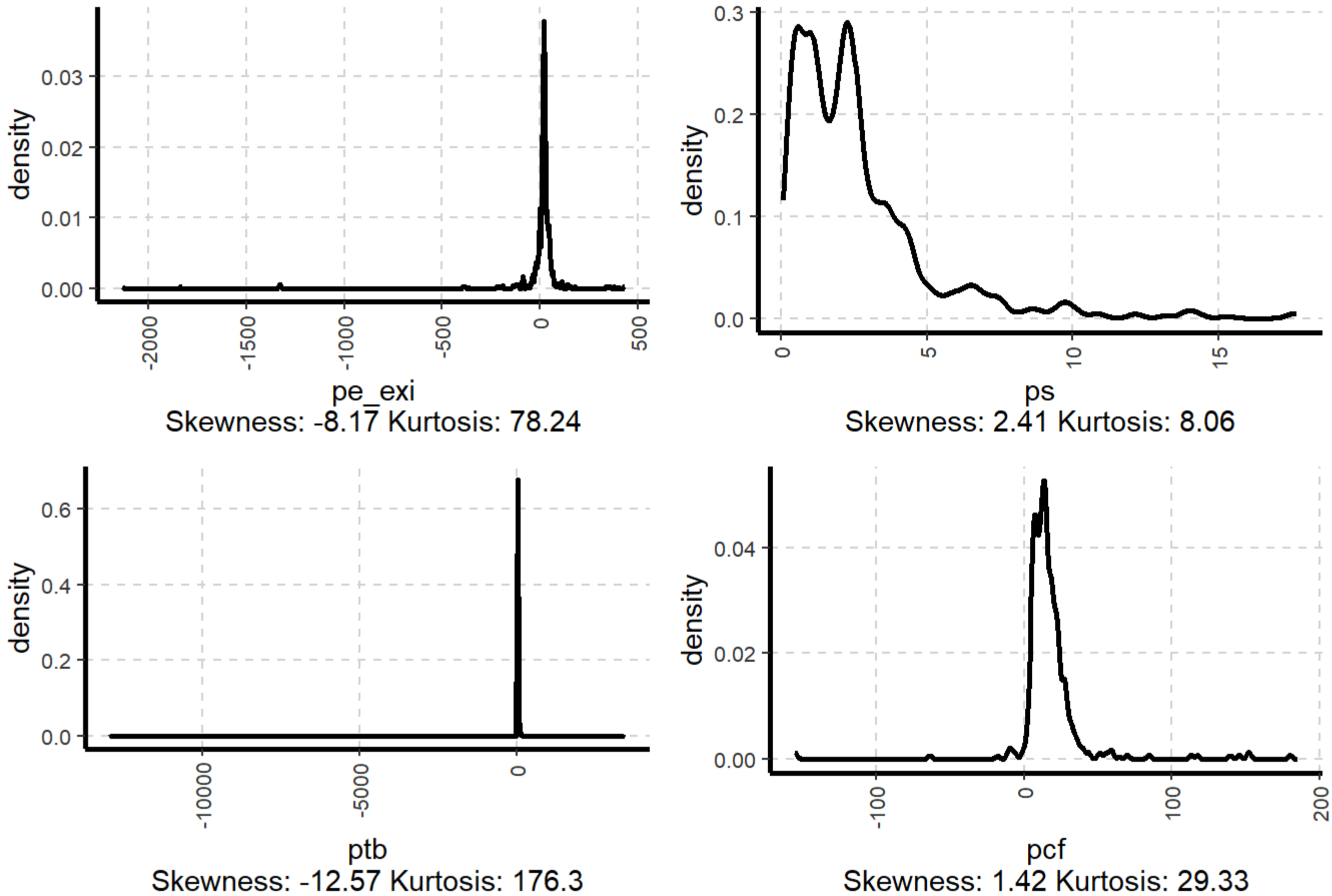


Observations : Majority of the companies' liquidity ratios ranges between 0 and 2, with a few companies having higher ratios and performing better in terms of liquidity. However, most companies appears to be healthy in terms of solvency, with relatively lower debt to assets ratio. Debt to equity ratios peak at 1 indicating low reliance of debts compared to equity for these companies, indicative of companies' strong financial position with minimal financial risk associated with debt.

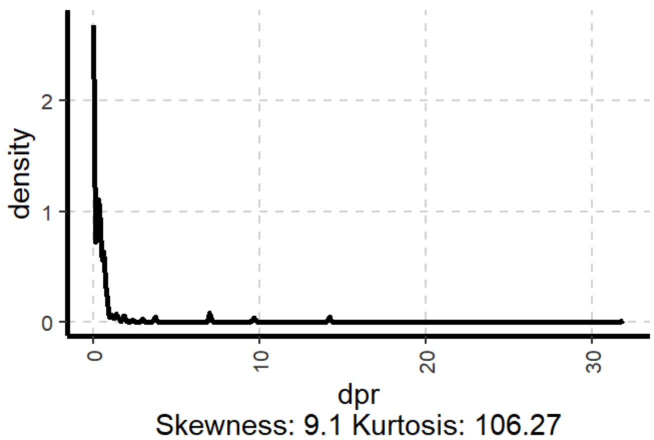
```
#examine the distribution of valuation ratios
Train %>%
  select(pe_exi, ptb, ps, pcf, dpr)%>%
  ExpNumViz(target = NULL,
            nlim = 10,
            Page = c(2,2))
```

```
$`0`
```

page 1 of 2



page 2 of 2



Observation: Price earnings (pe_exi) measures a company's current share price to its earnings per share. higher PE ratio suggests that investors are willing to pay more for each unit of earnings, indicating a potentially overvalued stock.

Price to sales (ps) ratio helps evaluate a company's valuation relative to its sales. A lower PS ratio may indicate that the stock is undervalued compared to its revenue.

Price to book (ptb) ratio compares a company's market capitalization to its book value per share (BVPS). A lower PTB ratio may suggest that the stock is undervalued relative to its book value.

Price to cashflow (pcf) ratio compares a company's market capitalization to its cash flow per share. It helps assess a company's valuation relative to its cash flow generation. A lower PCF ratio may indicate that the stock is undervalued compared to its cash flow

Majority of the companies has lower pe/ps/ptb/pcf ratio as suggested by the density plot may suggest possible chances of these companies stock being undervalued.

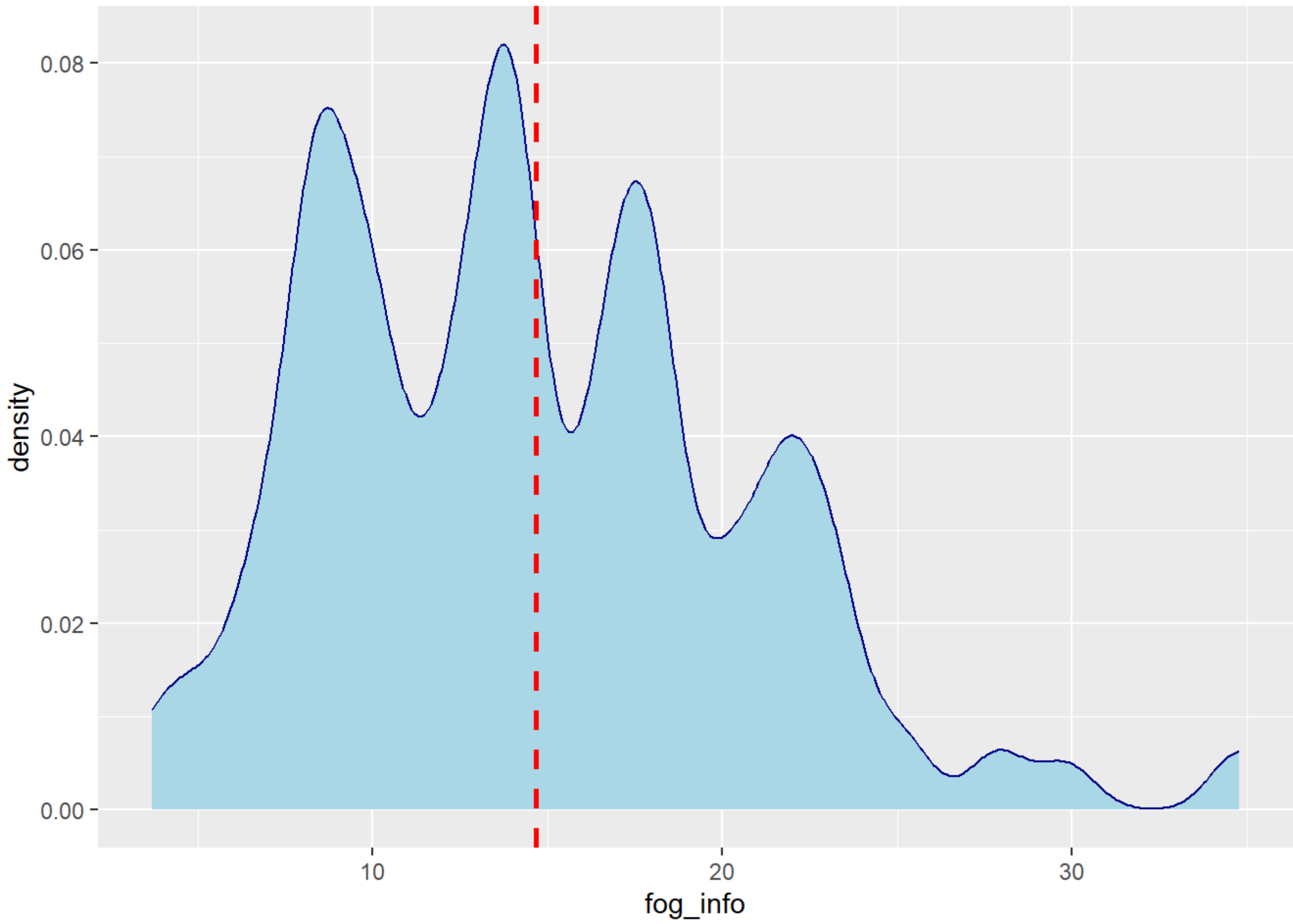
Dividend payout ratio (dpr) provides valuable insights into how much of a company's earnings are returned to shareholders as dividends.The dpr is relatively consistent across industry indicative that most companies pays about the same proportions of their earnings as dividends compared to their industry peers

4.6 Visual Analysis of Profile Bio Readability

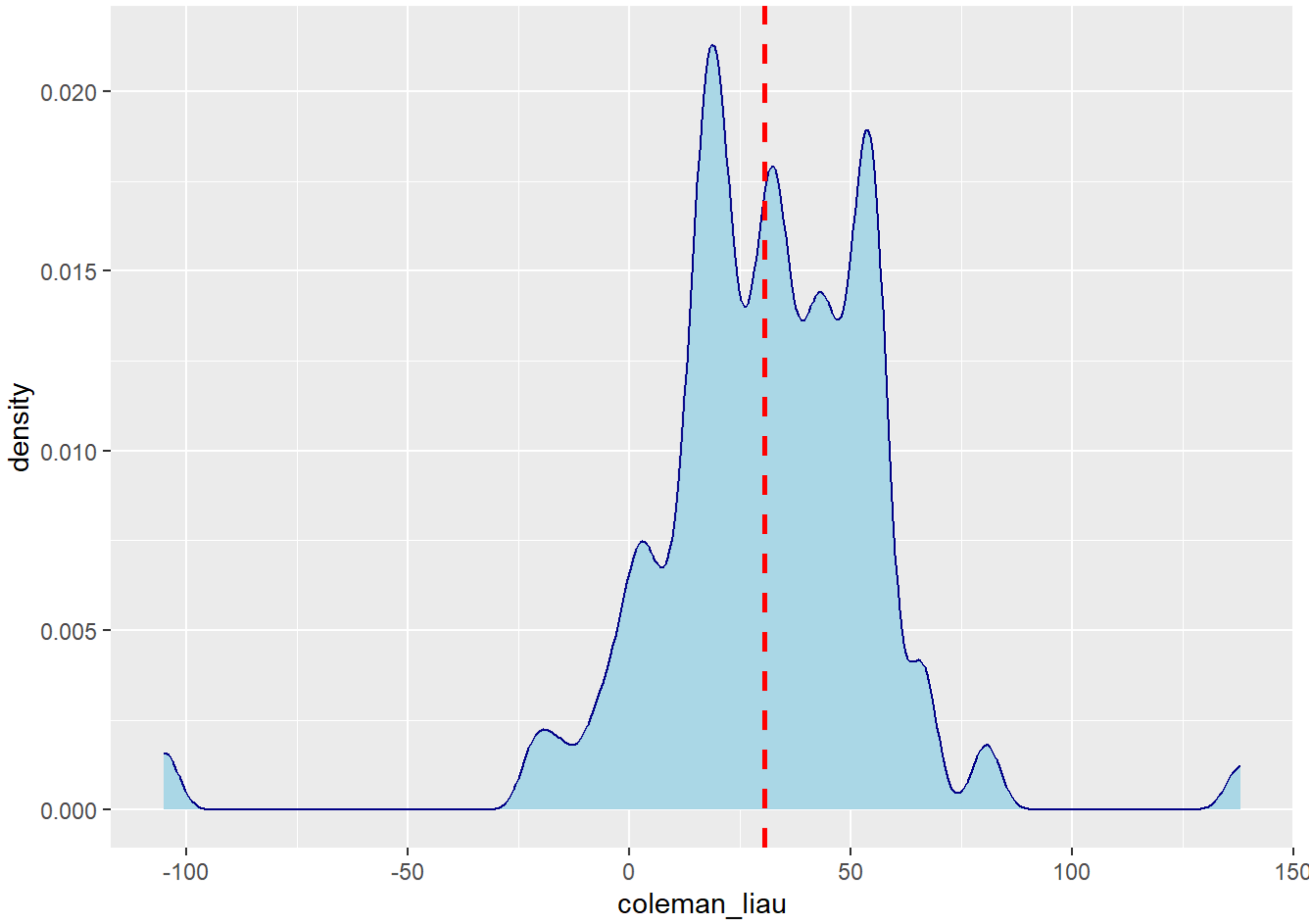
Using ggplot aes function, we perform visual analysis of the readability of the profile bio across the 95 companies

```
#examine the readability across the profile bio of the 95 companies
#fog_info
ggplot(Train, aes(x=fog_info)) + geom_density(color="darkblue", fill="lightblue")+
geom_vline(aes(xintercept=mean(fog_info)), color="red", linetype="dashed", size=1)
```

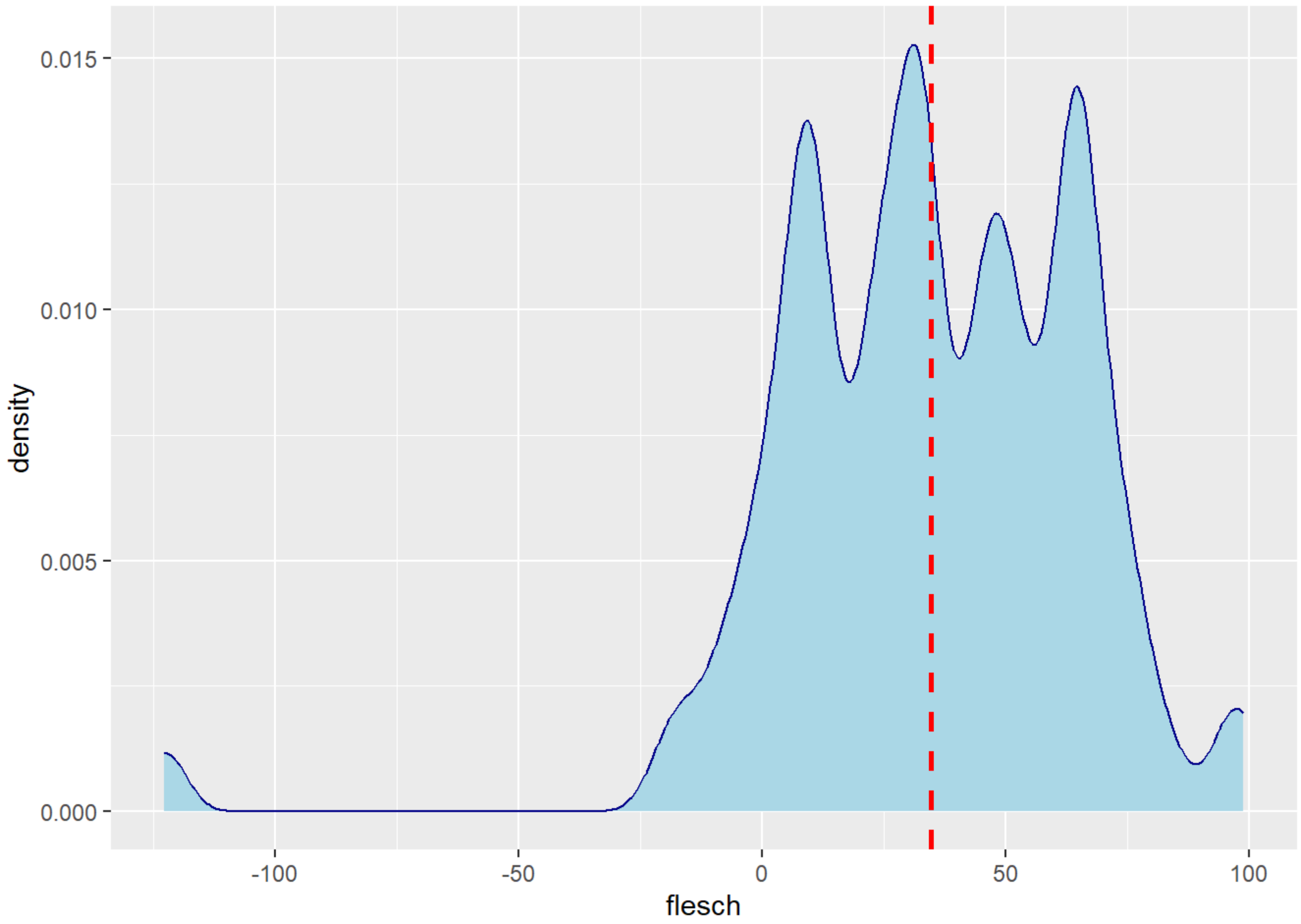
```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
ℹ Please use `linewidth` instead.
```




```
#coleman_liau
ggplot(Train, aes(x=coleman_liau)) + geom_density(color="darkblue", fill="lightblue")+
geom_vline(aes(xintercept=mean(coleman_liau)), color="red", linetype="dashed", size=1)
```

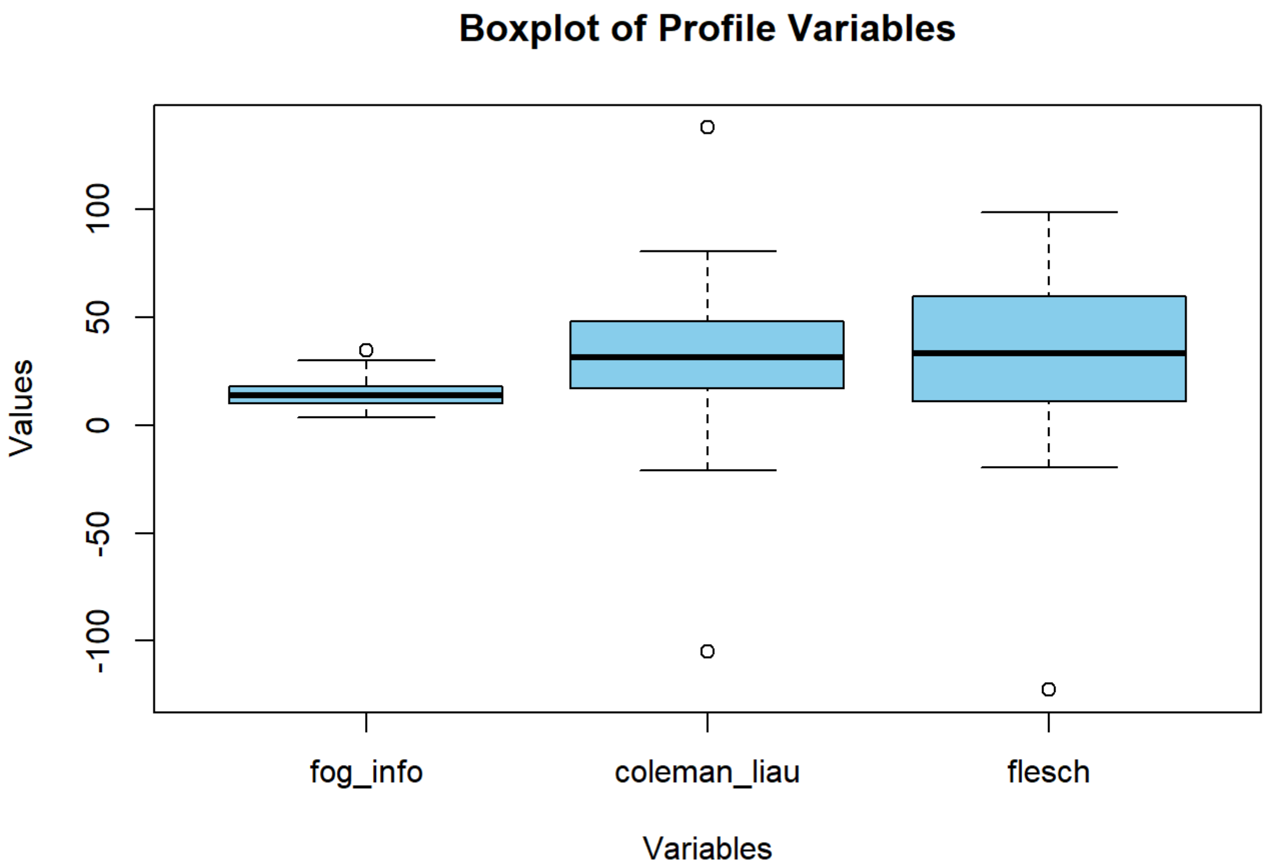



```
#flesch
ggplot(Train, aes(x=flesch)) + geom_density(color="darkblue", fill="lightblue")+
geom_vline(aes(xintercept=mean(flesch)), color="red", linetype="dashed", size=1)
```



Observation: the mean of flesch

```
# select the columns containing financial ratios and profile variables
variables_of_interest <- c("fog_info", "coleman_liau", "flesch")

# create a boxplot for each variable
boxplot(Train[, variables_of_interest],
        main = "Boxplot of Profile Variables",
        xlab = "Variables",
        ylab = "Values",
        col = "skyblue",  # Boxplot color
        border = "black", # Boxplot border color
        horizontal = FALSE)  # Horizontal boxplot for better visualization if many variables
```



Observation: Fog_info and Coleman_liau measures readability of profile_bio in the same measure. The lower the score the better the readability. The Flesch score on the other hand is reflects better readability with a higher score.

Intuitively, it measures the years of formal education required for reading a document. The interquartile range for the Fog_info measurement on the companies profile bios readability lies between 0 to 30 with a median of around 10. The low median (10) for Fog_info is indicative that the companies' profile bios can be generally understand by person with little formal education.

While the measurement of Coleman_liau readability on the companies' profile bios has an interquartile range of 10-40 with a median of around 25. This indicates that the twitter companies profile bios can be easily understood by the general public, which is aligned to our understanding that most profile bio is designed to be well understood and appealing to Twitter followers.

It is noted that there is a few outliers in the readability score which could be a result of removal of hashtags from the profile bio before assessing its readability. some of the profile bios are predominantly hashtags and removing the hashtags before assessing its readability can result in distorted results. In order to not penalise profile bios with many hashtags on readability, we balance it out by including the no of hashtags as part of our profile variables for prediction of Twitter followers growth.

4.7 Descriptive Statistic Table

Using the descriptive statistic table, we assess the mean, standard deviation, median of the financial variables. This provides insights on the financial ratio's distribution and variability.

```r
# examine the descriptive statistics (mean, standard deviation, median)
library(psych)
```

```
Warning: package 'psych' was built under R version 4.3.3


Attaching package: 'psych'

The following object is masked from 'package:randomForest':

    outlier

The following objects are masked from 'package:ggplot2':

    %+%, alpha
```

```r
table_stat <- describe(Train)
table_stat[,c(3,4,5,8,9)]
```

```
                         mean       sd      median         min         max
date              20170367.49   171.07 20170331.00 20170102.00 20170630.00
gvkey                52637.36 64812.77    14412.00     1161.00   184500.00
atq                  28847.89 68309.71     5345.10       28.76   500162.00
p_daily_change           0.00     0.03        0.00       -0.39        0.48
volatile                 0.02     0.02        0.01        0.00        0.56
p_daily_change_lag       0.00     0.02        0.00       -0.39        0.48
volatile_lag             0.02     0.02        0.01        0.00        0.69
bm                       0.38     0.34        0.31       -0.25        2.35
pe_exi                   0.90   182.47       20.32    -2135.00      432.60
ps                       2.62     2.46        2.13        0.07       17.70
pcf                     17.13    22.48       14.10     -154.72      184.75
dpr                      0.62     2.19        0.24        0.00       31.92
npm                      0.06     0.11        0.07       -0.54        0.38
opmbd                    0.18     0.13        0.15       -0.15        0.56
opmad                    0.12     0.14        0.12       -0.54        0.50
gpm                      0.44     0.21        0.36       -0.11        0.91
roa                      0.15     0.14        0.12       -0.08        0.90
roe                     -0.24     8.07        0.12     -114.00       35.65
roce                     0.19     0.23        0.14       -0.16        1.36
debt_at                  0.32     0.38        0.29        0.00        3.16
de_ratio                 1.80    16.43        1.64     -141.44      134.26
intcov                  44.63   477.78        4.79      -26.34     7082.56
cash_ratio               0.73     0.75        0.48        0.00        4.67
quick_ratio              1.43     0.88        1.25        0.13        5.50
curr_ratio               1.80     1.01        1.53        0.40        6.20
inv_turn                28.79    72.67        5.50        0.57      446.27
at_turn                  0.98     0.72        0.71        0.11        3.67
debt_assets              0.70     0.41        0.66        0.11        3.72
ptb                    -36.48   928.54        2.85   -12929.64     3399.45
rect_turn               15.11    20.86        7.42        0.14      109.77
sale_nwc                26.63   213.01        3.93      -84.96     2500.65
divyield                 0.02     0.01        0.02        0.00        0.12
bm_lag                   0.39     0.34        0.32       -0.27        2.35
```

pe_exi_lag	-4.66	213.60	20.16	-2260.00	432.60
ps_lag	2.57	2.37	2.10	0.11	17.70
pcf_lag	16.24	21.34	14.16	-166.46	179.62
dpr_lag	0.54	1.51	0.24	0.00	31.92
npm_lag	0.06	0.11	0.07	-0.54	0.38
opmbd_lag	0.18	0.13	0.15	-0.14	0.56
opmad_lag	0.12	0.14	0.12	-0.54	0.50
gpm_lag	0.43	0.22	0.36	-0.11	0.91
roa_lag	0.15	0.13	0.12	-0.08	0.90
roe_lag	-0.06	6.60	0.12	-114.00	35.65
roce_lag	0.19	0.23	0.14	-0.16	1.36
debt_at_lag	0.32	0.38	0.28	0.00	3.16
de_ratio_lag	1.37	16.21	1.60	-141.44	134.26
intcov_lag	62.94	595.90	5.02	-26.34	7082.56
cash_ratio_lag	0.73	0.75	0.48	0.00	4.67
quick_ratio_lag	1.44	0.88	1.25	0.13	5.50
curr_ratio_lag	1.81	0.99	1.53	0.40	6.20
inv_turn_lag	31.44	79.27	5.50	0.57	446.27
at_turn_lag	0.99	0.72	0.71	0.11	3.67
debt_assets_lag	0.70	0.41	0.66	0.11	3.72
ptb_lag	-16.79	766.18	2.82	-12929.64	3399.45
rect_turn_lag	15.08	20.42	7.42	0.14	109.77
sale_nwc_lag	21.24	176.71	3.84	-84.96	2500.65
divyield_lag	0.02	0.01	0.02	0.00	0.12
is_annouce	0.07	0.26	0.00	0.00	1.00
red_annouce	0.01	0.08	0.00	0.00	1.00
no_annoucement	0.14	0.68	0.00	0.00	18.00
TFF	2079.82	9803.79	36.85	0.10	77777.78
coleman_liau	30.53	27.67	31.63	-104.88	137.96
flesch	34.69	31.90	33.32	-122.60	98.84
fog_info	14.67	6.03	14.00	3.70	34.80
num_words	15.14	7.35	18.00	0.00	28.00
bio_complete	0.17	0.38	0.00	0.00	1.00
has_url	0.89	0.31	1.00	0.00	1.00
hashtags_count	0.41	1.47	0.00	0.00	11.00
has_hashtags	0.18	0.38	0.00	0.00	1.00
contains_company_name	0.64	0.48	1.00	0.00	1.00
contains_words	0.20	0.40	0.00	0.00	1.00
custom_content	0.17	0.37	0.00	0.00	1.00
days_joined	2280.92	1205.10	2762.00	-2098.00	3377.00
years_joined	6.24	3.30	7.56	-5.74	9.25
months_joined	74.55	39.34	90.00	-68.00	110.00
pop_ind	0.31	0.46	0.00	0.00	1.00
is_mon	0.16	0.36	0.00	0.00	1.00
is_tue	0.17	0.37	0.00	0.00	1.00
is_wed	0.17	0.37	0.00	0.00	1.00
is_thu	0.17	0.38	0.00	0.00	1.00
is_fri	0.17	0.37	0.00	0.00	1.00
is_sat	0.09	0.29	0.00	0.00	1.00
followers	195248.35	363236.77	24772.50	246.00	1929880.00
ID*	6047.50	3491.38	6047.50	1.00	12094.00
sic	5056.90	1834.70	5065.00	1311.00	8742.00
followers_lag	195128.61	362916.40	24765.50	246.00	1925753.00
growth_rate	0.00	0.00	0.00	-0.03	0.05

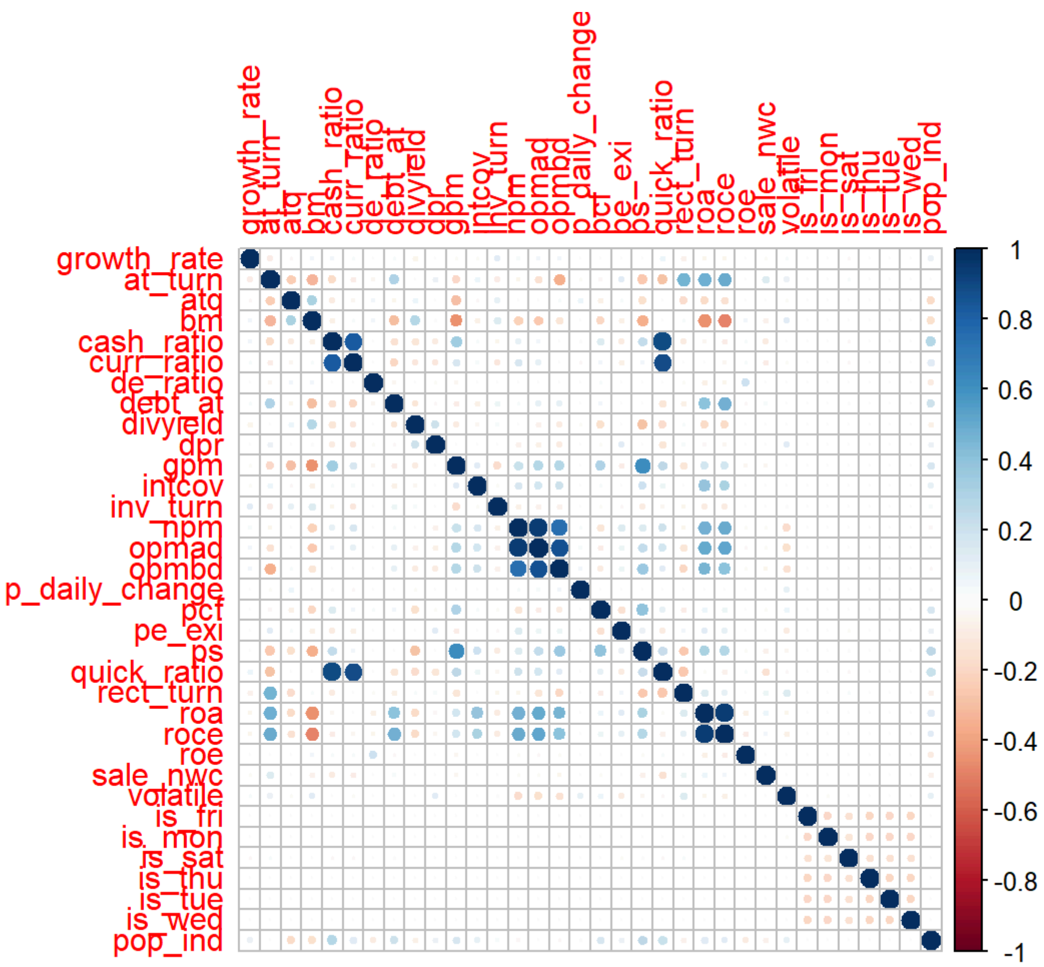
4.8 Multicollinearity Test

To study the multicollinearity between financial variables, we construct a correlation matrix to explore pairwise correlations among the variables. By visualizing the correlation matrix using corrplot(), we are able to identify the correlated pairs. Consequently, we remove 1 of the correlated variables in our regression model to reduce the introduction of error term in the model (i.e.one of the assumptions of regression is that variables are not correlated.)

```
# multicollinearity test on financial variables
# select financial correlation variables
correlation_fin_var <- Train %>% select(growth_rate, at_turn, atq, bm, cash_ratio, curr_ratio,
                                        de_ratio, debt_at, divyield, dpr, gpm, intcov, inv_turn, n
                                        opmad, opmbd, p_daily_change, pcf, pe_exi, ps, quick_ratio
                                        rect_turn, roa, roce, roe, sale_nwc, volatile, is_fri, is_
                                        is_sat, is_thu, is_tue,is_wed, pop_ind)

# correlation matrix
library(corrplot)
```

```
corrplot 0.92 loaded
```

```
correlation <- cor(correlation_fin_var, use="pairwise.complete.obs")
corrplot(correlation)
```



Similarly, we conducted a multicollinearity test on non-financial variables. We first construct a correlation matrix to explore pairwise correlations among non financial variables. By visualizing the correlation matrix using corrplot(), we aim to identify multicollinearity issues. It is worthy to note that the days_joined, months_joined and years_joined are highly correlated. Thus, we only retain days_joined variable in the construction of our non financial variables model as the days_joined variable has greater variability in terms of data compared to the other two.

```
# multicollinearity test on non financial data

# select non financial correlated variables
correlation_non_fin_var <- Train %>% select(atq, is_annouce, red_annouce, no_annoucement, TFF,
pop_ind,is_mon,is_tue,is_wed,is_thu,is_fri,is_sat)

# correlation matrix
cor(correlation_non_fin_var)
```

```
                            atq is_annouce red_annouce no_annoucement       TFF
atq                    1.000000  0.0843605   0.0436976      0.1093040  0.338662
is_annouce             0.084360  1.0000000   0.2721607      0.7277760  0.072768
red_annouce            0.043698  0.2721607   1.0000000      0.2232720  0.042558
no_annoucement         0.109304  0.7277760   0.2232720      1.0000000  0.040710
TFF                    0.338662  0.0727680   0.0425583      0.0407099  1.000000
coleman_liau          -0.013430 -0.0230916   0.0095778     -0.0270647  0.033211
flesch                 0.025514 -0.0131367   0.0028175     -0.0142399  0.106254
fog_info              -0.055794 -0.0041122   0.0062094     -0.0016299 -0.047365
num_words              0.051410  0.0082365  -0.0204960     -0.0111726  0.018341
bio_complete           0.231862  0.0430806   0.0137354      0.0377542  0.201936
has_url                0.125296  0.0432897   0.0058323      0.0395845  0.073014
hashtags_count        -0.086447 -0.0008502   0.0021443     -0.0006791 -0.047018
has_hashtags          -0.155222 -0.0165677   0.0038319     -0.0140991 -0.076617
contains_company_name -0.033779  0.0216150   0.0221172      0.0035510  0.142158
contains_words         0.040459 -0.0570505  -0.0219036     -0.0393262 -0.103136
custom_content         0.000336  0.0331894  -0.0051374      0.0375625 -0.073381
days_joined            0.148216  0.0226383   0.0020527      0.0185333  0.142188
years_joined           0.148216  0.0226383   0.0020527      0.0185333  0.142188
months_joined          0.148184  0.0226844   0.0021156      0.0185043  0.142632
pop_ind               -0.181242 -0.0031139   0.0068569     -0.0065528 -0.082566
is_mon                 0.001364 -0.0143537  -0.0181555      0.0028744 -0.005480
is_tue                 0.003186  0.0619763   0.0004612      0.0505000 -0.008301
is_wed                -0.001562  0.0773822   0.0116880      0.0455152 -0.008319
is_thu                 0.002114  0.0344638   0.0026271      0.0224441 -0.009388
is_fri                 0.001622 -0.0351769   0.0354713     -0.0287121 -0.007434
is_sat                 0.001006 -0.0843301  -0.0206937     -0.0632632  0.025104
                      coleman_liau    flesch   fog_info  num_words bio_complete
atq                      -0.013430  0.025514 -0.0557935  0.0514098   0.23186201
is_annouce               -0.023092 -0.013137 -0.0041122  0.0082365   0.04308063
red_annouce               0.009578  0.002818  0.0062094 -0.0204960   0.01373544
no_annoucement           -0.027065 -0.014240 -0.0016299 -0.0111726   0.03775420
TFF                       0.033211  0.106254 -0.0473646  0.0183407   0.20193568
coleman_liau              1.000000  0.845450 -0.6288060  0.0555051  -0.23987680
flesch                    0.845450  1.000000 -0.8636924  0.0886831  -0.09731709
fog_info                 -0.628806 -0.863692  1.0000000 -0.0452220   0.04932968
num_words                 0.055505  0.088683 -0.0452220  1.0000000   0.00038193
bio_complete             -0.239877 -0.097317  0.0493297  0.0003819   1.00000000
```

has_url	-0.218247	-0.136684	0.1012728	0.2792596	0.15864096
hashtags_count	-0.550152	-0.561895	0.3862491	-0.0363403	0.28596530
has_hashtags	-0.265200	-0.249212	0.1398090	-0.0970815	0.12916526
contains_company_name	-0.042966	-0.016914	0.0397934	-0.0955569	-0.00421789
contains_words	0.098010	0.117261	-0.1397014	0.1950505	-0.05006877
custom_content	-0.049471	-0.103818	0.0982687	-0.0854487	0.05913076
days_joined	0.020558	0.056873	-0.0789304	0.0526937	0.11534936
years_joined	0.020558	0.056873	-0.0789304	0.0526937	0.11534936
months_joined	0.020502	0.056919	-0.0792814	0.0525856	0.11506048
pop_ind	0.007111	-0.033980	0.1561772	-0.2160284	0.03527170
is_mon	-0.005104	-0.006770	0.0095900	0.0121228	0.00369265
is_tue	-0.002420	-0.006138	0.0034999	0.0076919	-0.00111442
is_wed	-0.003527	-0.008023	0.0086007	0.0124769	0.00331628
is_thu	0.003951	-0.002045	0.0006862	0.0034673	-0.00240149
is_fri	-0.001749	-0.006172	0.0052833	0.0063726	0.00253564
is_sat	0.005409	0.017249	-0.0173454	-0.0296775	-0.00005808

	has_url	hashtags_count	has_hashtags
atq	0.125296	-0.0864470	-0.1552220
is_annouce	0.043290	-0.0008502	-0.0165677
red_annouce	0.005832	0.0021443	0.0038319
no_annoucement	0.039584	-0.0006791	-0.0140991
TFF	0.073014	-0.0470181	-0.0766173
coleman_liau	-0.218247	-0.5501522	-0.2652000
flesch	-0.136684	-0.5618954	-0.2492118
fog_info	0.101273	0.3862491	0.1398090
num_words	0.279260	-0.0363403	-0.0970815
bio_complete	0.158641	0.2859653	0.1291653
has_url	1.000000	0.0972832	0.1620381
hashtags_count	0.097283	1.0000000	0.6003727
has_hashtags	0.162038	0.6003727	1.0000000
contains_company_name	0.059505	0.1428952	0.0896903
contains_words	-0.216535	-0.1213723	-0.1641489
custom_content	0.155128	-0.0547855	0.0599135
days_joined	0.418491	-0.0400359	-0.1412338
years_joined	0.418491	-0.0400359	-0.1412338
months_joined	0.418967	-0.0400364	-0.1410084
pop_ind	0.059980	0.0880436	0.2911415
is_mon	0.004508	0.0023155	0.0045004
is_tue	0.009272	0.0029429	-0.0012972
is_wed	0.010062	0.0038059	0.0012707
is_thu	0.003909	0.0013867	-0.0021378
is_fri	0.005882	0.0003518	-0.0017363
is_sat	-0.031290	0.0022889	-0.0006543

	contains_company_name	contains_words	custom_content
atq	-0.033779	0.040459	0.000336023
is_annouce	0.021615	-0.057051	0.033189365
red_annouce	0.022117	-0.021904	-0.005137428
no_annoucement	0.003551	-0.039326	0.037562544
TFF	0.142158	-0.103136	-0.073381111
coleman_liau	-0.042966	0.098010	-0.049471273
flesch	-0.016914	0.117261	-0.103817648
fog_info	0.039793	-0.139701	0.098268685
num_words	-0.095557	0.195051	-0.085448742
bio_complete	-0.004218	-0.050069	0.059130759

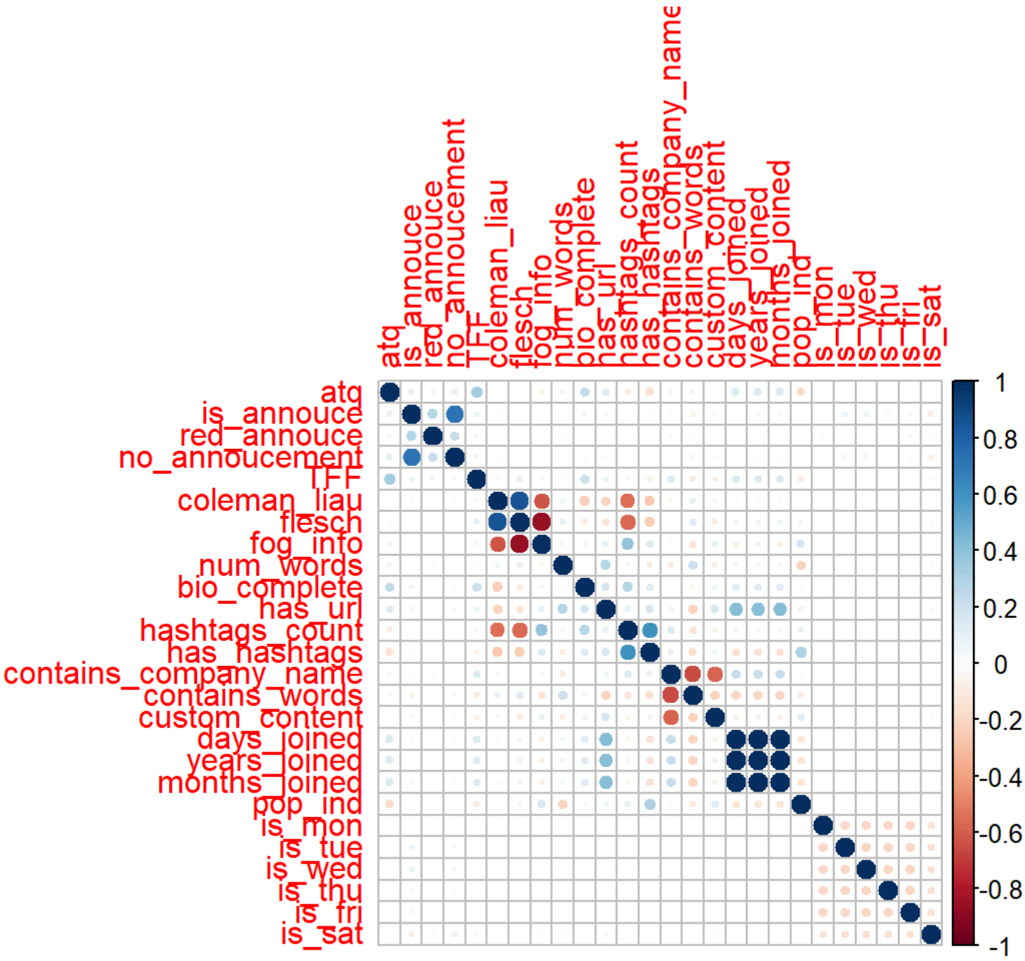
has_url	0.059505	-0.216535	0.155128192
hashtags_count	0.142895	-0.121372	-0.054785492
has_hashtags	0.089690	-0.164149	0.059913543
contains_company_name	1.000000	-0.657283	-0.589322508
contains_words	-0.657283	1.000000	-0.221516251
custom_content	-0.589323	-0.221516	1.000000000
days_joined	0.221526	-0.207438	-0.064265174
years_joined	0.221526	-0.207438	-0.064265174
months_joined	0.221167	-0.207292	-0.063957336
pop_ind	-0.017978	-0.091310	0.121144722
is_mon	0.005799	-0.006994	-0.000006787
is_tue	-0.009766	0.004120	0.008220092
is_wed	0.002340	-0.006962	0.004435851
is_thu	0.001417	-0.006578	0.005218390
is_fri	-0.004491	0.007684	-0.002426031
is_sat	0.007855	0.006388	-0.017011809

	days_joined	years_joined	months_joined	pop_ind
atq	0.1482161	0.1482161	0.148184	-0.1812424
is_annouce	0.0226383	0.0226383	0.022684	-0.0031139
red_annouce	0.0020527	0.0020527	0.002116	0.0068569
no_annoucement	0.0185333	0.0185333	0.018504	-0.0065528
TFF	0.1421880	0.1421880	0.142632	-0.0825660
coleman_liau	0.0205582	0.0205582	0.020502	0.0071108
flesch	0.0568726	0.0568726	0.056919	-0.0339804
fog_info	-0.0789304	-0.0789304	-0.079281	0.1561772
num_words	0.0526937	0.0526937	0.052586	-0.2160284
bio_complete	0.1153494	0.1153494	0.115060	0.0352717
has_url	0.4184905	0.4184905	0.418967	0.0599803
hashtags_count	-0.0400359	-0.0400359	-0.040036	0.0880436
has_hashtags	-0.1412338	-0.1412338	-0.141008	0.2911415
contains_company_name	0.2215255	0.2215255	0.221167	-0.0179779
contains_words	-0.2074384	-0.2074384	-0.207292	-0.0913095
custom_content	-0.0642652	-0.0642652	-0.063957	0.1211447
days_joined	1.0000000	1.0000000	0.999968	-0.1257918
years_joined	1.0000000	1.0000000	0.999968	-0.1257918
months_joined	0.9999680	0.9999680	1.000000	-0.1254456
pop_ind	-0.1257918	-0.1257918	-0.125446	1.0000000
is_mon	-0.0036301	-0.0036301	-0.003784	0.0038728
is_tue	0.0063350	0.0063350	0.005960	-0.0037930
is_wed	0.0009435	0.0009435	0.001005	-0.0005398
is_thu	0.0040083	0.0040083	0.004261	-0.0028371
is_fri	0.0042870	0.0042870	0.004160	-0.0094730
is_sat	-0.0163659	-0.0163659	-0.016062	0.0084642

	is_mon	is_tue	is_wed	is_thu	is_fri
atq	0.001364365	0.0031859	-0.0015622	0.0021141	0.0016215
is_annouce	-0.014353684	0.0619763	0.0773822	0.0344638	-0.0351769
red_annouce	-0.018155523	0.0004612	0.0116880	0.0026271	0.0354713
no_annoucement	0.002874424	0.0505000	0.0455152	0.0224441	-0.0287121
TFF	-0.005480064	-0.0083009	-0.0083185	-0.0093876	-0.0074343
coleman_liau	-0.005104150	-0.0024203	-0.0035268	0.0039512	-0.0017487
flesch	-0.006770158	-0.0061384	-0.0080234	-0.0020448	-0.0061724
fog_info	0.009590001	0.0034999	0.0086007	0.0006862	0.0052833
num_words	0.012122756	0.0076919	0.0124769	0.0034673	0.0063726
bio_complete	0.003692654	-0.0011144	0.0033163	-0.0024015	0.0025356

```
has_url                 0.004507570  0.0092715  0.0100616  0.0039088  0.0058815
hashtags_count          0.002315524  0.0029429  0.0038059  0.0013867  0.0003518
has_hashtags            0.004500390 -0.0012972  0.0012707 -0.0021378 -0.0017363
contains_company_name   0.005799316 -0.0097658  0.0023397  0.0014170 -0.0044915
contains_words         -0.006993614  0.0041195 -0.0069621 -0.0065783  0.0076845
custom_content         -0.000006787  0.0082201  0.0044359  0.0052184 -0.0024260
days_joined            -0.003630061  0.0063350  0.0009435  0.0040083  0.0042870
years_joined           -0.003630061  0.0063350  0.0009435  0.0040083  0.0042870
months_joined          -0.003784189  0.0059599  0.0010049  0.0042605  0.0041604
pop_ind                 0.003872841 -0.0037930 -0.0005398 -0.0028371 -0.0094730
is_mon                  1.000000000 -0.1924850 -0.1936292 -0.1948852 -0.1920840
is_tue                 -0.192484957  1.0000000 -0.2013486 -0.2026548 -0.1997418
is_wed                 -0.193629161 -0.2013486  1.0000000 -0.2038594 -0.2009291
is_thu                 -0.194885245 -0.2026548 -0.2038594  1.0000000 -0.2022326
is_fri                 -0.192083953 -0.1997418 -0.2009291 -0.2022326  1.0000000
is_sat                 -0.136837778 -0.1422931 -0.1431389 -0.1440675 -0.1419967
                            is_sat
atq                     0.00100635
is_annouce             -0.08433008
red_annouce            -0.02069373
no_annoucement         -0.06326317
TFF                     0.02510435
coleman_liau            0.00540888
flesch                  0.01724905
fog_info               -0.01734543
num_words              -0.02967752
bio_complete           -0.00005808
has_url                -0.03128965
hashtags_count          0.00228888
has_hashtags           -0.00065432
contains_company_name   0.00785485
contains_words          0.00638836
custom_content         -0.01701181
days_joined            -0.01636586
years_joined           -0.01636586
months_joined          -0.01606182
pop_ind                 0.00846416
is_mon                 -0.13683778
is_tue                 -0.14229311
is_wed                 -0.14313895
is_thu                 -0.14406750
is_fri                 -0.14199667
is_sat                  1.00000000
```

```
library(corrplot)
correlation <- cor(correlation_non_fin_var, use="pairwise.complete.obs")
corrplot(correlation)
```



5. Financial Model

5.1 Financial Model Construction

In this section, we build a model utilizing financial data exclusively. The 6 models constructed based on financial ratios is as follows:

Model 1: Financial ratios + stock price (daily change rate and daily volatile) + industry dummy (popular industry = 1) + week dummy + log (asset)

We create a linear regression model (`Model1`) to predict the growth rate (`growth_rate`). The formula for the model (`Mod1_formula`) is specified to capture the relationship between the growth rate and the predictor variables. Specifically, the formula encompasses financial ratios, along with indicators for each day of the week, and the logarithm of "atq" as a measure of company size.

```
# Model 1 formula : stock price and financial ratios model

Mod1_formula <- as.formula("growth_rate ~ bm + pe_exi + ps + pcf + dpr + npm + opmbd + opmad +

# run regression Model 1
Model1 <- lm(Mod1_formula,data=Train)
summary(Model1)
```

```
Call:
lm(formula = Mod1_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03475 -0.00050 -0.00016  0.00017  0.04853 

Coefficients:
                    Estimate     Std. Error t value             Pr(>|t|)    
(Intercept)      0.00099129740  0.00018940402    5.23           0.00000017 ***
bm               0.00020943956  0.00006506980    3.22              0.00129 ** 
pe_exi           0.00000016681  0.00000009153    1.82              0.06842 .  
ps              -0.00002530565  0.00001091684   -2.32              0.02046 *  
pcf              0.00000047145  0.00000077664    0.61              0.54383    
dpr              0.00001537373  0.00000730305    2.11              0.03530 *  
npm              0.00157818107  0.00050080720    3.15              0.00163 ** 
opmbd            0.00116122525  0.00035804934    3.24              0.00119 ** 
opmad           -0.00040303160  0.00050955556   -0.79              0.42899    
gpm             -0.00022668388  0.00012086483   -1.88              0.06075 .  
roa             -0.00139662961  0.00043590357   -3.20              0.00136 ** 
roe             -0.00001004940  0.00000945547   -1.06              0.28789    
roce            -0.00012981365  0.00025878989   -0.50              0.61595    
debt_at          0.00072203753  0.00014763756    4.89           0.00000102 ***
de_ratio         0.00000071846  0.00000094921    0.76              0.44913    
intcov          -0.00000000566  0.00000003631   -0.16              0.87606    
cash_ratio       0.00006458046  0.00005719414    1.13              0.25886    
quick_ratio     -0.00003988850  0.00006430089   -0.62              0.53504    
curr_ratio      -0.00001623344  0.00004013120   -0.40              0.68585    
inv_turn         0.00000228300  0.00000021738   10.50 < 0.0000000000000002 ***
at_turn          0.00005591257  0.00004939197    1.13              0.25765    
debt_assets     -0.00056024975  0.00015340403   -3.65              0.00026 ***
ptb              0.00000007935  0.00000008121    0.98              0.32853    
rect_turn       -0.00000393660  0.00000098520   -4.00           0.00006488 ***
sale_nwc        -0.00000002751  0.00000007240   -0.38              0.70392    
divyield        -0.01115519878  0.00125972699   -8.86 < 0.0000000000000002 ***
p_daily_change  -0.00012340065  0.00058028897   -0.21              0.83160    
volatile         0.00272414757  0.00084364896    3.23              0.00125 ** 
pop_ind          0.00033778263  0.00003810149    8.87 < 0.0000000000000002 ***
is_mon           0.00018968282  0.00006445277    2.94              0.00326 ** 
is_tue           0.00017777981  0.00006377328    2.79              0.00532 ** 
is_wed           0.00017917172  0.00006366429    2.81              0.00490 ** 
is_thu           0.00013251282  0.00006357472    2.08              0.03715 *  
is_fri           0.00016543170  0.00006379994    2.59              0.00953 ** 
is_sat          -0.00001432343  0.00007151164   -0.20              0.84125    
log(atq)        -0.00002954885  0.00001081365   -2.73              0.00629 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00163 on 12058 degrees of freedom
Multiple R-squared:  0.0498,	Adjusted R-squared:  0.047 
F-statistic: 18.1 on 35 and 12058 DF,  p-value: <0.0000000000000002
```

Model 2: Financial ratios lag value + stock price (daily change rate and daily volatile) + industry dummy (popular industry = 1) + week dummy + log (asset).

We create a linear regression model (`Model2`) to predict the growth rate (`growth_rate`). The formula for the model (`Mod2_formula`) is specified to capture the relationship between the growth rate and a set of lagged predictor variables encompassing financial metrics, along with indicators for each day of the week, and the logarithm of “atq” as a measure of company size.

```
# Model 2 formula : financial ratios (lag), stock price (lag)
Mod2_formula <- as.formula("growth_rate ~ bm_lag + pe_exi_lag + ps_lag + pcf_lag + dpr_lag +
                          npm_lag+ opmbd_lag + opmad_lag + gpm_lag + roa_lag +roe_lag + roce_l
                          quick_ratio_lag + curr_ratio_lag + inv_turn_lag + at_turn_lag +
                          debt_assets_lag + ptb_lag + rect_turn_lag + sale_nwc_lag +
                          divyield_lag + p_daily_change_lag + volatile_lag +
                          pop_ind + is_mon + is_tue + is_wed + is_thu + is_fri + is_sat +
                          log(atq)")

# run regression Model 2
Model2 <- lm(Mod2_formula,data=Train)
summary(Model2)
```

```
Call:
lm(formula = Mod2_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03469 -0.00051 -0.00016  0.00017  0.04861 

Coefficients:
                      Estimate     Std. Error t value            Pr(>|t|)
(Intercept)        0.00120357014  0.00018937516    6.36       0.000000000215
bm_lag             0.00022125384  0.00006690617    3.31              0.00095
pe_exi_lag         0.00000019283  0.00000007826    2.46              0.01375
ps_lag            -0.00003239246  0.00001140321   -2.84              0.00451
pcf_lag           -0.00000103336  0.00000077146   -1.34              0.18044
dpr_lag            0.00004966276  0.00001104911    4.49       0.000007030782
npm_lag            0.00122650210  0.00047923354    2.56              0.01050
opmbd_lag          0.00116271580  0.00037053664    3.14              0.00171
opmad_lag         -0.00018050930  0.00050638688   -0.36              0.72150
gpm_lag           -0.00014463761  0.00012184062   -1.19              0.23521
roa_lag           -0.00203778475  0.00045739802   -4.46       0.000008458280
roe_lag           -0.00001166243  0.00000946454   -1.23              0.21789
roce_lag           0.00029579451  0.00025996517    1.14              0.25522
debt_at_lag        0.00105044872  0.00015127257    6.94       0.000000000004
de_ratio_lag       0.00000076882  0.00000095456    0.81              0.42059
intcov_lag         0.00000000252  0.00000003293    0.08              0.93907
cash_ratio_lag     0.00005735713  0.00005647762    1.02              0.30985
quick_ratio_lag   -0.00006210663  0.00006536935   -0.95              0.34209
curr_ratio_lag    -0.00001299673  0.00004242373   -0.31              0.75934
inv_turn_lag       0.00000197711  0.00000020259    9.76 < 0.0000000000000002
at_turn_lag        0.00004942583  0.00005155061    0.96              0.33769
debt_assets_lag   -0.00088809916  0.00015654748   -5.67       0.000000014352
ptb_lag            0.00000009139  0.00000008074    1.13              0.25771
rect_turn_lag     -0.00000443952  0.00000102097   -4.35       0.000013830973
```

```
sale_nwc_lag       -0.00000002592  0.00000008633   -0.30              0.76402
divyield_lag       -0.01374826753  0.00141439410   -9.72 < 0.0000000000000002
p_daily_change_lag -0.00026916281  0.00059838587   -0.45              0.65285
volatile_lag        0.00258866040  0.00082678012    3.13              0.00175
pop_ind             0.00034780813  0.00003808504    9.13 < 0.0000000000000002
is_mon              0.00018300397  0.00006442314    2.84              0.00451
is_tue              0.00017469651  0.00006374512    2.74              0.00614
is_wed              0.00017349648  0.00006363996    2.73              0.00642
is_thu              0.00012652383  0.00006354403    1.99              0.04649
is_fri              0.00015722571  0.00006378081    2.47              0.01371
is_sat             -0.00001905166  0.00007148469   -0.27              0.78985
log(atq)           -0.00002817688  0.00001077594   -2.61              0.00894

(Intercept)        ***
bm_lag             ***
pe_exi_lag         *
ps_lag             **
pcf_lag
dpr_lag            ***
npm_lag            *
opmbd_lag          **
opmad_lag
gpm_lag
roa_lag            ***
roe_lag
roce_lag
debt_at_lag        ***
de_ratio_lag
intcov_lag
cash_ratio_lag
quick_ratio_lag
curr_ratio_lag
inv_turn_lag       ***
at_turn_lag
debt_assets_lag    ***
ptb_lag
rect_turn_lag      ***
sale_nwc_lag
divyield_lag       ***
p_daily_change_lag
volatile_lag       **
pop_ind            ***
is_mon             **
is_tue             **
is_wed             **
is_thu             *
is_fri             *
is_sat
log(atq)           **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00163 on 12058 degrees of freedom
```

```
Multiple R-squared:  0.0503,	Adjusted R-squared:  0.0476 
F-statistic: 18.3 on 35 and 12058 DF,  p-value: <0.0000000000000002
```

Model 3: Financial ratios value + stock price (daily change rate and daily volatile) + industry dummy (popular industry = 1) + week dummy + log (asset) + industry fixed effect

This model is used to predict the growth rate (`growth_rate`) while incorporating industry fixed effects. The `update()` function is used to modify the original formula (`Mod1_formula`) by adding a categorical variable representing the industry (`sic`) as a fixed effect. This sic variable is converted into a factor using the `factor()` function.

```
# run regression Model 3 : Model 1 with industry FE
Mod3_formula <- update(Mod1_formula, . ~ . + factor(sic))
Model3 <- lm(Mod3_formula,data=Train)
summary(Model3)
```

```

Call:
lm(formula = Mod3_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03609 -0.00036 -0.00007  0.00018  0.04721 

Coefficients:
                    Estimate    Std. Error t value             Pr(>|t|)    
(Intercept)      0.0008602457  0.0005390637    1.60              0.11056    
bm              -0.0004421752  0.0001335717   -3.31              0.00093 ***
pe_exi          -0.0000000300  0.0000001020   -0.29              0.76830    
ps               0.0000093544  0.0000173804    0.54              0.59044    
pcf              0.0000007190  0.0000007933    0.91              0.36474    
dpr             -0.0000092755  0.0000084064   -1.10              0.26988    
npm              0.0031370410  0.0007355032    4.27    0.000020128696290 ***
opmbd            0.0021090813  0.0011747501    1.80              0.07262 .  
opmad           -0.0016109971  0.0012457667   -1.29              0.19597    
gpm             -0.0009739161  0.0002905770   -3.35              0.00081 ***
roa             -0.0014708261  0.0010432730   -1.41              0.15862    
roe             -0.0000174416  0.0000106225   -1.64              0.10062    
roce            -0.0013271764  0.0005689873   -2.33              0.01969 *  
debt_at          0.0008158083  0.0004288199    1.90              0.05714 .  
de_ratio         0.0000004936  0.0000009432    0.52              0.60078    
intcov           0.0000000891  0.0000000390    2.28              0.02255 *  
cash_ratio      -0.0010865415  0.0001583392   -6.86    0.000000000007119 ***
quick_ratio      0.0026319988  0.0002458311   10.71 < 0.0000000000000002 ***
curr_ratio      -0.0015265212  0.0001522969  -10.02 < 0.0000000000000002 ***
inv_turn         0.0000020878  0.0000003897    5.36    0.000000085870701 ***
at_turn         -0.0008020013  0.0001278927   -6.27    0.000000000371315 ***
debt_assets      0.0002096741  0.0004037895    0.52              0.60359    
ptb              0.0000001364  0.0000000879    1.55              0.12074    
rect_turn        0.0000011292  0.0000026281    0.43              0.66744    
sale_nwc         0.0000000857  0.0000000966    0.89              0.37524    
divyield        -0.0061391740  0.0024956957   -2.46              0.01391 *  
p_daily_change   0.0000080407  0.0005682501    0.01              0.98871    
volatile         0.0023314624  0.0009267399    2.52              0.01189 *  
```

```
pop_ind          0.0000832446  0.0001317979    0.63              0.52766
is_mon           0.0001240753  0.0000618797    2.01              0.04497 *
is_tue           0.0000940221  0.0000612736    1.53              0.12494
is_wed           0.0000888785  0.0000612061    1.45              0.14649
is_thu           0.0000537128  0.0000611160    0.88              0.37949
is_fri           0.0000896585  0.0000613022    1.46              0.14361
is_sat          -0.0000120099  0.0000681641   -0.18              0.86015
log(atq)        -0.0000573868  0.0000208374   -2.75              0.00590 **
factor(sic)1531  0.0010614464  0.0005175982    2.05              0.04032 *
factor(sic)1623  0.0070522251  0.0005858117   12.04 < 0.0000000000000002 ***
factor(sic)2080  0.0006393169  0.0004808969    1.33              0.18373
factor(sic)2300  0.0018296282  0.0005037721    3.63              0.00028 ***
factor(sic)2320  0.0038508208  0.0006162263    6.25    0.000000000426982 ***
factor(sic)2510  0.0032205681  0.0005344948    6.03    0.000000001735927 ***
factor(sic)2621  0.0016669715  0.0004906381    3.40              0.00068 ***
factor(sic)2810  0.0018688442  0.0004127154    4.53    0.000006007097736 ***
factor(sic)2820  0.0008608949  0.0004734431    1.82              0.06903 .
factor(sic)2836  0.0007540075  0.0004811989    1.57              0.11716
factor(sic)2844  0.0015080077  0.0005325769    2.83              0.00464 **
factor(sic)2851  0.0020156451  0.0005420475    3.72              0.00020 ***
factor(sic)2911  0.0021393581  0.0005061620    4.23    0.000023897564561 ***
factor(sic)3490  0.0008352728  0.0004878747    1.71              0.08691 .
factor(sic)3561 -0.0009164828  0.0004746911   -1.93              0.05354 .
factor(sic)3576  0.0006011404  0.0004713003    1.28              0.20216
factor(sic)3674  0.0013087563  0.0004771826    2.74              0.00610 **
factor(sic)3711  0.0000259235  0.0004443623    0.06              0.95348
factor(sic)3724  0.0008843235  0.0004864321    1.82              0.06909 .
factor(sic)3812  0.0014618566  0.0004589261    3.19              0.00145 **
factor(sic)3823  0.0003754989  0.0004757034    0.79              0.42992
factor(sic)3826  0.0011378323  0.0004795408    2.37              0.01767 *
factor(sic)3843  0.0002751386  0.0005228264    0.53              0.59872
factor(sic)3845  0.0011523264  0.0004995936    2.31              0.02110 *
factor(sic)3990  0.0015052285  0.0005378495    2.80              0.00514 **
factor(sic)4213  0.0003103543  0.0005121974    0.61              0.54457
factor(sic)4833 -0.0000325884  0.0004649674   -0.07              0.94413
factor(sic)4841 -0.0001077783  0.0003729221   -0.29              0.77258
factor(sic)4911  0.0000799546  0.0004031063    0.20              0.84278
factor(sic)4931 -0.0001009220  0.0003759306   -0.27              0.78835
factor(sic)5047  0.0022947163  0.0005289257    4.34    0.000014466601752 ***
factor(sic)5065  0.0015066925  0.0004942610    3.05              0.00231 **
factor(sic)5140  0.0033562443  0.0005490950    6.11    0.000000001012408 ***
factor(sic)5150  0.0024610006  0.0005048339    4.87    0.000001102734254 ***
factor(sic)5211  0.0028369489  0.0005388068    5.27    0.000000142422645 ***
factor(sic)5661  0.0049574599  0.0006470455    7.66    0.000000000000020 ***
factor(sic)5734  0.0031020265  0.0005601924    5.54    0.000000031339788 ***
factor(sic)5812  0.0007545778  0.0004349220    1.73              0.08277 .
factor(sic)5912  0.0017063081  0.0005043709    3.38              0.00072 ***
factor(sic)5940  0.0034056123  0.0005542520    6.14    0.000000000827474 ***
factor(sic)5944  0.0053093053  0.0006891348    7.70    0.000000000000014 ***
factor(sic)5961  0.0030907123  0.0005382081    5.74    0.000000009549752 ***
factor(sic)6141 -0.0026286703  0.0005182272   -5.07    0.000000398640244 ***
factor(sic)6200  0.0007477579  0.0006656698    1.12              0.26133
factor(sic)6211 -0.0000052263  0.0005689366   -0.01              0.99267
factor(sic)6282 -0.0004264325  0.0005334485   -0.80              0.42408
```

```
factor(sic)6311  0.0024781902  0.0006194150    4.00    0.000063492720392 ***
factor(sic)6324  0.0012973701  0.0005149416    2.52              0.01177 *
factor(sic)6331  0.0003852145  0.0005287085    0.73              0.46626
factor(sic)6798 -0.0013783516  0.0003409607   -4.04    0.000053202638860 ***
factor(sic)7310 -0.0003458020  0.0004856303   -0.71              0.47644
factor(sic)7323  0.0001159347  0.0005083384    0.23              0.81960
factor(sic)7370  0.0008322779  0.0004419068    1.88              0.05967 .
factor(sic)7372  0.0003732385  0.0004944361    0.75              0.45034
factor(sic)7389 -0.0003472054  0.0004814406   -0.72              0.47081
factor(sic)7900 -0.0003020987  0.0004990926   -0.61              0.54499
factor(sic)7990  0.0011687506  0.0004740163    2.47              0.01369 *
factor(sic)8700  0.0006414839  0.0005282550    1.21              0.22464
factor(sic)8721  0.0000507784  0.0005656626    0.09              0.92847
factor(sic)8731 -0.0004247040  0.0004259101   -1.00              0.31870
factor(sic)8742  0.0001940028  0.0004478536    0.43              0.66489
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00155 on 11997 degrees of freedom
Multiple R-squared:  0.143, Adjusted R-squared:  0.137
F-statistic: 20.9 on 96 and 11997 DF,  p-value: <0.0000000000000002
```

Model 4: Financial ratios lag value + stock price (daily change rate and daily volatile) + industry dummy (popular industry = 1) + week dummy + log (asset) + industry fixed effect

This model is used to predict the growth rate (`growth_rate`) while integrating lagged predictor variables and industry fixed effects. The formula for the model (`Mod4_formula`) is updated using the `update()` function, where the original formula (`Mod2_formula`) is modified to include the industry (`sic`) as a fixed effect. The industry variable is converted into a factor using the `factor()` function.

```
# run regression Model 4 : Model 2 with industry FE
Mod4_formula <- update(Mod2_formula, . ~ . + factor(sic))
Model4 <- lm(Mod4_formula,data=Train)
summary(Model4)
```

```
Call:
lm(formula = Mod4_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03620 -0.00037 -0.00007  0.00018  0.04711 

Coefficients:
                        Estimate    Std. Error t value            Pr(>|t|)    
(Intercept)          0.0015890880  0.0006188367    2.57             0.01024 *  
bm_lag              -0.0004632444  0.0001462050   -3.17             0.00154 ** 
pe_exi_lag           0.0000000164  0.0000000923    0.18             0.85892    
ps_lag               0.0000051047  0.0000185917    0.27             0.78365    
pcf_lag             -0.0000002124  0.0000007965   -0.27             0.78973    
dpr_lag             -0.0000308349  0.0000168886   -1.83             0.06791 .  
npm_lag              0.0032502320  0.0006992424    4.65     0.00000338381349 ***
opmbd_lag            0.0011105735  0.0013166953    0.84             0.39899    
```

```
opmad_lag          -0.0003080087  0.0013167211   -0.23              0.81505
gpm_lag            -0.0009194872  0.0003154941   -2.91              0.00357 **
roa_lag            -0.0025261113  0.0011294959   -2.24              0.02534 *
roe_lag            -0.0000262679  0.0000103832   -2.53              0.01142 *
roce_lag           -0.0010872896  0.0005746123   -1.89              0.05849 .
debt_at_lag         0.0010948101  0.0004506028    2.43              0.01513 *
de_ratio_lag        0.0000006180  0.0000009425    0.66              0.51204
intcov_lag          0.0000000934  0.0000000387    2.41              0.01590 *
cash_ratio_lag     -0.0011459730  0.0001565613   -7.32      0.00000000000026 ***
quick_ratio_lag     0.0025623046  0.0002438983   10.51 < 0.0000000000000002 ***
curr_ratio_lag     -0.0014371282  0.0001536279   -9.35 < 0.0000000000000002 ***
inv_turn_lag        0.0000016491  0.0000003521    4.68      0.00000284155243 ***
at_turn_lag        -0.0008036768  0.0001418969   -5.66      0.00000001514517 ***
debt_assets_lag    -0.0000150637  0.0004287205   -0.04              0.97197
ptb_lag             0.0000002047  0.0000000861    2.38              0.01746 *
rect_turn_lag       0.0000005591  0.0000028096    0.20              0.84228
sale_nwc_lag        0.0000000975  0.0000001028    0.95              0.34258
divyield_lag       -0.0042278494  0.0032056579   -1.32              0.18724
p_daily_change_lag -0.0002969075  0.0005708728   -0.52              0.60301
volatile_lag        0.0020642831  0.0009052554    2.28              0.02261 *
pop_ind             0.0001193662  0.0001316882    0.91              0.36473
is_mon              0.0001287317  0.0000618501    2.08              0.03742 *
is_tue              0.0001007012  0.0000612612    1.64              0.10024
is_wed              0.0000936480  0.0000611837    1.53              0.12589
is_thu              0.0000581053  0.0000610924    0.95              0.34157
is_fri              0.0000937272  0.0000612833    1.53              0.12619
is_sat             -0.0000098169  0.0000681171   -0.14              0.88541
log(atq)           -0.0000556142  0.0000214461   -2.59              0.00952 **
factor(sic)1531     0.0004103402  0.0006018672    0.68              0.49539
factor(sic)1623     0.0065383322  0.0006511850   10.04 < 0.0000000000000002 ***
factor(sic)2080     0.0000294888  0.0005672224    0.05              0.95854
factor(sic)2300     0.0010172384  0.0005812650    1.75              0.08014 .
factor(sic)2320     0.0031346385  0.0006867404    4.56      0.00000505620213 ***
factor(sic)2510     0.0025936676  0.0006153916    4.21      0.00002519834795 ***
factor(sic)2621     0.0011000116  0.0005669467    1.94              0.05237 .
factor(sic)2810     0.0011805739  0.0004913450    2.40              0.01629 *
factor(sic)2820     0.0001412247  0.0005499261    0.26              0.79733
factor(sic)2836     0.0001187418  0.0005706537    0.21              0.83517
factor(sic)2844     0.0008348056  0.0006088244    1.37              0.17035
factor(sic)2851     0.0015107868  0.0006206851    2.43              0.01494 *
factor(sic)2911     0.0014710861  0.0005833076    2.52              0.01168 *
factor(sic)3490     0.0002131320  0.0005732592    0.37              0.71006
factor(sic)3561    -0.0015641966  0.0005549402   -2.82              0.00483 **
factor(sic)3576     0.0000334876  0.0005578217    0.06              0.95213
factor(sic)3674     0.0005830650  0.0005577465    1.05              0.29586
factor(sic)3711    -0.0007380000  0.0005201701   -1.42              0.15599
factor(sic)3724     0.0002605157  0.0005716610    0.46              0.64860
factor(sic)3812     0.0007764773  0.0005455087    1.42              0.15465
factor(sic)3823    -0.0003183568  0.0005558011   -0.57              0.56680
factor(sic)3826     0.0005698197  0.0005664681    1.01              0.31448
factor(sic)3843    -0.0002133623  0.0006096666   -0.35              0.72637
factor(sic)3845     0.0005364047  0.0005794030    0.93              0.35457
factor(sic)3990     0.0009067818  0.0006155962    1.47              0.14077
factor(sic)4213    -0.0002142254  0.0005822343   -0.37              0.71293
```

```
factor(sic)4833    -0.0007487184  0.0005488184   -1.36              0.17252    
factor(sic)4841    -0.0005336037  0.0004495316   -1.19              0.23524    
factor(sic)4911    -0.0005339714  0.0004789100   -1.11              0.26488    
factor(sic)4931    -0.0006570626  0.0004492933   -1.46              0.14365    
factor(sic)5047     0.0015507364  0.0005980090    2.59              0.00952 ** 
factor(sic)5065     0.0007998854  0.0005783414    1.38              0.16667    
factor(sic)5140     0.0028269952  0.0006283622    4.50     0.00000689188553 ***
factor(sic)5150     0.0018148607  0.0005930322    3.06              0.00222 ** 
factor(sic)5211     0.0022191219  0.0006111946    3.63              0.00028 ***
factor(sic)5661     0.0043434675  0.0007105468    6.11     0.00000000100905 ***
factor(sic)5734     0.0024557288  0.0006247752    3.93     0.00008521699131 ***
factor(sic)5812     0.0002530822  0.0005159076    0.49              0.62375    
factor(sic)5912     0.0010618025  0.0005882311    1.81              0.07109 .  
factor(sic)5940     0.0028412162  0.0006391687    4.45     0.00000886090133 ***
factor(sic)5944     0.0044781479  0.0007522349    5.95     0.00000000270427 ***
factor(sic)5961     0.0024555178  0.0006149290    3.99     0.00006558364871 ***
factor(sic)6141    -0.0034178371  0.0006130471   -5.58     0.00000002526427 ***
factor(sic)6200     0.0006094240  0.0006892890    0.88              0.37664    
factor(sic)6211    -0.0006544987  0.0006523197   -1.00              0.31572    
factor(sic)6282    -0.0010416622  0.0006037270   -1.73              0.08448 .  
factor(sic)6311     0.0019566257  0.0007082856    2.76              0.00575 ** 
factor(sic)6324     0.0007589432  0.0006027044    1.26              0.20797    
factor(sic)6331    -0.0002851980  0.0006181561   -0.46              0.64454    
factor(sic)6798    -0.0018672103  0.0003937988   -4.74     0.00000214547443 ***
factor(sic)7310    -0.0011536582  0.0005531814   -2.09              0.03705 *  
factor(sic)7323    -0.0004487190  0.0005968441   -0.75              0.45217    
factor(sic)7370     0.0003955656  0.0005316781    0.74              0.45689    
factor(sic)7372    -0.0002224203  0.0005865445   -0.38              0.70454    
factor(sic)7389    -0.0009801915  0.0005729299   -1.71              0.08714 .  
factor(sic)7900    -0.0006906373  0.0005705196   -1.21              0.22610    
factor(sic)7990     0.0006168698  0.0005566582    1.11              0.26781    
factor(sic)8700     0.0000932744  0.0006152956    0.15              0.87951    
factor(sic)8721    -0.0006241204  0.0006417312   -0.97              0.33079    
factor(sic)8731    -0.0010163610  0.0005031592   -2.02              0.04341 *  
factor(sic)8742    -0.0002711206  0.0005209207   -0.52              0.60275    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00155 on 11997 degrees of freedom
Multiple R-squared:  0.145, Adjusted R-squared:  0.138 
F-statistic: 21.1 on 96 and 11997 DF,  p-value: <0.0000000000000002
```

Model 5: Financial ratios value + stock price (daily change rate and daily volatile) + industry dummy (popular industry = 1) + week dummy + log (asset) + industry fixed effect +firm fixed effect

This model is used to forecast the growth rate (`growth_rate`) while incorporating both lagged predictor variables, firm fixed effects (`gvkey`), and industry fixed effects (`sic`). The formula for the model (`Mod5_formula`) is updated using the `update()` function, where the original formula (`Mod2_formula`) is modified to include firm fixed effects (`gvkey`) and industry fixed effects (`sic`). Both `gvkey` and `sic` are converted into factors using the `factor()` function.

```
# run regression Model 5 : Model 2 with company FE and industry FE
Mod5_formula <- update(Mod2_formula, . ~ . + factor(gvkey) + factor(sic))
```

```r
Model5 <- lm(Mod5_formula,data=Train)
summary(Model5)
```

```
Call:
lm(formula = Mod5_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03651 -0.00030 -0.00006  0.00015  0.04680 

Coefficients: (62 not defined because of singularities)
                          Estimate     Std. Error t value             Pr(>|t|)
(Intercept)          0.00452763266  0.00116923392    3.87              0.00011
bm_lag               0.00085493766  0.00037086970    2.31              0.02117
pe_exi_lag           0.00000000646  0.00000011559    0.06              0.95545
ps_lag              -0.00003141415  0.00003020849   -1.04              0.29840
pcf_lag             -0.00000107286  0.00000093445   -1.15              0.25094
dpr_lag             -0.00001885346  0.00002000577   -0.94              0.34601
npm_lag              0.00166005886  0.00103189369    1.61              0.10770
opmbd_lag           -0.00743509571  0.00220309101   -3.37              0.00074
opmad_lag            0.00491773297  0.00171489466    2.87              0.00414
gpm_lag             -0.00045163437  0.00085332679   -0.53              0.59663
roa_lag             -0.00100297024  0.00261248168   -0.38              0.70105
roe_lag             -0.00001189215  0.00001132547   -1.05              0.29372
roce_lag            -0.00008191210  0.00095832944   -0.09              0.93189
debt_at_lag         -0.00168732092  0.00111223421   -1.52              0.12928
de_ratio_lag        -0.00000004077  0.00000094365   -0.04              0.96554
intcov_lag           0.00000001289  0.00000004948    0.26              0.79444
cash_ratio_lag      -0.00057624270  0.00023695385   -2.43              0.01504
quick_ratio_lag      0.00013312675  0.00047589708    0.28              0.77968
curr_ratio_lag       0.00050187678  0.00038995733    1.29              0.19812
inv_turn_lag         0.00000063374  0.00000037370    1.70              0.08994
at_turn_lag         -0.00009569241  0.00040920857   -0.23              0.81511
debt_assets_lag      0.00221533934  0.00099428261    2.23              0.02589
ptb_lag              0.00000009987  0.00000009237    1.08              0.27960
rect_turn_lag        0.00000347927  0.00000452747    0.77              0.44222
sale_nwc_lag        -0.00000000536  0.00000010779   -0.05              0.96032
divyield_lag         0.00389416894  0.00551575465    0.71              0.48020
p_daily_change_lag  -0.00029134686  0.00056241143   -0.52              0.60445
volatile_lag         0.00161099110  0.00090754127    1.78              0.07590
pop_ind             -0.00520934063  0.00055626963   -9.36 < 0.0000000000000002
is_mon               0.00012938928  0.00006097105    2.12              0.03385
is_tue               0.00010201623  0.00006037679    1.69              0.09112
is_wed               0.00009021586  0.00006030108    1.50              0.13466
is_thu               0.00005824061  0.00006020852    0.97              0.33341
is_fri               0.00009150897  0.00006041086    1.51              0.12986
is_sat              -0.00000539891  0.00006706163   -0.08              0.93584
log(atq)            -0.00000074506  0.00004496642   -0.02              0.98678
factor(gvkey)1447   -0.00671564140  0.00093086275   -7.21   0.00000000000057428
factor(gvkey)1487   -0.00732270674  0.00089306254   -8.20   0.00000000000000027
factor(gvkey)1659   -0.00617807149  0.00052598229  -11.75 < 0.0000000000000002
factor(gvkey)1878    0.00075187392  0.00069809300    1.08              0.28148
factor(gvkey)1919   -0.00610477601  0.00050735296  -12.03 < 0.0000000000000002
```

factor(gvkey)1920	-0.00031452025	0.00036724222	-0.86	0.39177
factor(gvkey)3170	0.00086305471	0.00068653458	1.26	0.20874
factor(gvkey)3226	-0.00430599543	0.00087424709	-4.93	0.00000085323186833
factor(gvkey)3336	0.00156958076	0.00057715956	2.72	0.00655
factor(gvkey)3708	0.00374153557	0.00068733828	5.44	0.00000005326968506
factor(gvkey)3814	-0.00480914882	0.00081234963	-5.92	0.00000000330671091
factor(gvkey)4060	-0.00017037568	0.00053448616	-0.32	0.74991
factor(gvkey)4201	0.00038965239	0.00070126882	0.56	0.57847
factor(gvkey)4321	-0.00487176865	0.00075324001	-6.47	0.00000000010334434
factor(gvkey)4839	-0.00583650877	0.00085305592	-6.84	0.00000000000819674
factor(gvkey)4988	-0.00389989738	0.00091227580	-4.27	0.00001926951889275
factor(gvkey)5492	-0.00395808974	0.00072051435	-5.49	0.00000004022740253
factor(gvkey)5742	-0.00475033520	0.00081801233	-5.81	0.00000000651529940
factor(gvkey)6435	-0.00448311473	0.00072170622	-6.21	0.00000000054108333
factor(gvkey)6788	-0.00236814805	0.00126581759	-1.87	0.06139
factor(gvkey)6829	-0.00524093769	0.00058554180	-8.95	< 0.0000000000000002
factor(gvkey)7163	-0.00387922537	0.00096834161	-4.01	0.00006211588756033
factor(gvkey)8113	-0.00460905408	0.00086789952	-5.31	0.00000011123030333
factor(gvkey)8358	-0.00535140672	0.00061623490	-8.68	< 0.0000000000000002
factor(gvkey)8402	-0.00477469986	0.00100800455	-4.74	0.00000219630512991
factor(gvkey)8463	-0.00555693776	0.00071966249	-7.72	0.00000000000001240
factor(gvkey)8479	-0.00480003870	0.00078756386	-6.09	0.00000000112948984
factor(gvkey)8549	-0.00275928592	0.00107149701	-2.58	0.01003
factor(gvkey)9599	0.00184225954	0.00057354544	3.21	0.00132
factor(gvkey)9667	-0.00521890537	0.00062231001	-8.39	< 0.0000000000000002
factor(gvkey)10247	-0.00505583009	0.00066926584	-7.55	0.00000000000004517
factor(gvkey)10983	-0.00521989931	0.00071545400	-7.30	0.00000000000031533
factor(gvkey)11115	0.00100918176	0.00054352610	1.86	0.06337
factor(gvkey)11264	-0.00584284269	0.00052108360	-11.21	< 0.0000000000000002
factor(gvkey)11304	-0.00517284643	0.00083688171	-6.18	0.00000000065727661
factor(gvkey)11584	-0.00597844202	0.00082621516	-7.24	0.00000000000049054
factor(gvkey)11669	-0.00615565881	0.00101490576	-6.07	0.00000000135721721
factor(gvkey)11770	-0.00121264442	0.00071539169	-1.70	0.09009
factor(gvkey)12441	-0.00532624146	0.00072805491	-7.32	0.00000000000027235
factor(gvkey)13092	0.00051213628	0.00066827925	0.77	0.44348
factor(gvkey)14412	-0.00488655565	0.00079243277	-6.17	0.00000000072056003
factor(gvkey)17035	-0.00665354255	0.00067336336	-9.88	< 0.0000000000000002
factor(gvkey)23225	-0.00308894694	0.00086991605	-3.55	0.00039
factor(gvkey)23252	-0.00559167258	0.00074060563	-7.55	0.00000000000004663
factor(gvkey)24468	-0.00300816830	0.00094618264	-3.18	0.00148
factor(gvkey)24800	0.00106232032	0.00074907459	1.42	0.15617
factor(gvkey)25124	-0.00209017943	0.00085069472	-2.46	0.01402
factor(gvkey)25340	-0.00562052121	0.00083788656	-6.71	0.00000000002062298
factor(gvkey)26011	-0.00511004618	0.00079753791	-6.41	0.00000000015369655
factor(gvkey)27794	-0.00164760829	0.00066394066	-2.48	0.01309
factor(gvkey)27914	-0.00538501617	0.00048525056	-11.10	< 0.0000000000000002
factor(gvkey)28034	-0.00664660247	0.00079303793	-8.38	< 0.0000000000000002
factor(gvkey)28216	-0.00023688593	0.00080747758	-0.29	0.76925
factor(gvkey)28303	-0.00534622044	0.00064297217	-8.31	< 0.0000000000000002
factor(gvkey)28629	0.00163545592	0.00082385450	1.99	0.04715
factor(gvkey)28924	-0.00671679158	0.00061130647	-10.99	< 0.0000000000000002
factor(gvkey)29011	-0.00295467962	0.00106696063	-2.77	0.00563
factor(gvkey)29241	0.00096733036	0.00070303400	1.38	0.16887
factor(gvkey)29612	-0.00589541756	0.00084685473	-6.96	0.00000000000354142

```
factor(gvkey)29710  -0.00566538118  0.00055559544  -10.20 < 0.0000000000000002
factor(gvkey)29751  -0.00256590291  0.00081986231   -3.13             0.00175
factor(gvkey)30138  -0.00596190312  0.00050041160  -11.91 < 0.0000000000000002
factor(gvkey)61494  -0.00003675263  0.00040293917   -0.09             0.92733
factor(gvkey)65772  -0.00390978464  0.00071158660   -5.49  0.00000003999207634
factor(gvkey)116504 -0.00519439007  0.00063338237   -8.20  0.00000000000000026
factor(gvkey)118502 -0.00408213232  0.00074357475   -5.49  0.00000004104173154
factor(gvkey)121718  0.00104971613  0.00068715251    1.53             0.12663
factor(gvkey)122841  0.00058919651  0.00071702850    0.82             0.41125
factor(gvkey)126554 -0.00464473337  0.00074935113   -6.20  0.00000000058937433
factor(gvkey)136725  0.00390779815  0.00104552825    3.74             0.00019
factor(gvkey)137131 -0.00475178806  0.00075877479   -6.26  0.00000000039195608
factor(gvkey)141384 -0.00428412743  0.00087772101   -4.88  0.00000106928462886
factor(gvkey)145049 -0.00112316882  0.00055615362   -2.02             0.04345
factor(gvkey)148470 -0.00009835606  0.00047165499   -0.21             0.83482
factor(gvkey)150139 -0.00630839309  0.00068008812   -9.28 < 0.0000000000000002
factor(gvkey)156617 -0.00600195438  0.00088878679   -6.75  0.00000000001515422
factor(gvkey)160211 -0.00053134159  0.00227732953   -0.23             0.81552
factor(gvkey)164416 -0.00507063617  0.00061552927   -8.24 < 0.0000000000000002
factor(gvkey)164664 -0.00023430446  0.00056463511   -0.41             0.67817
factor(gvkey)165746 -0.00585428073  0.00059807345   -9.79 < 0.0000000000000002
factor(gvkey)170841 -0.00515088907  0.00056370481   -9.14 < 0.0000000000000002
factor(gvkey)174317 -0.00551809747  0.00067358528   -8.19  0.00000000000000028
factor(gvkey)177376 -0.00646849302  0.00094286712   -6.86  0.00000000000720252
factor(gvkey)178493 -0.00611899348  0.00076621308   -7.99  0.00000000000000152
factor(gvkey)179666  0.00087759172  0.00079108501    1.11             0.26730
factor(gvkey)180402 -0.00573402734  0.00059774697   -9.59 < 0.0000000000000002
factor(gvkey)180646             NA             NA      NA                  NA
factor(gvkey)184500  0.00173900260  0.00152058403    1.14             0.25280
factor(sic)1531                 NA             NA      NA                  NA
factor(sic)1623                 NA             NA      NA                  NA
factor(sic)2080                 NA             NA      NA                  NA
factor(sic)2300                 NA             NA      NA                  NA
factor(sic)2320                 NA             NA      NA                  NA
factor(sic)2510                 NA             NA      NA                  NA
factor(sic)2621                 NA             NA      NA                  NA
factor(sic)2810                 NA             NA      NA                  NA
factor(sic)2820                 NA             NA      NA                  NA
factor(sic)2836                 NA             NA      NA                  NA
factor(sic)2844                 NA             NA      NA                  NA
factor(sic)2851                 NA             NA      NA                  NA
factor(sic)2911                 NA             NA      NA                  NA
factor(sic)3490                 NA             NA      NA                  NA
factor(sic)3561                 NA             NA      NA                  NA
factor(sic)3576                 NA             NA      NA                  NA
factor(sic)3674                 NA             NA      NA                  NA
factor(sic)3711                 NA             NA      NA                  NA
factor(sic)3724                 NA             NA      NA                  NA
factor(sic)3812                 NA             NA      NA                  NA
factor(sic)3823                 NA             NA      NA                  NA
factor(sic)3826                 NA             NA      NA                  NA
factor(sic)3843                 NA             NA      NA                  NA
factor(sic)3845                 NA             NA      NA                  NA
factor(sic)3990                 NA             NA      NA                  NA
```

```
factor(sic)4213                  NA             NA      NA                  NA
factor(sic)4833                  NA             NA      NA                  NA
factor(sic)4841                  NA             NA      NA                  NA
factor(sic)4911                  NA             NA      NA                  NA
factor(sic)4931                  NA             NA      NA                  NA
factor(sic)5047                  NA             NA      NA                  NA
factor(sic)5065                  NA             NA      NA                  NA
factor(sic)5140                  NA             NA      NA                  NA
factor(sic)5150                  NA             NA      NA                  NA
factor(sic)5211                  NA             NA      NA                  NA
factor(sic)5661                  NA             NA      NA                  NA
factor(sic)5734                  NA             NA      NA                  NA
factor(sic)5812                  NA             NA      NA                  NA
factor(sic)5912                  NA             NA      NA                  NA
factor(sic)5940                  NA             NA      NA                  NA
factor(sic)5944                  NA             NA      NA                  NA
factor(sic)5961                  NA             NA      NA                  NA
factor(sic)6141                  NA             NA      NA                  NA
factor(sic)6200                  NA             NA      NA                  NA
factor(sic)6211                  NA             NA      NA                  NA
factor(sic)6282                  NA             NA      NA                  NA
factor(sic)6311                  NA             NA      NA                  NA
factor(sic)6324                  NA             NA      NA                  NA
factor(sic)6331                  NA             NA      NA                  NA
factor(sic)6798                  NA             NA      NA                  NA
factor(sic)7310                  NA             NA      NA                  NA
factor(sic)7323                  NA             NA      NA                  NA
factor(sic)7370                  NA             NA      NA                  NA
factor(sic)7372                  NA             NA      NA                  NA
factor(sic)7389                  NA             NA      NA                  NA
factor(sic)7900                  NA             NA      NA                  NA
factor(sic)7990                  NA             NA      NA                  NA
factor(sic)8700                  NA             NA      NA                  NA
factor(sic)8721                  NA             NA      NA                  NA
factor(sic)8731                  NA             NA      NA                  NA
factor(sic)8742                  NA             NA      NA                  NA

(Intercept)         ***
bm_lag              *
pe_exi_lag
ps_lag
pcf_lag
dpr_lag
npm_lag
opmbd_lag           ***
opmad_lag           **
gpm_lag
roa_lag
roe_lag
roce_lag
debt_at_lag
de_ratio_lag
intcov_lag
cash_ratio_lag      *
```

quick_ratio_lag	
curr_ratio_lag	
inv_turn_lag	.
at_turn_lag	
debt_assets_lag	*
ptb_lag	
rect_turn_lag	
sale_nwc_lag	
divyield_lag	
p_daily_change_lag	
volatile_lag	.
pop_ind	***
is_mon	*
is_tue	.
is_wed	
is_thu	
is_fri	
is_sat	
log(atq)	
factor(gvkey)1447	***
factor(gvkey)1487	***
factor(gvkey)1659	***
factor(gvkey)1878	
factor(gvkey)1919	***
factor(gvkey)1920	
factor(gvkey)3170	
factor(gvkey)3226	***
factor(gvkey)3336	**
factor(gvkey)3708	***
factor(gvkey)3814	***
factor(gvkey)4060	
factor(gvkey)4201	
factor(gvkey)4321	***
factor(gvkey)4839	***
factor(gvkey)4988	***
factor(gvkey)5492	***
factor(gvkey)5742	***
factor(gvkey)6435	***
factor(gvkey)6788	.
factor(gvkey)6829	***
factor(gvkey)7163	***
factor(gvkey)8113	***
factor(gvkey)8358	***
factor(gvkey)8402	***
factor(gvkey)8463	***
factor(gvkey)8479	***
factor(gvkey)8549	*
factor(gvkey)9599	**
factor(gvkey)9667	***
factor(gvkey)10247	***
factor(gvkey)10983	***
factor(gvkey)11115	.
factor(gvkey)11264	***
factor(gvkey)11304	***

```
factor(gvkey)11584  ***
factor(gvkey)11669  ***
factor(gvkey)11770  .
factor(gvkey)12441  ***
factor(gvkey)13092
factor(gvkey)14412  ***
factor(gvkey)17035  ***
factor(gvkey)23225  ***
factor(gvkey)23252  ***
factor(gvkey)24468  **
factor(gvkey)24800
factor(gvkey)25124  *
factor(gvkey)25340  ***
factor(gvkey)26011  ***
factor(gvkey)27794  *
factor(gvkey)27914  ***
factor(gvkey)28034  ***
factor(gvkey)28216
factor(gvkey)28303  ***
factor(gvkey)28629  *
factor(gvkey)28924  ***
factor(gvkey)29011  **
factor(gvkey)29241
factor(gvkey)29612  ***
factor(gvkey)29710  ***
factor(gvkey)29751  **
factor(gvkey)30138  ***
factor(gvkey)61494
factor(gvkey)65772  ***
factor(gvkey)116504 ***
factor(gvkey)118502 ***
factor(gvkey)121718
factor(gvkey)122841
factor(gvkey)126554 ***
factor(gvkey)136725 ***
factor(gvkey)137131 ***
factor(gvkey)141384 ***
factor(gvkey)145049 *
factor(gvkey)148470
factor(gvkey)150139 ***
factor(gvkey)156617 ***
factor(gvkey)160211
factor(gvkey)164416 ***
factor(gvkey)164664
factor(gvkey)165746 ***
factor(gvkey)170841 ***
factor(gvkey)174317 ***
factor(gvkey)177376 ***
factor(gvkey)178493 ***
factor(gvkey)179666
factor(gvkey)180402 ***
factor(gvkey)180646
factor(gvkey)184500
factor(sic)1531
```

```
factor(sic)1623
factor(sic)2080
factor(sic)2300
factor(sic)2320
factor(sic)2510
factor(sic)2621
factor(sic)2810
factor(sic)2820
factor(sic)2836
factor(sic)2844
factor(sic)2851
factor(sic)2911
factor(sic)3490
factor(sic)3561
factor(sic)3576
factor(sic)3674
factor(sic)3711
factor(sic)3724
factor(sic)3812
factor(sic)3823
factor(sic)3826
factor(sic)3843
factor(sic)3845
factor(sic)3990
factor(sic)4213
factor(sic)4833
factor(sic)4841
factor(sic)4911
factor(sic)4931
factor(sic)5047
factor(sic)5065
factor(sic)5140
factor(sic)5150
factor(sic)5211
factor(sic)5661
factor(sic)5734
factor(sic)5812
factor(sic)5912
factor(sic)5940
factor(sic)5944
factor(sic)5961
factor(sic)6141
factor(sic)6200
factor(sic)6211
factor(sic)6282
factor(sic)6311
factor(sic)6324
factor(sic)6331
factor(sic)6798
factor(sic)7310
factor(sic)7323
factor(sic)7370
factor(sic)7372
factor(sic)7389
```

```
factor(sic)7900
factor(sic)7990
factor(sic)8700
factor(sic)8721
factor(sic)8731
factor(sic)8742
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00152 on 11971 degrees of freedom
Multiple R-squared:  0.173, Adjusted R-squared:  0.165
F-statistic: 20.6 on 122 and 11971 DF,  p-value: <0.0000000000000002
```

Model 6: Financial ratios lag value + stock price (daily change rate and daily volatile) + industry dummy (popular industry = 1) + week dummy + log (asset) + industry fixed effect +firm fixed effect

This model is used to predict the growth rate (`growth_rate`) while incorporating industry fixed effects (`sic`) and firm fixed effects (`gvkey`). The formula for the model (`Mod6_formula`) is updated using the `update()` function, where the original formula (`Mod1_formula`) is modified to include both firm and industry fixed effects. Both `gvkey` and `sic` are converted into factors using the `factor()` function.

```
# run regression Model 6 : Model 1 with company FE and industry FE
Mod6_formula <- update(Mod1_formula, . ~ . + factor(gvkey) + factor(sic))
Model6 <- lm(Mod6_formula,data=Train)
summary(Model6)
```

```
Call:
lm(formula = Mod6_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.03671 -0.00031 -0.00006  0.00016  0.04659

Coefficients: (62 not defined because of singularities)
                      Estimate    Std. Error t value              Pr(>|t|)
(Intercept)         0.0035672945  0.0009754092    3.66               0.00026
bm                  0.0014408866  0.0003253341    4.43   0.00000955353169446
pe_exi             -0.0000000213  0.0000001177   -0.18               0.85642
ps                 -0.0000503214  0.0000303301   -1.66               0.09712
pcf                -0.0000001370  0.0000008555   -0.16               0.87278
dpr                 0.0000019816  0.0000088131    0.22               0.82210
npm                 0.0012562163  0.0009750375    1.29               0.19764
opmbd              -0.0050791682  0.0017350288   -2.93               0.00342
opmad               0.0021831339  0.0015218565    1.43               0.15145
gpm                -0.0014996273  0.0007731375   -1.94               0.05244
roa                 0.0014671344  0.0019956029    0.74               0.46224
roe                -0.0000020906  0.0000123424   -0.17               0.86550
roce               -0.0006860936  0.0009366868   -0.73               0.46390
debt_at            -0.0036091814  0.0009866014   -3.66               0.00026
de_ratio            0.0000000193  0.0000009370    0.02               0.98354
intcov              0.0000000346  0.0000000464    0.75               0.45600
```

cash_ratio	-0.0003748254	0.0002359929	-1.59	0.11225
quick_ratio	0.0002443913	0.0003881562	0.63	0.52895
curr_ratio	0.0004019197	0.0002996748	1.34	0.17989
inv_turn	0.0000006977	0.0000004098	1.70	0.08869
at_turn	-0.0001742061	0.0002651174	-0.66	0.51114
debt_assets	0.0038764513	0.0008634498	4.49	0.00000720602171180
ptb	0.0000000236	0.0000000990	0.24	0.81121
rect_turn	0.0000005103	0.0000038474	0.13	0.89449
sale_nwc	0.0000000399	0.0000000991	0.40	0.68728
divyield	-0.0007860379	0.0036235840	-0.22	0.82827
p_daily_change	0.0001171311	0.0005636690	0.21	0.83539
volatile	0.0020679656	0.0009336653	2.21	0.02679
pop_ind	-0.0045604931	0.0005240149	-8.70	< 0.0000000000000002
is_mon	0.0001294522	0.0000609232	2.12	0.03362
is_tue	0.0001043972	0.0000603152	1.73	0.08350
is_wed	0.0000927066	0.0000602530	1.54	0.12392
is_thu	0.0000625261	0.0000601571	1.04	0.29865
is_fri	0.0000952836	0.0000603507	1.58	0.11440
is_sat	-0.0000035097	0.0000670164	-0.05	0.95823
log(atq)	0.0000079592	0.0000423442	0.19	0.85091
factor(gvkey)1447	-0.0065152299	0.0007520972	-8.66	< 0.0000000000000002
factor(gvkey)1487	-0.0084974472	0.0007926708	-10.72	< 0.0000000000000002
factor(gvkey)1659	-0.0061137471	0.0005015948	-12.19	< 0.0000000000000002
factor(gvkey)1878	0.0008281179	0.0006083855	1.36	0.17348
factor(gvkey)1919	-0.0061402192	0.0004635898	-13.24	< 0.0000000000000002
factor(gvkey)1920	-0.0001143527	0.0003293730	-0.35	0.72846
factor(gvkey)3170	0.0010143179	0.0005435355	1.87	0.06205
factor(gvkey)3226	-0.0042719509	0.0006767273	-6.31	0.00000000028397014
factor(gvkey)3336	0.0008558095	0.0004120492	2.08	0.03783
factor(gvkey)3708	0.0037385966	0.0005267929	7.10	0.00000000000134788
factor(gvkey)3814	-0.0046836368	0.0006859639	-6.83	0.00000000000903821
factor(gvkey)4060	-0.0005392684	0.0004353697	-1.24	0.21550
factor(gvkey)4201	0.0001811296	0.0005698713	0.32	0.75061
factor(gvkey)4321	-0.0041960453	0.0005928287	-7.08	0.00000000000154410
factor(gvkey)4839	-0.0054501903	0.0006371980	-8.55	< 0.0000000000000002
factor(gvkey)4988	-0.0030009612	0.0007063486	-4.25	0.00002167806907049
factor(gvkey)5492	-0.0033310405	0.0005903121	-5.64	0.00000001710585003
factor(gvkey)5742	-0.0050699194	0.0006604102	-7.68	0.00000000000001756
factor(gvkey)6435	-0.0039451877	0.0005708901	-6.91	0.00000000000507119
factor(gvkey)6788	-0.0033406597	0.0008853952	-3.77	0.00016
factor(gvkey)6829	-0.0046416792	0.0005580987	-8.32	< 0.0000000000000002
factor(gvkey)7163	-0.0029917921	0.0007803037	-3.83	0.00013
factor(gvkey)8113	-0.0044839806	0.0007203815	-6.22	0.00000000049939273
factor(gvkey)8358	-0.0050077678	0.0005140682	-9.74	< 0.0000000000000002
factor(gvkey)8402	-0.0040373312	0.0009044880	-4.46	0.00000813059652120
factor(gvkey)8463	-0.0051307586	0.0005462700	-9.39	< 0.0000000000000002
factor(gvkey)8479	-0.0040279033	0.0006115245	-6.59	0.00000000004686627
factor(gvkey)8549	-0.0038272721	0.0008189025	-4.67	0.00000299105628888
factor(gvkey)9599	0.0020430656	0.0005092082	4.01	0.00006051114790887
factor(gvkey)9667	-0.0046921146	0.0005312835	-8.83	< 0.0000000000000002
factor(gvkey)10247	-0.0045169988	0.0006008471	-7.52	0.00000000000005971
factor(gvkey)10983	-0.0048823487	0.0005784672	-8.44	< 0.0000000000000002
factor(gvkey)11115	0.0008549593	0.0004604988	1.86	0.06339
factor(gvkey)11264	-0.0053859822	0.0004847902	-11.11	< 0.0000000000000002

factor(gvkey)11304	-0.0051102854	0.0006725864	-7.60	0.00000000000003232
factor(gvkey)11584	-0.0055315693	0.0007733385	-7.15	0.00000000000089950
factor(gvkey)11669	-0.0051888826	0.0009142796	-5.68	0.00000001415956451
factor(gvkey)11770	-0.0024974697	0.0006343533	-3.94	0.00008296248237630
factor(gvkey)12441	-0.0048060874	0.0006116865	-7.86	0.00000000000000427
factor(gvkey)13092	0.0004589171	0.0005044574	0.91	0.36299
factor(gvkey)14412	-0.0042323363	0.0006407163	-6.61	0.00000000004125843
factor(gvkey)17035	-0.0068316999	0.0005926474	-11.53	< 0.0000000000000002
factor(gvkey)23225	-0.0028011229	0.0007173603	-3.90	0.00009483492255900
factor(gvkey)23252	-0.0045621028	0.0006895122	-6.62	0.00000000003836836
factor(gvkey)24468	-0.0018765950	0.0007818519	-2.40	0.01640
factor(gvkey)24800	0.0011306731	0.0005931221	1.91	0.05663
factor(gvkey)25124	-0.0013527608	0.0006763965	-2.00	0.04553
factor(gvkey)25340	-0.0048744931	0.0006673716	-7.30	0.00000000000029701
factor(gvkey)26011	-0.0043706676	0.0007254164	-6.03	0.00000000174028733
factor(gvkey)27794	-0.0021106296	0.0006398570	-3.30	0.00097
factor(gvkey)27914	-0.0054254387	0.0004680824	-11.59	< 0.0000000000000002
factor(gvkey)28034	-0.0071389531	0.0006893400	-10.36	< 0.0000000000000002
factor(gvkey)28216	-0.0018971351	0.0007444483	-2.55	0.01083
factor(gvkey)28303	-0.0049828172	0.0006144079	-8.11	0.00000000000000056
factor(gvkey)28629	0.0015723077	0.0006704745	2.35	0.01904
factor(gvkey)28924	-0.0069571908	0.0005753993	-12.09	< 0.0000000000000002
factor(gvkey)29011	-0.0020427621	0.0008236239	-2.48	0.01314
factor(gvkey)29241	0.0003723210	0.0006256761	0.60	0.55181
factor(gvkey)29612	-0.0067890231	0.0007135287	-9.51	< 0.0000000000000002
factor(gvkey)29710	-0.0057506878	0.0004911316	-11.71	< 0.0000000000000002
factor(gvkey)29751	-0.0020505173	0.0006902907	-2.97	0.00298
factor(gvkey)30138	-0.0053130883	0.0004796209	-11.08	< 0.0000000000000002
factor(gvkey)61494	0.0000151713	0.0003733015	0.04	0.96758
factor(gvkey)65772	-0.0031761754	0.0005792639	-5.48	0.00000004263580368
factor(gvkey)116504	-0.0047808323	0.0005981624	-7.99	0.00000000000000144
factor(gvkey)118502	-0.0035219974	0.0005965896	-5.90	0.00000000365405053
factor(gvkey)121718	0.0009542738	0.0005365059	1.78	0.07532
factor(gvkey)122841	0.0007643660	0.0006335097	1.21	0.22763
factor(gvkey)126554	-0.0041430565	0.0006428371	-6.44	0.00000000012007218
factor(gvkey)136725	0.0040679500	0.0010087540	4.03	0.00005549388418461
factor(gvkey)137131	-0.0040622160	0.0005909613	-6.87	0.00000000000655658
factor(gvkey)141384	-0.0035410978	0.0007792907	-4.54	0.00000557365929477
factor(gvkey)145049	-0.0013745979	0.0004600861	-2.99	0.00282
factor(gvkey)148470	-0.0003361111	0.0003777395	-0.89	0.37359
factor(gvkey)150139	-0.0058920461	0.0006420892	-9.18	< 0.0000000000000002
factor(gvkey)156617	-0.0053372872	0.0007355503	-7.26	0.00000000000042275
factor(gvkey)160211	-0.0004896264	0.0019241575	-0.25	0.79914
factor(gvkey)164416	-0.0046142997	0.0005846483	-7.89	0.00000000000000322
factor(gvkey)164664	-0.0002743970	0.0005338425	-0.51	0.60726
factor(gvkey)165746	-0.0055506550	0.0005253835	-10.56	< 0.0000000000000002
factor(gvkey)170841	-0.0050329659	0.0005095553	-9.88	< 0.0000000000000002
factor(gvkey)174317	-0.0050662063	0.0005974773	-8.48	< 0.0000000000000002
factor(gvkey)177376	-0.0060990320	0.0007980697	-7.64	0.00000000000002298
factor(gvkey)178493	-0.0055939979	0.0006442683	-8.68	< 0.0000000000000002
factor(gvkey)179666	0.0010692689	0.0007737167	1.38	0.16700
factor(gvkey)180402	-0.0050451336	0.0005334268	-9.46	< 0.0000000000000002
factor(gvkey)180646	NA	NA	NA	NA
factor(gvkey)184500	0.0008147922	0.0011999368	0.68	0.49713

factor(sic)1531	NA	NA	NA	NA
factor(sic)1623	NA	NA	NA	NA
factor(sic)2080	NA	NA	NA	NA
factor(sic)2300	NA	NA	NA	NA
factor(sic)2320	NA	NA	NA	NA
factor(sic)2510	NA	NA	NA	NA
factor(sic)2621	NA	NA	NA	NA
factor(sic)2810	NA	NA	NA	NA
factor(sic)2820	NA	NA	NA	NA
factor(sic)2836	NA	NA	NA	NA
factor(sic)2844	NA	NA	NA	NA
factor(sic)2851	NA	NA	NA	NA
factor(sic)2911	NA	NA	NA	NA
factor(sic)3490	NA	NA	NA	NA
factor(sic)3561	NA	NA	NA	NA
factor(sic)3576	NA	NA	NA	NA
factor(sic)3674	NA	NA	NA	NA
factor(sic)3711	NA	NA	NA	NA
factor(sic)3724	NA	NA	NA	NA
factor(sic)3812	NA	NA	NA	NA
factor(sic)3823	NA	NA	NA	NA
factor(sic)3826	NA	NA	NA	NA
factor(sic)3843	NA	NA	NA	NA
factor(sic)3845	NA	NA	NA	NA
factor(sic)3990	NA	NA	NA	NA
factor(sic)4213	NA	NA	NA	NA
factor(sic)4833	NA	NA	NA	NA
factor(sic)4841	NA	NA	NA	NA
factor(sic)4911	NA	NA	NA	NA
factor(sic)4931	NA	NA	NA	NA
factor(sic)5047	NA	NA	NA	NA
factor(sic)5065	NA	NA	NA	NA
factor(sic)5140	NA	NA	NA	NA
factor(sic)5150	NA	NA	NA	NA
factor(sic)5211	NA	NA	NA	NA
factor(sic)5661	NA	NA	NA	NA
factor(sic)5734	NA	NA	NA	NA
factor(sic)5812	NA	NA	NA	NA
factor(sic)5912	NA	NA	NA	NA
factor(sic)5940	NA	NA	NA	NA
factor(sic)5944	NA	NA	NA	NA
factor(sic)5961	NA	NA	NA	NA
factor(sic)6141	NA	NA	NA	NA
factor(sic)6200	NA	NA	NA	NA
factor(sic)6211	NA	NA	NA	NA
factor(sic)6282	NA	NA	NA	NA
factor(sic)6311	NA	NA	NA	NA
factor(sic)6324	NA	NA	NA	NA
factor(sic)6331	NA	NA	NA	NA
factor(sic)6798	NA	NA	NA	NA
factor(sic)7310	NA	NA	NA	NA
factor(sic)7323	NA	NA	NA	NA
factor(sic)7370	NA	NA	NA	NA
factor(sic)7372	NA	NA	NA	NA

factor(sic)7389	NA	NA	NA	NA
factor(sic)7900	NA	NA	NA	NA
factor(sic)7990	NA	NA	NA	NA
factor(sic)8700	NA	NA	NA	NA
factor(sic)8721	NA	NA	NA	NA
factor(sic)8731	NA	NA	NA	NA
factor(sic)8742	NA	NA	NA	NA

(Intercept)	***
bm	***
pe_exi	
ps	.
pcf	
dpr	
npm	
opmbd	**
opmad	
gpm	.
roa	
roe	
roce	
debt_at	***
de_ratio	
intcov	
cash_ratio	
quick_ratio	
curr_ratio	
inv_turn	.
at_turn	
debt_assets	***
ptb	
rect_turn	
sale_nwc	
divyield	
p_daily_change	
volatile	*
pop_ind	***
is_mon	*
is_tue	.
is_wed	
is_thu	
is_fri	
is_sat	
log(atq)	
factor(gvkey)1447	***
factor(gvkey)1487	***
factor(gvkey)1659	***
factor(gvkey)1878	
factor(gvkey)1919	***
factor(gvkey)1920	
factor(gvkey)3170	.
factor(gvkey)3226	***
factor(gvkey)3336	*
factor(gvkey)3708	***

```
factor(gvkey)3814   ***
factor(gvkey)4060
factor(gvkey)4201
factor(gvkey)4321   ***
factor(gvkey)4839   ***
factor(gvkey)4988   ***
factor(gvkey)5492   ***
factor(gvkey)5742   ***
factor(gvkey)6435   ***
factor(gvkey)6788   ***
factor(gvkey)6829   ***
factor(gvkey)7163   ***
factor(gvkey)8113   ***
factor(gvkey)8358   ***
factor(gvkey)8402   ***
factor(gvkey)8463   ***
factor(gvkey)8479   ***
factor(gvkey)8549   ***
factor(gvkey)9599   ***
factor(gvkey)9667   ***
factor(gvkey)10247  ***
factor(gvkey)10983  ***
factor(gvkey)11115  .
factor(gvkey)11264  ***
factor(gvkey)11304  ***
factor(gvkey)11584  ***
factor(gvkey)11669  ***
factor(gvkey)11770  ***
factor(gvkey)12441  ***
factor(gvkey)13092
factor(gvkey)14412  ***
factor(gvkey)17035  ***
factor(gvkey)23225  ***
factor(gvkey)23252  ***
factor(gvkey)24468  *
factor(gvkey)24800  .
factor(gvkey)25124  *
factor(gvkey)25340  ***
factor(gvkey)26011  ***
factor(gvkey)27794  ***
factor(gvkey)27914  ***
factor(gvkey)28034  ***
factor(gvkey)28216  *
factor(gvkey)28303  ***
factor(gvkey)28629  *
factor(gvkey)28924  ***
factor(gvkey)29011  *
factor(gvkey)29241
factor(gvkey)29612  ***
factor(gvkey)29710  ***
factor(gvkey)29751  **
factor(gvkey)30138  ***
factor(gvkey)61494
factor(gvkey)65772  ***
```

```
factor(gvkey)116504 ***
factor(gvkey)118502 ***
factor(gvkey)121718 .
factor(gvkey)122841
factor(gvkey)126554 ***
factor(gvkey)136725 ***
factor(gvkey)137131 ***
factor(gvkey)141384 ***
factor(gvkey)145049 **
factor(gvkey)148470
factor(gvkey)150139 ***
factor(gvkey)156617 ***
factor(gvkey)160211
factor(gvkey)164416 ***
factor(gvkey)164664
factor(gvkey)165746 ***
factor(gvkey)170841 ***
factor(gvkey)174317 ***
factor(gvkey)177376 ***
factor(gvkey)178493 ***
factor(gvkey)179666
factor(gvkey)180402 ***
factor(gvkey)180646
factor(gvkey)184500
factor(sic)1531
factor(sic)1623
factor(sic)2080
factor(sic)2300
factor(sic)2320
factor(sic)2510
factor(sic)2621
factor(sic)2810
factor(sic)2820
factor(sic)2836
factor(sic)2844
factor(sic)2851
factor(sic)2911
factor(sic)3490
factor(sic)3561
factor(sic)3576
factor(sic)3674
factor(sic)3711
factor(sic)3724
factor(sic)3812
factor(sic)3823
factor(sic)3826
factor(sic)3843
factor(sic)3845
factor(sic)3990
factor(sic)4213
factor(sic)4833
factor(sic)4841
factor(sic)4911
factor(sic)4931
```

```
factor(sic)5047
factor(sic)5065
factor(sic)5140
factor(sic)5150
factor(sic)5211
factor(sic)5661
factor(sic)5734
factor(sic)5812
factor(sic)5912
factor(sic)5940
factor(sic)5944
factor(sic)5961
factor(sic)6141
factor(sic)6200
factor(sic)6211
factor(sic)6282
factor(sic)6311
factor(sic)6324
factor(sic)6331
factor(sic)6798
factor(sic)7310
factor(sic)7323
factor(sic)7370
factor(sic)7372
factor(sic)7389
factor(sic)7900
factor(sic)7990
factor(sic)8700
factor(sic)8721
factor(sic)8731
factor(sic)8742
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00152 on 11971 degrees of freedom
Multiple R-squared:  0.174, Adjusted R-squared:  0.166
F-statistic: 20.7 on 122 and 11971 DF,  p-value: <0.0000000000000002
```

5.2 Financial (remove correlated variables)

We also constructed six more models with highly correlated variables removed based on the multicollinearity test conducted in section 4.5 of EDA. Specifically, we remove roce, quick_ratio, opmbd and opmad from our model before running the regression again.

```
# remove some high correlated variables
# remove roce,quick_ratio,opmbd,opmad

Mod7_formula<- as.formula("growth_rate ~ bm + pe_exi + ps + pcf + dpr + npm + gpm + roa + roe

Mod8_formula<- as.formula("growth_rate ~ bm_lag + pe_exi_lag + ps_lag + pcf_lag + dpr_lag +
npm_lag + gpm_lag + roa_lag +roe_lag + debt_at_lag + de_ratio_lag + intcov_lag + cash_ratio_la
```

```r
# # run regression Model7 : stock price and financial ratios (remove correlated variables)
Model7 <- lm(Mod7_formula,data=Train)

# run regression Model8 : stock price (lag) and financial ratios (lag, remove correlated varia
Model8 <-lm(Mod8_formula,data=Train)

# run regression Model9 : Model7 (remove correlated variables) with industry FE
Mod9_formula <- update(Mod7_formula, . ~ . + factor(sic))
Model9 <- lm(Mod9_formula,data=Train)

# run regression Model10 : Model8 (lag, remove correlated variables) with industry FE
Mod10_formula <- update(Mod8_formula, . ~ . + factor(sic))
Model10 <-lm(Mod10_formula,data=Train)

# run regression Model11 : Model7 (remove correlated variables) with industry FE and firm FE
Mod11_formula <- update(Mod7_formula, . ~ . + factor(gvkey) + factor(sic))
Model11 <-lm(Mod11_formula,data=Train)

# run regression Model12 : Model8 (lag, remove correlated variables) with industry FE and firm
Mod12_formula <- update(Mod8_formula, . ~ . + factor(gvkey) + factor(sic))
Model12 <-lm(Mod12_formula,data=Train)
```

5.3 In Sample Evaluation

The growth rates are predicted using twelve different models (Model1 through Model12) and stored in separate variables (`Train$Pre_rate_1` through `Train$Pre_rate_6`) within the `Train` dataset. These predicted growth rates are then used to estimate the followers for each model, considering the lagged followers and the predicted growth rates. The root mean square error (RMSE) function (`rmse()`) calculates the accuracy of each model by comparing the actual followers with the predicted followers, offering an in-sample accuracy assessment. Finally, the RMSE values for each model are computed.

```r
# predict followers growth rate on Train
Train$Pre_rate_1 <- predict(Model1, Train)
Train$Pre_rate_2 <- predict(Model2, Train)
Train$Pre_rate_3 <- predict(Model3, Train)
Train$Pre_rate_4 <- predict(Model4, Train)
Train$Pre_rate_5 <- predict(Model5, Train)
```

```
Warning in predict.lm(Model5, Train): prediction from rank-deficient fit;
attr(*, "non-estim") has doubtful cases
```

```r
Train$Pre_rate_6 <- predict(Model6, Train)
Train$Pre_rate_7 <- predict(Model7, Train)
Train$Pre_rate_8 <- predict(Model8, Train)
Train$Pre_rate_9 <- predict(Model9, Train)
Train$Pre_rate_10 <- predict(Model10, Train)
Train$Pre_rate_11 <- predict(Model11, Train)
Train$Pre_rate_12 <- predict(Model12, Train)
```

```
Warning in predict.lm(Model12, Train): prediction from rank-deficient fit;
attr(*, "non-estim") has doubtful cases
```

```r
# compute Train predicted followers based on predicted growth rate
Train$Followers_M1 <- (1+Train$Pre_rate_1)*Train$followers_lag
Train$Followers_M2 <- (1+Train$Pre_rate_2)*Train$followers_lag
Train$Followers_M3 <- (1+Train$Pre_rate_3)*Train$followers_lag
Train$Followers_M4 <- (1+Train$Pre_rate_4)*Train$followers_lag
Train$Followers_M5 <- (1+Train$Pre_rate_5)*Train$followers_lag
Train$Followers_M6 <- (1+Train$Pre_rate_6)*Train$followers_lag
Train$Followers_M7 <- (1+Train$Pre_rate_7)*Train$followers_lag
Train$Followers_M8 <- (1+Train$Pre_rate_8)*Train$followers_lag
Train$Followers_M9 <- (1+Train$Pre_rate_9)*Train$followers_lag
Train$Followers_M10 <- (1+Train$Pre_rate_10)*Train$followers_lag
Train$Followers_M11 <- (1+Train$Pre_rate_11)*Train$followers_lag
Train$Followers_M12 <- (1+Train$Pre_rate_12)*Train$followers_lag

# compute In Sample Accuracy
rmse <- function(v1, v2) {
    sqrt(mean((v1 - v2)^2, na.rm = T))
  }


RMSE <- c(rmse(Train$followers,Train$Followers_M1),
          rmse(Train$followers,Train$Followers_M2),
          rmse(Train$followers,Train$Followers_M3),
          rmse(Train$followers,Train$Followers_M4),
          rmse(Train$followers,Train$Followers_M5),
          rmse(Train$followers,Train$Followers_M6),
          rmse(Train$followers,Train$Followers_M7),
          rmse(Train$followers,Train$Followers_M8),
          rmse(Train$followers,Train$Followers_M9),
          rmse(Train$followers,Train$Followers_M10),
          rmse(Train$followers,Train$Followers_M11),
          rmse(Train$followers,Train$Followers_M12))

# retrieve RMSE for all 12 financial models
names(RMSE) <- c("Model1", "Model2", "Model3", "Model4","Model5","Model6",
                "Model7", "Model8", "Model9", "Model10","Model11","Model12")
RMSE
```

```
 Model1  Model2  Model3  Model4  Model5  Model6  Model7  Model8  Model9 Model10
  934.6   936.8   893.3   892.2   903.7   895.7   938.1   941.6   897.0   898.5
Model11 Model12
  897.7   904.4
```

5.4 Prediction of Test followers

We retrieve the number of followers for each company (`gvkey`) from the train data, sorts the data in descending order by `gvkey` and date. Subesquently, we used the slice (1) function to obtain the last day's followers of each company in the train dataset.

The last day's followers is then joined with the test data using a full join on `gvkey` and `date`. This ensures that each company's last day followers from the Train data are paired with corresponding rows in the test data.

Using the regression Model, we then perform out of sample prediction for growth rates on the test data. With a while loop, the code fills the missing values of the `followers` column in the test data by iteratively calculating the followers using the daily predicted growth rate and the lagged followers value. This continues until no missing values are left.

```
#obtain the last day no.of followers from training data and apply the growth rate for Test dat
last_day_followers <- train %>%
  group_by(gvkey) %>%
  arrange(gvkey, desc(date)) %>%
  dplyr::slice(1) %>%
  ungroup()

# remove the 6 outlier companies
last_day_followers <- last_day_followers %>% filter(!gvkey %in% c("10005", "12233", "21238", "

# create a function to fill in test data with the predicting result
predict_test <- function(last_day_followers, Test_data, Model_no){

  Test_temp <- full_join(last_day_followers, Test_data, by = join_by(gvkey, date)) %>%
    arrange(gvkey, date) %>%
    select(-sic.x) %>%
    rename(`sic` = sic.y) %>%
    select(-ID.x) %>%
    rename(`ID` = ID.y)

  Test_temp <- Test_temp %>% mutate(Pre_rate_1 = predict(Model_no, Test_temp))

  while (any(is.na(Test_temp[["followers"]]))) {
    Test_temp <- Test_temp %>%
      mutate(followers= ifelse(is.na(followers), lag(followers) * (1+Pre_rate_1), followers))
  }

  return (Test_temp)

}


Test1 <- predict_test(last_day_followers, Test, Model1)
Test2 <- predict_test(last_day_followers, Test, Model2)
Test3 <- predict_test(last_day_followers, Test, Model3)
Test4 <- predict_test(last_day_followers, Test, Model4)
Test5 <- predict_test(last_day_followers, Test, Model5)
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `Pre_rate_1 = predict(Model_no, Test_temp)`.
Caused by warning in `predict.lm()`:
! prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
Test6 <- predict_test(last_day_followers, Test, Model6)
Test7 <- predict_test(last_day_followers, Test, Model7)
Test8 <- predict_test(last_day_followers, Test, Model8)
Test9 <- predict_test(last_day_followers, Test, Model9)
Test10 <- predict_test(last_day_followers, Test, Model10)
Test11 <- predict_test(last_day_followers, Test, Model11)
Test12 <- predict_test(last_day_followers, Test, Model12)
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `Pre_rate_1 = predict(Model_no, Test_temp)`.
Caused by warning in `predict.lm()`:
! prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

In this section, we fill in the number of followers for six special companies not included in our initial model. We adopt two methods to achieve this. Initially, we merge the testing data (Test) with financial variables (combined_vars) based on the company identifier (gvkey) and date. We then extract the last day's number of followers for these six companies from the training data (Train), ensuring inclusion of only the required companies.

Subsequently, we fill in missing values in the followers' column with the latest available data from earlier dates for each of the six companies. This is accomplished using the fill() function, which propagates non-missing values forward to ensure completeness in the dataset. Finally, the resulting dataset (Test_6) is organized by gvkey, date, and other relevant identifiers, providing a complete set of observations for the specified companies.

```
# fill in those six companies with last day in June
# should have 313 obs
Test_6<- left_join(test,combined_vars,by=c("gvkey","date"))
Test_6 <- Test_6 %>% filter(gvkey %in% c("10005", "12233", "21238", "25405", "147661", "178548

last_day_followers_6 <- train %>%
  group_by(gvkey) %>%
  arrange(gvkey, desc(date)) %>%
  dplyr::slice(1) %>%
  ungroup()

last_day_followers_6 <- last_day_followers_6 %>% filter(gvkey %in% c("10005", "12233", "21238"
Test_6 <- full_join(last_day_followers_6, Test_6)
```

```
Joining with `by = join_by(gvkey, date, ID)`
```

```
Test_6 <- Test_6 %>% arrange(gvkey, date)

# fill in the number of followers with the latest available in early days
Test_6 <- Test_6 %>%
  arrange(gvkey, date) %>%
  group_by(gvkey) %>%
  fill(followers,.direction = 'down') %>%
  ungroup()
Test_6 <- Test_6 %>% select(gvkey, date, followers, ID)
```

We fill in missing data for six specific companies based on calculated growth rates. Initially, the testing data (`Test_6`) is merged with financial variables (`combined_vars`) using the company identifier (`gvkey`) and the date. Then, a date sequence from 1 July 2017 to 30 September 2017 is generated to ensure comprehensive coverage.

Next, we cross join between the unique `gvkey` values and the date sequence to obtain a full combination of `gvkey` and date. The date format is converted into integers for merging purposes.

The last day's followers for the six companies are extracted from the training data and joined with `Test_6`. Following this, the growth rates from the training data for the period 31 March to 30 June are combined.

Subsequently, missing values in the followers' column are filled with the latest available data from both earlier and later dates for each company, ensuring completeness in the dataset.

However, this fill in method doesn't give us improved results compared with the first fill in method, so we don't use this approach and just put it here for showing different approaches we have tried.

```
#-------------------start of second fill in method----------------------------------#
# fill in those six companies with calculated based on growth rate
#Test_6<- left_join(test,combined_vars,by=c("gvkey","date"))
#Test_6 <- Test_6 %>% filter(gvkey %in% c("10005", "12233", "21238", "25405", "147661", "17854

# create date sequence
#date_df <- data.frame(date = seq(from = ymd("2017-07-01"), to = ymd("2017-09-30"), by = "1 da

# crossjoin the date_df and unique_gvkey to get the full combination of gvkey and date
#unique_gvkey <- as.data.frame(unique(Test_6$gvkey))
#full_gvkey_date <- cross_join(unique_gvkey, date_df)
#colnames(full_gvkey_date) <- c("gvkey", "date")

## Convert datadate into integer for merging
#full_gvkey_date$date<- as.Date(full_gvkey_date$date)
#full_gvkey_date$date<- as.integer(format(full_gvkey_date$date, "%Y%m%d"))

# merge Test_6 to full_gvkey_date
#Test_6 <- merge(x = full_gvkey_date, y = Test_6, by = c("gvkey", "date"), all = TRUE)

#last_day_followers_6 <- train %>%
#  group_by(gvkey) %>%
#  arrange(gvkey, desc(date)) %>%
#  dplyr::slice(1) %>%
#  ungroup()
#last_day_followers_6 <- last_day_followers_6 %>% filter(gvkey %in% c("10005", "12233", "21238
#Test_6 <- full_join(last_day_followers_6, Test_6)
#Test_6 <- Test_6 %>% arrange(gvkey, date)


# combine with growth rate in train data from 03/31 to 06/30
#Train_6 <- left_join(combined_vars_train,train,by=c("gvkey","date"))
#Train_6 <- select(Train_6, -sic.x)
```

```
#Train_6 <- Train_6 %>% rename(`sic` = sic.y)
#summary(Train_6)

# remove 6 companies that don't have account
#Train_6 <- Train_6 %>% filter(gvkey %in% c("10005", "12233", "21238", "25405", "147661", "178
#summary(Train_6) # no NA for vars need

# fill in the number of followers with the latest available in early days
#Train_6 <- Train_6 %>%
#  arrange(gvkey, date) %>%
#  group_by(gvkey) %>%
#  fill(followers,.direction = 'down') %>%
#  ungroup()

#Train_6 <- Train_6 %>%
#  arrange(gvkey, date) %>%
#  group_by(gvkey) %>%
#  fill(sic,.direction = 'down') %>%
#  ungroup()

# fill in the number of followers with the latest available in later days
#Train_6 <- Train_6 %>%
#  arrange(gvkey, date) %>%
#  group_by(gvkey) %>%
#  fill(followers,.direction = 'up') %>%
#  ungroup()

#Train_6 <- Train_6 %>%
#  arrange(gvkey, date) %>%
#  group_by(gvkey) %>%
#  fill(sic,.direction = 'up') %>%
#  ungroup()

#summary(Train_6) # no NA for followers and sic
```

We calculate growth rates for each company in the dataset `Train_6` by comparing current followers to the lagged followers. After computing the growth rates, the summary reveals 6 NA values, indicating missing or insufficient data for certain companies. The subsequent instruction aims to ensure that both `Train_6` and the testing dataset `Test` contain the expected number of observations (465) and are organized by `gvkey` and date, ensuring consistency in analysis and modeling efforts.

```
#Train_6<- Train_6 %>%
#  group_by(gvkey) %>%
#  mutate(followers_lag=lag(followers,1))%>%
#  mutate(growth_rate = (followers / lag(followers)-1)) %>%
#  ungroup()

## 6 NA Value for growth rate
# make sure Train_6 and Test both have 465 obs and arrange by gvkey and date
#summary(Train_6)
```

In this process, we streamline the datasets Train_6 and Test_6 by removing extraneous columns and filtering out specific gvkey records. Both datasets are sorted by gvkey and date to maintain consistency. Subsequently, Train_6 columns are merged with Test_6 to synchronize the data. Any missing values in the followers column of Test_6 are filled iteratively using a growth rate formula. Finally, we retain only the essential columns (gvkey, date, followers, ID) in Test_6 and ensure that the date is after 30 June 2017, and that ID is not missing. This method ensures Test_6 contains the required 313 observations and is ready for further analysis.

```
#Train_6 <- Train_6 %>% select(date, gvkey, growth_rate)
#Train_6 <- Train_6 %>% filter(date > 20170329)
#Train_6 <- Train_6 %>% filter(!gvkey == 21238)
#Train_6 <- Train_6 %>% arrange(gvkey, date)
#Test_6 <- Test_6 %>% select(gvkey, date, followers, ID)
#Test_6 <- Test_6 %>% filter(!gvkey == 21238)
#Test_6 <- Test_6 %>% arrange(gvkey, date)

# bind columns of Train_6 and Test_6
#Test_6 <- cbind(Test_6, Train_6)
#while (any(is.na(Test_6$followers))) {
#  Test_6$followers = ifelse(is.na(Test_6$followers),
#                            lag(Test_6$followers) * (1+Test_6$growth_rate),
#                            Test_6$followers)
#}

# bind rows with Test data and then innerjoin with submit sample
# make sure the first four columns are gvkey, date, followers ID
# Test_6 should have 313 obs
#Test_6 <- Test_6[,1:4]
#Test_6 <- Test_6 %>% filter(date > 20170630 & !is.na(ID))
#--------------------end of second fill in method---------------------------------------#
```

We merge the six specific companies with the rest of the dataset. The merged data is filtered to include only records from 1 July 2017 onwards. The follower counts are rounded to ensure integer values. From the resulting dataset, only the columns for ID and followers are selected, representing the necessary data for submission. This subset of data is saved as a CSV file named "submission_M1_fin.csv", excluding row numbers.

```
pacman::p_load(tidyverse)
# merge the six companies with other companies
Test_submission1 <- Test1 %>% select(gvkey, date, followers, ID)
Test_submission1 <- rbind(Test_submission1, Test_6)
Test_submission1 <- Test_submission1 %>% filter(date>=20170701)
Test_submission1$followers <- round(Test_submission1$followers)
Followers_submisson_M1 <- Test_submission1 %>% select(ID, followers)

Test_submission2 <- Test2 %>% select(gvkey, date, followers, ID)
Test_submission2 <- rbind(Test_submission2, Test_6)
Test_submission2 <- Test_submission2 %>% filter(date>=20170701)
Test_submission2$followers <- round(Test_submission2$followers)
Followers_submisson_M2 <- Test_submission2 %>% select(ID, followers)

Test_submission3 <- Test3 %>% select(gvkey, date, followers, ID)
```

```r
Test_submission3 <- rbind(Test_submission3, Test_6)
Test_submission3 <- Test_submission3 %>% filter(date>=20170701)
Test_submission3$followers <- round(Test_submission3$followers)
Followers_submisson_M3 <- Test_submission3 %>% select(ID, followers)

Test_submission4 <- Test4 %>% select(gvkey, date, followers, ID)
Test_submission4 <- rbind(Test_submission4, Test_6)
Test_submission4 <- Test_submission4 %>% filter(date>=20170701)
Test_submission4$followers <- round(Test_submission4$followers)
Followers_submisson_M4 <- Test_submission4 %>% select(ID, followers)

Test_submission5 <- Test5 %>% select(gvkey, date, followers, ID)
Test_submission5 <- rbind(Test_submission5, Test_6)
Test_submission5 <- Test_submission5 %>% filter(date>=20170701)
Test_submission5$followers <- round(Test_submission5$followers)
Followers_submisson_M5 <- Test_submission5 %>% select(ID, followers)

Test_submission6 <- Test6 %>% select(gvkey, date, followers, ID)
Test_submission6 <- rbind(Test_submission6, Test_6)
Test_submission6 <- Test_submission6 %>% filter(date>=20170701)
Test_submission6$followers <- round(Test_submission6$followers)
Followers_submisson_M6 <- Test_submission6 %>% select(ID, followers)

Test_submission7 <- Test7 %>% select(gvkey, date, followers, ID)
Test_submission7 <- rbind(Test_submission7, Test_6)
Test_submission7 <- Test_submission7 %>% filter(date>=20170701)
Test_submission7$followers <- round(Test_submission7$followers)
Followers_submisson_M7 <- Test_submission7 %>% select(ID, followers)

Test_submission8 <- Test8 %>% select(gvkey, date, followers, ID)
Test_submission8 <- rbind(Test_submission8, Test_6)
Test_submission8 <- Test_submission8 %>% filter(date>=20170701)
Test_submission8$followers <- round(Test_submission8$followers)
Followers_submisson_M8 <- Test_submission8 %>% select(ID, followers)

Test_submission9 <- Test9 %>% select(gvkey, date, followers, ID)
Test_submission9 <- rbind(Test_submission9, Test_6)
Test_submission9 <- Test_submission9 %>% filter(date>=20170701)
Test_submission9$followers <- round(Test_submission9$followers)
Followers_submisson_M9 <- Test_submission9 %>% select(ID, followers)

Test_submission10 <- Test10 %>% select(gvkey, date, followers, ID)
Test_submission10 <- rbind(Test_submission10, Test_6)
Test_submission10 <- Test_submission10 %>% filter(date>=20170701)
Test_submission10$followers <- round(Test_submission10$followers)
Followers_submisson_M10 <- Test_submission10 %>% select(ID, followers)

Test_submission11 <- Test11 %>% select(gvkey, date, followers, ID)
Test_submission11 <- rbind(Test_submission11, Test_6)
Test_submission11 <- Test_submission11 %>% filter(date>=20170701)
Test_submission11$followers <- round(Test_submission11$followers)
Followers_submisson_M11 <- Test_submission11 %>% select(ID, followers)

Test_submission12 <- Test12 %>% select(gvkey, date, followers, ID)
```

```r
Test_submission12 <- rbind(Test_submission12, Test_6)
Test_submission12 <- Test_submission12 %>% filter(date>=20170701)
Test_submission12$followers <- round(Test_submission12$followers)
Followers_submisson_M12 <- Test_submission12 %>% select(ID, followers)

# save the sample of submission
write.csv(Followers_submisson_M1, "submission_M1.csv", row.names = FALSE)
write.csv(Followers_submisson_M2, "submission_M2.csv", row.names = FALSE)
write.csv(Followers_submisson_M3, "submission_M3.csv", row.names = FALSE)
write.csv(Followers_submisson_M4, "submission_M4.csv", row.names = FALSE)
write.csv(Followers_submisson_M5, "submission_M5.csv", row.names = FALSE)
write.csv(Followers_submisson_M6, "submission_M6.csv", row.names = FALSE)
write.csv(Followers_submisson_M7, "submission_M7.csv", row.names = FALSE)
write.csv(Followers_submisson_M8, "submission_M8.csv", row.names = FALSE)
write.csv(Followers_submisson_M9, "submission_M9.csv", row.names = FALSE)
write.csv(Followers_submisson_M10, "submission_M10.csv", row.names = FALSE)
write.csv(Followers_submisson_M11, "submission_M11.csv", row.names = FALSE)
write.csv(Followers_submisson_M12, "submission_M12.csv", row.names = FALSE)
```

5.2 Out of Sample Evaluation

The submission CSV files for the 12 models have been exported, and the analysis indicates that Model 1 and Model 3_1 have the most favorable outcome among them.

submission_M6.csv Complete · now	**5130.81508**
submission_M5.csv Complete · 13s ago	**5222.18741**
submission_M4.csv Complete · 26s ago	**4828.89567**
submission_M3.csv Complete · 37s ago	**4717.58394**
submission_M2.csv Complete · 1m ago	**3522.23707**
submission_M1.csv Complete · 1m ago	**3381.59815**

submission_M12.csv Complete · now	5489.16648	☐
submission_M11.csv Complete · 13s ago	5103.70607	☐
submission_M10.csv Complete · 23s ago	3563.6714	☐
submission_M9.csv Complete · 34s ago	3362.3681	☐
submission_M8.csv Complete · 44s ago	3392.02717	☐
submission_M7.csv Complete · 1m ago	3436.44338	☐

6. Non-financial Model

6.1 Non-financial Model Construction

In this section, we build a model using non-financial data exclusively. Based on the multicollinearity test conducted in section 4.5 of EDA, days_joined, months_joined and years_joined are analysed as highly correlated.Thus, we only retain days_joined variable in the construction of our non financial variables model as the days_joined variable has greater variability in terms of data compared to the other two.

Hence, the 3 non-financial models constructed are as follows:

Model13: Growth_rate = Profile vars + announcements vars + popular ind + week dummy + ln(asset) Model14: Growth_rate = Profile vars + announcements vars + popular ind + week dummy + ln(asset) + industry FE Model15: Growth_rate = Profile vars + announcements vars + popular ind + week dummy + ln(asset) + industry FE + firm FE

The above mentioned 3 non financial OLS regression models uses the same set of dependent and independent variables. The difference between the models are on the fixed effect levels applied. We added industry level fixed effect in model14 and we added industry and firm level fixed effect in model15. The fixed effects allows for control on the industry and firm level varying features. The formula for the model (`Mod13_formula`) is specified to capture the relationship between the growth rate and the non financial variables. Specifically, the formula encompasses profile_info, announcements related data, alongside dummies for each day of the week, and log "atq" as a measure of company size.

```
# create formula
Mod13_formula <- as.formula("growth_rate ~ is_annouce + red_annouce + no_annoucement + TFF + c
pop_ind + is_mon + is_tue + is_wed + is_thu + is_fri + is_sat + log(atq)")

# run regression Model13 : non financial variables
Model13 <- lm(Mod13_formula,data=Train)
summary(Model13)
```

```
Call:
lm(formula = Mod13_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03480 -0.00049 -0.00023  0.00014  0.04849 

Coefficients: (1 not defined because of singularities)
                            Estimate     Std. Error t value
(Intercept)           -0.00021920305  0.00017936259   -1.22
is_annouce            -0.00019560134  0.00008530193   -2.29
red_annouce            0.00112050881  0.00020386272    5.50
no_annoucement         0.00008874467  0.00003196948    2.78
TFF                   -0.00000000413  0.00000000166   -2.49
coleman_liau           0.00000458147  0.00000118653    3.86
flesch                -0.00000338959  0.00000172042   -1.97
fog_info               0.00000403556  0.00000604250    0.67
num_words              0.00002084353  0.00000235400    8.85
bio_complete           0.00039435380  0.00004529792    8.71
has_url               -0.00010274438  0.00006284697   -1.63
hashtags_count        -0.00000867484  0.00001659062   -0.52
has_hashtags          -0.00007636390  0.00005470784   -1.40
contains_company_name  0.00017272016  0.00004380049    3.94
contains_words        -0.00007420327  0.00005501823   -1.35
custom_content                    NA             NA      NA
days_joined            0.00000000990  0.00000001492    0.66
pop_ind                0.00038361237  0.00003739998   10.26
is_mon                 0.00018728825  0.00006507043    2.88
is_tue                 0.00019129709  0.00006457106    2.96
is_wed                 0.00018295498  0.00006452894    2.84
is_thu                 0.00014317596  0.00006427314    2.23
is_fri                 0.00016852505  0.00006438387    2.62
is_sat                -0.00000809517  0.00007218723   -0.11
log(atq)               0.00001023262  0.00000951659    1.08
                                  Pr(>|t|)    
(Intercept)                        0.22169    
is_annouce                         0.02186 *  
red_annouce                     0.00000004 ***
no_annoucement                     0.00551 ** 
TFF                                0.01289 *  
coleman_liau                       0.00011 ***
flesch                             0.04884 *  
fog_info                           0.50423    
num_words             < 0.0000000000000002 ***
bio_complete          < 0.0000000000000002 ***
has_url                            0.10211    
hashtags_count                     0.60107    
has_hashtags                       0.16279    
contains_company_name           0.00008081 ***
contains_words                     0.17746    
custom_content                          NA    
days_joined                        0.50709    
pop_ind               < 0.0000000000000002 ***
```

```
is_mon                            0.00401 **
is_tue                            0.00306 **
is_wed                            0.00459 **
is_thu                            0.02592 *
is_fri                            0.00887 **
is_sat                            0.91071
log(atq)                          0.28229
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00164 on 12070 degrees of freedom
Multiple R-squared:  0.0305,    Adjusted R-squared:  0.0287
F-statistic: 16.5 on 23 and 12070 DF,  p-value: <0.0000000000000002
```

```r
# run regression Model14 : non financial variables with industry FE
Mod14_formula <- update(Mod13_formula, . ~ . + factor(sic))
Model14 <- lm(Mod14_formula,data=Train)
summary(Model14)
```

```
Call:
lm(formula = Mod14_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.03636 -0.00036 -0.00007  0.00017  0.04696

Coefficients: (2 not defined because of singularities)
                           Estimate    Std. Error t value             Pr(>|t|)
(Intercept)            0.0014155328  0.0005094192    2.78              0.00547
is_annouce            -0.0002357122  0.0000806056   -2.92              0.00346
red_annouce            0.0012326152  0.0001922320    6.41   0.0000000001488863
no_annoucement         0.0001010130  0.0000302048    3.34              0.00083
TFF                    0.0000003346  0.0000000190   17.61 < 0.0000000000000002
coleman_liau          -0.0000189883  0.0000039731   -4.78   0.0000017810667136
flesch                 0.0000235789  0.0000042154    5.59   0.0000000227392499
fog_info               0.0000205598  0.0000126902    1.62              0.10523
num_words              0.0000128799  0.0000050810    2.53              0.01126
bio_complete           0.0007872563  0.0000829403    9.49 < 0.0000000000000002
has_url               -0.0008243067  0.0003462860   -2.38              0.01731
hashtags_count         0.0001530731  0.0000410969    3.72              0.00020
has_hashtags          -0.0004265401  0.0001467212   -2.91              0.00365
contains_company_name -0.0000696217  0.0000794153   -0.88              0.38068
contains_words        -0.0002898654  0.0001008573   -2.87              0.00406
custom_content                   NA            NA      NA                   NA
days_joined           -0.0000001591  0.0000000333   -4.77   0.0000018279190980
pop_ind                0.0006594267  0.0001350129    4.88   0.0000010520152741
is_mon                 0.0001244445  0.0000616680    2.02              0.04362
is_tue                 0.0000997849  0.0000612495    1.63              0.10331
is_wed                 0.0000921444  0.0000612210    1.51              0.13232
is_thu                 0.0000648783  0.0000609921    1.06              0.28748
is_fri                 0.0000824642  0.0000610611    1.35              0.17687
is_sat                -0.0000116508  0.0000678082   -0.17              0.86358
```

log(atq)	-0.0000187019	0.0000186719	-1.00	0.31655
factor(sic)1531	-0.0018088162	0.0002294152	-7.88	0.0000000000000034
factor(sic)1623	0.0053867878	0.0003813709	14.12	< 0.0000000000000002
factor(sic)2080	-0.0000623282	0.0001738899	-0.36	0.72002
factor(sic)2300	-0.0004111932	0.0001937816	-2.12	0.03386
factor(sic)2320	0.0005876909	0.0003862939	1.52	0.12820
factor(sic)2510	-0.0008444849	0.0001813372	-4.66	0.0000032431234283
factor(sic)2621	-0.0003410435	0.0002164761	-1.58	0.11518
factor(sic)2810	0.0014684067	0.0001898183	7.74	0.0000000000000111
factor(sic)2820	-0.0009034200	0.0002573775	-3.51	0.00045
factor(sic)2836	0.0001140600	0.0001476541	0.77	0.43984
factor(sic)2844	-0.0002147294	0.0002445913	-0.88	0.38001
factor(sic)2851	-0.0010282833	0.0002261145	-4.55	0.0000054786186180
factor(sic)2911	0.0009861074	0.0002300847	4.29	0.0000183464915369
factor(sic)3490	-0.0015211031	0.0004559276	-3.34	0.00085
factor(sic)3561	-0.0008668541	0.0001935753	-4.48	0.0000075996372241
factor(sic)3576	-0.0000504810	0.0001724996	-0.29	0.76980
factor(sic)3674	-0.0007123848	0.0002233250	-3.19	0.00143
factor(sic)3711	-0.0274666200	0.0014747018	-18.63	< 0.0000000000000002
factor(sic)3724	-0.0005728553	0.0002046169	-2.80	0.00512
factor(sic)3812	0.0002254689	0.0002362697	0.95	0.33996
factor(sic)3823	-0.0003920480	0.0002211868	-1.77	0.07634
factor(sic)3826	-0.0002882554	0.0001673626	-1.72	0.08503
factor(sic)3843	-0.0005477327	0.0002081874	-2.63	0.00853
factor(sic)3845	-0.0012036422	0.0002508653	-4.80	0.0000016221652269
factor(sic)3990	-0.0009726853	0.0003675837	-2.65	0.00815
factor(sic)4213	0.0002633521	0.0004001503	0.66	0.51047
factor(sic)4833	0.0008991118	0.0001707645	5.27	0.0000001424413975
factor(sic)4841	-0.0010188781	0.0001471419	-6.92	0.0000000000045994
factor(sic)4911	-0.0013083916	0.0003472884	-3.77	0.00017
factor(sic)4931	0.0001634107	0.0001491117	1.10	0.27315
factor(sic)5047	-0.0006957344	0.0003029944	-2.30	0.02168
factor(sic)5065	-0.0004766536	0.0002802853	-1.70	0.08904
factor(sic)5140	-0.0005235602	0.0002350513	-2.23	0.02594
factor(sic)5150	0.0000217241	0.0001973800	0.11	0.91236
factor(sic)5211	-0.0010017953	0.0002213105	-4.53	0.0000060502255278
factor(sic)5661	-0.0022533117	0.0002864675	-7.87	0.0000000000000040
factor(sic)5734	-0.0022213841	0.0003595014	-6.18	0.0000000006656751
factor(sic)5812	-0.0013515568	0.0002441602	-5.54	0.0000000316796455
factor(sic)5912	-0.0025264614	0.0002056017	-12.29	< 0.0000000000000002
factor(sic)5940	-0.0013571833	0.0001804653	-7.52	0.0000000000000585
factor(sic)5944	-0.0003188264	0.0002656482	-1.20	0.23009
factor(sic)5961	-0.0017724732	0.0004050043	-4.38	0.0000121657562414
factor(sic)6141	-0.0007709515	0.0001499768	-5.14	0.0000002783569414
factor(sic)6200	-0.0029078776	0.0003080139	-9.44	< 0.0000000000000002
factor(sic)6211	-0.0020326685	0.0002634058	-7.72	0.0000000000000129
factor(sic)6282	-0.0017291954	0.0002438376	-7.09	0.0000000000014002
factor(sic)6311	0.0004917489	0.0002973167	1.65	0.09816
factor(sic)6324	-0.0007462618	0.0001728512	-4.32	0.0000159169396426
factor(sic)6331	-0.0007522876	0.0001611938	-4.67	0.0000030896949084
factor(sic)6798	-0.0021337105	0.0003248528	-6.57	0.0000000000530144
factor(sic)7310	-0.0008465798	0.0002352383	-3.60	0.00032
factor(sic)7323	-0.0003770885	0.0002016180	-1.87	0.06146
factor(sic)7370	-0.0010816320	0.0002430529	-4.45	0.0000086566158313

```
factor(sic)7372       -0.0006782155  0.0001868186   -3.63              0.00028
factor(sic)7389       -0.0004681029  0.0001774517   -2.64              0.00835
factor(sic)7900       -0.0019013409  0.0003357991   -5.66   0.0000000152926508
factor(sic)7990        0.0013765806  0.0002569936    5.36   0.0000000864255742
factor(sic)8700       -0.0110847422  0.0006694792  -16.56 < 0.0000000000000002
factor(sic)8721       -0.0001251478  0.0002086083   -0.60              0.54857
factor(sic)8731       -0.0009673599  0.0002172653   -4.45   0.0000085667433845
factor(sic)8742                  NA            NA      NA                   NA

(Intercept)           **
is_annouce            **
red_annouce           ***
no_annoucement        ***
TFF                   ***
coleman_liau          ***
flesch                ***
fog_info
num_words             *
bio_complete          ***
has_url               *
hashtags_count        ***
has_hashtags          **
contains_company_name
contains_words        **
custom_content
days_joined           ***
pop_ind               ***
is_mon                *
is_tue
is_wed
is_thu
is_fri
is_sat
log(atq)
factor(sic)1531       ***
factor(sic)1623       ***
factor(sic)2080
factor(sic)2300       *
factor(sic)2320
factor(sic)2510       ***
factor(sic)2621
factor(sic)2810       ***
factor(sic)2820       ***
factor(sic)2836
factor(sic)2844
factor(sic)2851       ***
factor(sic)2911       ***
factor(sic)3490       ***
factor(sic)3561       ***
factor(sic)3576
factor(sic)3674       **
factor(sic)3711       ***
factor(sic)3724       **
factor(sic)3812
```

```
factor(sic)3823       .
factor(sic)3826       .
factor(sic)3843       **
factor(sic)3845       ***
factor(sic)3990       **
factor(sic)4213
factor(sic)4833       ***
factor(sic)4841       ***
factor(sic)4911       ***
factor(sic)4931
factor(sic)5047       *
factor(sic)5065       .
factor(sic)5140       *
factor(sic)5150
factor(sic)5211       ***
factor(sic)5661       ***
factor(sic)5734       ***
factor(sic)5812       ***
factor(sic)5912       ***
factor(sic)5940       ***
factor(sic)5944
factor(sic)5961       ***
factor(sic)6141       ***
factor(sic)6200       ***
factor(sic)6211       ***
factor(sic)6282       ***
factor(sic)6311       .
factor(sic)6324       ***
factor(sic)6331       ***
factor(sic)6798       ***
factor(sic)7310       ***
factor(sic)7323       .
factor(sic)7370       ***
factor(sic)7372       ***
factor(sic)7389       **
factor(sic)7900       ***
factor(sic)7990       ***
factor(sic)8700       ***
factor(sic)8721
factor(sic)8731       ***
factor(sic)8742
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00154 on 12010 degrees of freedom
Multiple R-squared:  0.151, Adjusted R-squared:  0.145
F-statistic: 25.8 on 83 and 12010 DF,  p-value: <0.0000000000000002
```

```r
# run regression Model15 : non financial variables with industry FE and firm FE
Mod15_formula <- update(Mod13_formula, . ~ . + factor(gvkey) + factor(sic))
Model15 <- lm(Mod15_formula,data=Train)
summary(Model15)
```

```
Call:
lm(formula = Mod15_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03643 -0.00030 -0.00005  0.00015  0.04688 

Coefficients: (74 not defined because of singularities)
                       Estimate   Std. Error t value             Pr(>|t|)    
(Intercept)         0.023317552  0.001541134   15.13 < 0.0000000000000002 ***
is_annouce         -0.000192510  0.000079809   -2.41              0.01587    
red_annouce         0.001280814  0.000189922    6.74   0.0000000000161287 
no_annoucement      0.000099030  0.000029923    3.31              0.00094    
TFF                 0.000002243  0.000000339    6.63   0.0000000000360369 
coleman_liau       -0.000721723  0.000047276  -15.27 < 0.0000000000000002 
flesch             -0.000078963  0.000013142   -6.01   0.0000000019278566 
fog_info           -0.000390439  0.000070034   -5.57   0.0000000252902668 
num_words          -0.000059989  0.000017036   -3.52              0.00043    
bio_complete       -0.000663706  0.000374804   -1.77              0.07662    
has_url             0.002061997  0.000533477    3.87              0.00011    
hashtags_count     -0.006636952  0.000510808  -12.99 < 0.0000000000000002 
has_hashtags       -0.019053822  0.002047403   -9.31 < 0.0000000000000002 
contains_company_name  0.010898763  0.000839076   12.99 < 0.0000000000000002 
contains_words      0.012870867  0.001558975    8.26 < 0.0000000000000002 
custom_content               NA           NA      NA                   NA    
days_joined        -0.000001273  0.000000268   -4.75   0.0000020707613696 
pop_ind             0.018132570  0.001116313   16.24 < 0.0000000000000002 
is_mon              0.000126092  0.000060937    2.07              0.03855    
is_tue              0.000097643  0.000060518    1.61              0.10667    
is_wed              0.000089931  0.000060489    1.49              0.13711    
is_thu              0.000060310  0.000060260    1.00              0.31693    
is_fri              0.000085149  0.000060332    1.41              0.15817    
is_sat             -0.000009291  0.000066920   -0.14              0.88958    
log(atq)            0.000018923  0.000039435    0.48              0.63134    
factor(gvkey)1447  -0.012746612  0.001458066   -8.74 < 0.0000000000000002 
factor(gvkey)1487  -0.014486514  0.001761841   -8.22 < 0.0000000000000002 
factor(gvkey)1659  -0.001686738  0.000521755   -3.23              0.00123    
factor(gvkey)1878   0.018358733  0.002385217    7.70   0.0000000000000150 
factor(gvkey)1919   0.025223671  0.001625435   15.52 < 0.0000000000000002 
factor(gvkey)1920  -0.003851960  0.001984444   -1.94              0.05227    
factor(gvkey)3170   0.019333036  0.001521126   12.71 < 0.0000000000000002 
factor(gvkey)3226  -0.016731532  0.002063103   -8.11   0.0000000000000005 
factor(gvkey)3336  -0.014488869  0.002153273   -6.73   0.0000000000178894 
factor(gvkey)3708  -0.014690014  0.002830267   -5.19   0.0000002133552174 
factor(gvkey)3814   0.003154294  0.000561068    5.62   0.0000000193033573 
factor(gvkey)4060  -0.007054418  0.000643971  -10.95 < 0.0000000000000002 
factor(gvkey)4201  -0.033319536  0.002013003  -16.55 < 0.0000000000000002 
factor(gvkey)4321  -0.015420195  0.001088198  -14.17 < 0.0000000000000002 
factor(gvkey)4839  -0.155970844  0.026114676   -5.97   0.0000000024021168 
factor(gvkey)4988  -0.009760500  0.000957505  -10.19 < 0.0000000000000002 
factor(gvkey)5492   0.017315241  0.002229128    7.77   0.0000000000000086 
factor(gvkey)5742   0.016154612  0.000987271   16.36 < 0.0000000000000002 
factor(gvkey)6435   0.004890224  0.000467548   10.46 < 0.0000000000000002 
```

factor(gvkey)6788	-0.009023742	0.000550581	-16.39	< 0.0000000000000002
factor(gvkey)6829	0.036370745	0.002172706	16.74	< 0.0000000000000002
factor(gvkey)7163	0.025455119	0.001512431	16.83	< 0.0000000000000002
factor(gvkey)8113	0.018027244	0.001168713	15.42	< 0.0000000000000002
factor(gvkey)8358	-0.011059743	0.001272151	-8.69	< 0.0000000000000002
factor(gvkey)8402	-0.037546361	0.002742025	-13.69	< 0.0000000000000002
factor(gvkey)8463	0.005606257	0.000462792	12.11	< 0.0000000000000002
factor(gvkey)8479	-0.014284593	0.001042025	-13.71	< 0.0000000000000002
factor(gvkey)8549	-0.014363174	0.001447475	-9.92	< 0.0000000000000002
factor(gvkey)9599	0.033479555	0.001866473	17.94	< 0.0000000000000002
factor(gvkey)9667	0.011887687	0.002439885	4.87	0.00000111754892193
factor(gvkey)10247	0.016791854	0.001068686	15.71	< 0.0000000000000002
factor(gvkey)10983	0.010148395	0.000766295	13.24	< 0.0000000000000002
factor(gvkey)11115	-0.021950450	0.001377442	-15.94	< 0.0000000000000002
factor(gvkey)11264	-0.001249521	0.002015715	-0.62	0.53534
factor(gvkey)11304	0.001900862	0.001099498	1.73	0.08386
factor(gvkey)11584	-0.030516738	0.002239016	-13.63	< 0.0000000000000002
factor(gvkey)11669	0.039688425	0.003251318	12.21	< 0.0000000000000002
factor(gvkey)11770	-0.021400411	0.001308605	-16.35	< 0.0000000000000002
factor(gvkey)12441	-0.107279352	0.012483404	-8.59	< 0.0000000000000002
factor(gvkey)13092	-0.001468698	0.001038693	-1.41	0.15739
factor(gvkey)14412	0.012592187	0.000844080	14.92	< 0.0000000000000002
factor(gvkey)17035	-0.008198371	0.000571289	-14.35	< 0.0000000000000002
factor(gvkey)23225	-0.004090853	0.000430441	-9.50	< 0.0000000000000002
factor(gvkey)23252	0.000923488	0.000444514	2.08	0.03777
factor(gvkey)24468	-0.018633623	0.001434477	-12.99	< 0.0000000000000002
factor(gvkey)24800	-0.025392171	0.001524903	-16.65	< 0.0000000000000002
factor(gvkey)25124	-0.014328711	0.001449479	-9.89	< 0.0000000000000002
factor(gvkey)25340	0.009771748	0.003116486	3.14	0.00172
factor(gvkey)26011	0.009112361	0.001427898	6.38	0.00000000018162811
factor(gvkey)27794	0.016275258	0.002048813	7.94	0.00000000000000214
factor(gvkey)27914	0.015069372	0.001095640	13.75	< 0.0000000000000002
factor(gvkey)28034	0.024005634	0.001458652	16.46	< 0.0000000000000002
factor(gvkey)28216	-0.005844970	0.000985209	-5.93	0.00000000306177814
factor(gvkey)28303	0.017572150	0.001102511	15.94	< 0.0000000000000002
factor(gvkey)28629	-0.012729829	0.001692673	-7.52	0.00000000000005843
factor(gvkey)28924	0.007880074	0.000585908	13.45	< 0.0000000000000002
factor(gvkey)29011	0.016838277	0.001329473	12.67	< 0.0000000000000002
factor(gvkey)29241	-0.042299680	0.002989675	-14.15	< 0.0000000000000002
factor(gvkey)29612	-0.002266656	0.000603601	-3.76	0.00017
factor(gvkey)29710	0.001510643	0.000941474	1.60	0.10862
factor(gvkey)29751	0.001353590	0.001278302	1.06	0.28967
factor(gvkey)30138	0.016129219	0.001514662	10.65	< 0.0000000000000002
factor(gvkey)61494	0.020421766	0.002048121	9.97	< 0.0000000000000002
factor(gvkey)65772	-0.013799996	0.000918827	-15.02	< 0.0000000000000002
factor(gvkey)116504	0.008116567	0.000705890	11.50	< 0.0000000000000002
factor(gvkey)118502	0.078667272	0.004962359	15.85	< 0.0000000000000002
factor(gvkey)121718	-0.039194587	0.002574099	-15.23	< 0.0000000000000002
factor(gvkey)122841	-0.028048667	0.001791550	-15.66	< 0.0000000000000002
factor(gvkey)126554	0.012084045	0.000748850	16.14	< 0.0000000000000002
factor(gvkey)136725	-0.006640784	0.001125859	-5.90	0.00000000376903155
factor(gvkey)137131	0.009612562	0.001439006	6.68	0.00000000002495011
factor(gvkey)141384	0.022831914	0.001529013	14.93	< 0.0000000000000002
factor(gvkey)145049	-0.024819829	0.001531925	-16.20	< 0.0000000000000002

factor(gvkey)148470	NA	NA	NA	NA
factor(gvkey)150139	0.004704963	0.000635544	7.40	0.00000000000014199
factor(gvkey)156617	-0.026237419	0.002442547	-10.74	< 0.0000000000000002
factor(gvkey)160211	-0.006455679	0.000750690	-8.60	< 0.0000000000000002
factor(gvkey)164416	NA	NA	NA	NA
factor(gvkey)164664	NA	NA	NA	NA
factor(gvkey)165746	NA	NA	NA	NA
factor(gvkey)170841	NA	NA	NA	NA
factor(gvkey)174317	NA	NA	NA	NA
factor(gvkey)177376	NA	NA	NA	NA
factor(gvkey)178493	NA	NA	NA	NA
factor(gvkey)179666	NA	NA	NA	NA
factor(gvkey)180402	NA	NA	NA	NA
factor(gvkey)180646	NA	NA	NA	NA
factor(gvkey)184500	NA	NA	NA	NA
factor(sic)1531	NA	NA	NA	NA
factor(sic)1623	NA	NA	NA	NA
factor(sic)2080	NA	NA	NA	NA
factor(sic)2300	NA	NA	NA	NA
factor(sic)2320	NA	NA	NA	NA
factor(sic)2510	NA	NA	NA	NA
factor(sic)2621	NA	NA	NA	NA
factor(sic)2810	NA	NA	NA	NA
factor(sic)2820	NA	NA	NA	NA
factor(sic)2836	NA	NA	NA	NA
factor(sic)2844	NA	NA	NA	NA
factor(sic)2851	NA	NA	NA	NA
factor(sic)2911	NA	NA	NA	NA
factor(sic)3490	NA	NA	NA	NA
factor(sic)3561	NA	NA	NA	NA
factor(sic)3576	NA	NA	NA	NA
factor(sic)3674	NA	NA	NA	NA
factor(sic)3711	NA	NA	NA	NA
factor(sic)3724	NA	NA	NA	NA
factor(sic)3812	NA	NA	NA	NA
factor(sic)3823	NA	NA	NA	NA
factor(sic)3826	NA	NA	NA	NA
factor(sic)3843	NA	NA	NA	NA
factor(sic)3845	NA	NA	NA	NA
factor(sic)3990	NA	NA	NA	NA
factor(sic)4213	NA	NA	NA	NA
factor(sic)4833	NA	NA	NA	NA
factor(sic)4841	NA	NA	NA	NA
factor(sic)4911	NA	NA	NA	NA
factor(sic)4931	NA	NA	NA	NA
factor(sic)5047	NA	NA	NA	NA
factor(sic)5065	NA	NA	NA	NA
factor(sic)5140	NA	NA	NA	NA
factor(sic)5150	NA	NA	NA	NA
factor(sic)5211	NA	NA	NA	NA
factor(sic)5661	NA	NA	NA	NA
factor(sic)5734	NA	NA	NA	NA
factor(sic)5812	NA	NA	NA	NA
factor(sic)5912	NA	NA	NA	NA

factor(sic)5940	NA	NA	NA	NA
factor(sic)5944	NA	NA	NA	NA
factor(sic)5961	NA	NA	NA	NA
factor(sic)6141	NA	NA	NA	NA
factor(sic)6200	NA	NA	NA	NA
factor(sic)6211	NA	NA	NA	NA
factor(sic)6282	NA	NA	NA	NA
factor(sic)6311	NA	NA	NA	NA
factor(sic)6324	NA	NA	NA	NA
factor(sic)6331	NA	NA	NA	NA
factor(sic)6798	NA	NA	NA	NA
factor(sic)7310	NA	NA	NA	NA
factor(sic)7323	NA	NA	NA	NA
factor(sic)7370	NA	NA	NA	NA
factor(sic)7372	NA	NA	NA	NA
factor(sic)7389	NA	NA	NA	NA
factor(sic)7900	NA	NA	NA	NA
factor(sic)7990	NA	NA	NA	NA
factor(sic)8700	NA	NA	NA	NA
factor(sic)8721	NA	NA	NA	NA
factor(sic)8731	NA	NA	NA	NA
factor(sic)8742	NA	NA	NA	NA

(Intercept)	***
is_annouce	*
red_annouce	***
no_annoucement	***
TFF	***
coleman_liau	***
flesch	***
fog_info	***
num_words	***
bio_complete	.
has_url	***
hashtags_count	***
has_hashtags	***
contains_company_name	***
contains_words	***
custom_content	
days_joined	***
pop_ind	***
is_mon	*
is_tue	
is_wed	
is_thu	
is_fri	
is_sat	
log(atq)	
factor(gvkey)1447	***
factor(gvkey)1487	***
factor(gvkey)1659	**
factor(gvkey)1878	***
factor(gvkey)1919	***
factor(gvkey)1920	.

```
factor(gvkey)3170     ***
factor(gvkey)3226     ***
factor(gvkey)3336     ***
factor(gvkey)3708     ***
factor(gvkey)3814     ***
factor(gvkey)4060     ***
factor(gvkey)4201     ***
factor(gvkey)4321     ***
factor(gvkey)4839     ***
factor(gvkey)4988     ***
factor(gvkey)5492     ***
factor(gvkey)5742     ***
factor(gvkey)6435     ***
factor(gvkey)6788     ***
factor(gvkey)6829     ***
factor(gvkey)7163     ***
factor(gvkey)8113     ***
factor(gvkey)8358     ***
factor(gvkey)8402     ***
factor(gvkey)8463     ***
factor(gvkey)8479     ***
factor(gvkey)8549     ***
factor(gvkey)9599     ***
factor(gvkey)9667     ***
factor(gvkey)10247    ***
factor(gvkey)10983    ***
factor(gvkey)11115    ***
factor(gvkey)11264
factor(gvkey)11304    .
factor(gvkey)11584    ***
factor(gvkey)11669    ***
factor(gvkey)11770    ***
factor(gvkey)12441    ***
factor(gvkey)13092
factor(gvkey)14412    ***
factor(gvkey)17035    ***
factor(gvkey)23225    ***
factor(gvkey)23252    *
factor(gvkey)24468    ***
factor(gvkey)24800    ***
factor(gvkey)25124    ***
factor(gvkey)25340    **
factor(gvkey)26011    ***
factor(gvkey)27794    ***
factor(gvkey)27914    ***
factor(gvkey)28034    ***
factor(gvkey)28216    ***
factor(gvkey)28303    ***
factor(gvkey)28629    ***
factor(gvkey)28924    ***
factor(gvkey)29011    ***
factor(gvkey)29241    ***
factor(gvkey)29612    ***
factor(gvkey)29710
```

```
factor(gvkey)29751
factor(gvkey)30138    ***
factor(gvkey)61494    ***
factor(gvkey)65772    ***
factor(gvkey)116504   ***
factor(gvkey)118502   ***
factor(gvkey)121718   ***
factor(gvkey)122841   ***
factor(gvkey)126554   ***
factor(gvkey)136725   ***
factor(gvkey)137131   ***
factor(gvkey)141384   ***
factor(gvkey)145049   ***
factor(gvkey)148470
factor(gvkey)150139   ***
factor(gvkey)156617   ***
factor(gvkey)160211   ***
factor(gvkey)164416
factor(gvkey)164664
factor(gvkey)165746
factor(gvkey)170841
factor(gvkey)174317
factor(gvkey)177376
factor(gvkey)178493
factor(gvkey)179666
factor(gvkey)180402
factor(gvkey)180646
factor(gvkey)184500
factor(sic)1531
factor(sic)1623
factor(sic)2080
factor(sic)2300
factor(sic)2320
factor(sic)2510
factor(sic)2621
factor(sic)2810
factor(sic)2820
factor(sic)2836
factor(sic)2844
factor(sic)2851
factor(sic)2911
factor(sic)3490
factor(sic)3561
factor(sic)3576
factor(sic)3674
factor(sic)3711
factor(sic)3724
factor(sic)3812
factor(sic)3823
factor(sic)3826
factor(sic)3843
factor(sic)3845
factor(sic)3990
factor(sic)4213
```

```
factor(sic)4833
factor(sic)4841
factor(sic)4911
factor(sic)4931
factor(sic)5047
factor(sic)5065
factor(sic)5140
factor(sic)5150
factor(sic)5211
factor(sic)5661
factor(sic)5734
factor(sic)5812
factor(sic)5912
factor(sic)5940
factor(sic)5944
factor(sic)5961
factor(sic)6141
factor(sic)6200
factor(sic)6211
factor(sic)6282
factor(sic)6311
factor(sic)6324
factor(sic)6331
factor(sic)6798
factor(sic)7310
factor(sic)7323
factor(sic)7370
factor(sic)7372
factor(sic)7389
factor(sic)7900
factor(sic)7990
factor(sic)8700
factor(sic)8721
factor(sic)8731
factor(sic)8742
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00152 on 11994 degrees of freedom
Multiple R-squared:  0.175, Adjusted R-squared:  0.168
F-statistic: 25.7 on 99 and 11994 DF,  p-value: <0.0000000000000002
```

6.2 In Sample Evaluation

The growth rates are predicted using 3 different models (Model1 through Model3) and stored in separate variables (`Train$Pre_rate_1` through `Train$Pre_rate_3`) within the `Train` dataset. These predicted growth rates are then used to estimate the followers for each model, considering the lagged followers and the predicted growth rates. The root mean square error (RMSE) function (`rmse()`) calculates the accuracy of each model by comparing the actual followers with the predicted followers, offering an in-sample accuracy assessment. Finally, the RMSE values for each model are computed.

```r
# predict Train followers growth rate
Train$Pre_rate_13 <- predict(Model13, Train)
Train$Pre_rate_14 <- predict(Model14, Train)
Train$Pre_rate_15 <- predict(Model15, Train)

# compute Train followers using predicted growth rate
Train$Followers_M13 <- (1+Train$Pre_rate_13)*Train$followers_lag
Train$Followers_M14 <- (1+Train$Pre_rate_14)*Train$followers_lag
Train$Followers_M15 <- (1+Train$Pre_rate_15)*Train$followers_lag


## compute In Sample accuracy
rmse <- function(v1, v2) {
  sqrt(mean((v1 - v2)^2, na.rm = T))
}

rmse <- c(rmse(Train$followers,Train$Followers_M13),
          rmse(Train$followers,Train$Followers_M14),
          rmse(Train$followers,Train$Followers_M15))

# retrieve RMSE for all the five models
names(rmse) <- c("Model13", "Model14", "Model15")
rmse
```

```
Model13 Model14 Model15
  984.1   901.4   904.5
```

6.3 Prediction of Test Followers

Using the same calculation procedures in financial model part, we retrieve the last day's number of followers from the training data for each company (`gvkey`). It sorts the data by `gvkey` and date in descending order to obtain the latest date's followers. Afterward, it filters out specific `gvkey` values.

Subsequently, it joins the last day's followers with the testing data using a full join based on `gvkey` and `date`. This ensures that each company's last day followers from the training data are paired with corresponding rows in the testing data.

Then, it predicts growth rates for the testing data using Model1 (`Model1`). If there are missing values in the `followers` column of the testing data, the code iteratively calculates the followers using the growth rate and the lagged followers value until there are no missing values left. This ensures that missing follower values are filled in based on the predicted growth rate.

```r
# use the function for 3 non financial predicted rates
Test13 <- predict_test(last_day_followers, Test, Model13)
Test14 <- predict_test(last_day_followers, Test, Model14)
Test15 <- predict_test(last_day_followers, Test, Model15)
```

We merge the six specific companies with the rest of the dataset. The merged data should have 6403 observations, which aligns with the format of submission sample. The follower counts are rounded to ensure integer values. From the resulting dataset, only the columns for ID and followers are selected, representing the necessary data for submission. This subset of data is saved as a CSV file named “submission_M1_fin.csv”, excluding row numbers.

```
# merge the six companies with other companies 6492
Test_submission13 <- Test13 %>% select(gvkey, date, followers, ID)
Test_submission13 <- rbind(Test_submission13, Test_6)

# final submission data should return 6403 observations
Test_submission13 <- Test_submission13 %>% filter(date>=20170701)
Test_submission13$followers <- round(Test_submission13$followers)
Followers_submisson_M13 <- Test_submission13 %>% select(ID, followers)

# merge the six companies with other companies 6492
Test_submission14 <- Test14 %>% select(gvkey, date, followers, ID)
Test_submission14 <- rbind(Test_submission14, Test_6)

# final submission data should return 6403 observations
Test_submission14 <- Test_submission14 %>% filter(date>=20170701)
Test_submission14$followers <- round(Test_submission14$followers)
Followers_submisson_M14 <- Test_submission14 %>% select(ID, followers)

# merge the six companies with other companies 6492
Test_submission15 <- Test15 %>% select(gvkey, date, followers, ID)
Test_submission15 <- rbind(Test_submission15, Test_6)

# final submission data should return 6403 observations
Test_submission15 <- Test_submission15 %>% filter(date>=20170701)
Test_submission15$followers <- round(Test_submission15$followers)
Followers_submisson_M15 <- Test_submission15 %>% select(ID, followers)

# save the sample of submission
write.csv(Followers_submisson_M13, "submission_M13.csv", row.names = FALSE)
write.csv(Followers_submisson_M14, "submission_M14.csv", row.names = FALSE)
write.csv(Followers_submisson_M15, "submission_M15.csv", row.names = FALSE)
```

6.4 Out of Sample Evaluation

The submission CSV files for the 3 models have been exported, and we still find that model 13 for the non-financial model gives better score. This is the model that gives us the best result for now, which makes us rank 3*rd* on the leaderboard temporarily.

submission_M15.csv Complete · now	**5544.19647**
submission_M14.csv Complete · 20s ago	**5038.20615**
submission_M13.csv Complete · 1m ago	**3088.92331**

7. Combined Model

7.1 Combined Model Construction

In this section, we combined all variables (stock price, financial ratios, announcement and profile variables) to construct a regression model for prediction of Twitter followers growth rate.

The 5 combined model constructed are as follows:

combined_Model1 : The response variable "growth_rate" is regressed against various financial ratios, stock price, announcements, and profile variables. The log-transformed value of `atq` is included as a variable to account for varying company size.

combined_Model2 : The response variable "growth_rate" is regressed against the lagged values of financial ratios and stock price variables. The model aims to predict `growth_rate` based on the previous period's values of these variables, along with the announcement and profile variables.

combined_Model3 : The removal of variables with potentially high multicollinearity is performed in this model. Specifically, `opmad`, `npm`, `quick_ratio`, `cash_ratio`, `ps`, `ptb`, and `roce` have been excluded from the formula. The removal of highly correlated variables is done with the expectation to reduce error term in the model thus improving accuracy.

combined_Model4 : Industry and firm fixed effects are added to combined_Model1

combined_Model5 : Industry and firm fixed effects are added to combined_Model2

```
#create formula

#combined_Mod1 : financial ratios, stock price , announcement, profile variables combined
Mod16_formula<- as.formula("growth_rate ~ p_daily_change + volatile + p_daily_change_lag + vol

#combined_Mod2 : financial ratios (lag), stock price (lag), announcement, profile variables co
Mod17_formula <- as.formula("growth_rate ~ bm_lag + pe_exi_lag + ps_lag + pcf_lag + dpr_lag +

#combined_Mod3 : remove variables with high multicollinearity (remove opmad,npm,quick ratio,ca
Mod18_formula <- as.formula("growth_rate ~ p_daily_change + volatile + p_daily_change_lag + vo
```

Using the above model formulaes created, run regression for the 5 combined model stated.

```
#run regression for combined_Mod1
Model16 <- lm(Mod16_formula, data = Train)
summary(Model16)
```

```
Call:
lm(formula = Mod16_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03526 -0.00048 -0.00017  0.00020  0.04803 

Coefficients: (1 not defined because of singularities)
                         Estimate     Std. Error t value
(Intercept)            0.00020793746  0.00026445021    0.79
```

p_daily_change	-0.00002522293	0.00060009324	-0.04
volatile	0.00247535049	0.00106088981	2.33
p_daily_change_lag	-0.00032853957	0.00061487197	-0.53
volatile_lag	0.00174278093	0.00103666595	1.68
bm	0.00025510608	0.00006750935	3.78
pe_exi	0.00000002346	0.00000009372	0.25
ps	-0.00004114538	0.00001169001	-3.52
pcf	0.00000154201	0.00000078062	1.98
dpr	-0.00000011590	0.00000768983	-0.02
npm	0.00234420129	0.00051178821	4.58
opmbd	0.00188190375	0.00038674657	4.87
opmad	-0.00136445421	0.00052677447	-2.59
gpm	0.00021548906	0.00013478987	1.60
roa	-0.00392121658	0.00052644837	-7.45
roe	-0.00002327457	0.00000951743	-2.45
roce	0.00078267317	0.00030914492	2.53
debt_at	0.00150594732	0.00018370550	8.20
de_ratio	-0.00000005511	0.00000095179	-0.06
intcov	-0.00000001513	0.00000003624	-0.42
cash_ratio	-0.00008520928	0.00005958915	-1.43
quick_ratio	0.00005801073	0.00006733604	0.86
curr_ratio	-0.00006194917	0.00004313088	-1.44
inv_turn	0.00000286856	0.00000023405	12.26
at_turn	0.00020056134	0.00005444694	3.68
debt_assets	-0.00130184083	0.00018952636	-6.87
ptb	0.00000018281	0.00000008136	2.25
rect_turn	-0.00000336373	0.00000108611	-3.10
sale_nwc	-0.00000015026	0.00000007426	-2.02
divyield	-0.01118267442	0.00137924882	-8.11
is_annouce	-0.00019667670	0.00008373633	-2.35
red_annouce	0.00112730954	0.00019950493	5.65
no_annoucement	0.00007521600	0.00003141952	2.39
TFF	-0.00000000623	0.00000000196	-3.18
coleman_liau	-0.00000349941	0.00000141503	-2.47
flesch	0.00001039087	0.00000223431	4.65
fog_info	0.00002978387	0.00000687055	4.34
num_words	0.00001199977	0.00000293871	4.08
bio_complete	0.00054271300	0.00005311011	10.22
has_url	0.00003667946	0.00006978093	0.53
hashtags_count	0.00001869264	0.00001888495	0.99
has_hashtags	-0.00008113784	0.00006204668	-1.31
contains_company_name	-0.00007252038	0.00004862344	-1.49
contains_words	-0.00038375995	0.00006106910	-6.28
custom_content	NA	NA	NA
days_joined	0.00000006644	0.00000001689	3.93
pop_ind	0.00036320894	0.00004337451	8.37
is_mon	0.00017526444	0.00006377736	2.75
is_tue	0.00016839397	0.00006331612	2.66
is_wed	0.00016220816	0.00006325394	2.56
is_thu	0.00011925315	0.00006302551	1.89
is_fri	0.00014130888	0.00006312846	2.24
is_sat	-0.00001309483	0.00007060235	-0.19
log(atq)	-0.00004452885	0.00001193323	-3.73

Pr(>|t|)

```
(Intercept)                           0.43171
p_daily_change                        0.96647
volatile                              0.01965 *
p_daily_change_lag                    0.59313
volatile_lag                          0.09276 .
bm                                    0.00016 ***
pe_exi                                0.80234
ps                                    0.00043 ***
pcf                                   0.04825 *
dpr                                   0.98798
npm                      0.00000468715212797 ***
opmbd                    0.00000115330129080 ***
opmad                                 0.00960 **
gpm                                   0.10991
roa                      0.00000000000010090 ***
roe                                   0.01448 *
roce                                  0.01136 *
debt_at                  0.00000000000000027 ***
de_ratio                              0.95383
intcov                                0.67630
cash_ratio                            0.15276
quick_ratio                           0.38897
curr_ratio                            0.15094
inv_turn               < 0.0000000000000002 ***
at_turn                               0.00023 ***
debt_assets              0.00000000000678745 ***
ptb                                   0.02466 *
rect_turn                             0.00196 **
sale_nwc                              0.04304 *
divyield                 0.00000000000000057 ***
is_annouce                            0.01885 *
red_annouce              0.00000001635786412 ***
no_annoucement                        0.01668 *
TFF                                   0.00149 **
coleman_liau                          0.01341 *
flesch                   0.00000334508476512 ***
fog_info                 0.00001469404462107 ***
num_words                0.00004467951260749 ***
bio_complete           < 0.0000000000000002 ***
has_url                               0.59915
hashtags_count                        0.32228
has_hashtags                          0.19100
contains_company_name                 0.13586
contains_words           0.00000000034129632 ***
custom_content                             NA
days_joined              0.00008408266192469 ***
pop_ind                < 0.0000000000000002 ***
is_mon                                0.00600 **
is_tue                                0.00783 **
is_wed                                0.01035 *
is_thu                                0.05850 .
is_fri                                0.02521 *
is_sat                                0.85286
log(atq)                              0.00019 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00161 on 12041 degrees of freedom
Multiple R-squared:  0.0756,    Adjusted R-squared:  0.0716 
F-statistic: 18.9 on 52 and 12041 DF,  p-value: <0.0000000000000002
```

```
#run regression for combined_Mod2
Model17 <- lm(Mod17_formula, data = Train)
summary(Model17)
```

```

Call:
lm(formula = Mod17_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.03520 -0.00048 -0.00017  0.00020  0.04811 

Coefficients: (1 not defined because of singularities)
                          Estimate     Std. Error t value
(Intercept)            0.00051439649  0.00026533476    1.94
bm_lag                 0.00026394897  0.00006911299    3.82
pe_exi_lag             0.00000004184  0.00000008079    0.52
ps_lag                -0.00003972028  0.00001225039   -3.24
pcf_lag                0.00000031516  0.00000077891    0.40
dpr_lag                0.00000687191  0.00001216877    0.56
npm_lag                0.00183253316  0.00048765573    3.76
opmbd_lag              0.00202915920  0.00040140954    5.06
opmad_lag             -0.00108807697  0.00052385355   -2.08
gpm_lag                0.00024270584  0.00013609847    1.78
roa_lag               -0.00450270763  0.00054230234   -8.30
roe_lag               -0.00002388606  0.00000950580   -2.51
roce_lag               0.00108917746  0.00031247049    3.49
debt_at_lag            0.00177928024  0.00018536106    9.60
de_ratio_lag          -0.00000028546  0.00000095539   -0.30
intcov_lag            -0.00000001741  0.00000003325   -0.52
cash_ratio_lag        -0.00009102956  0.00005868861   -1.55
quick_ratio_lag        0.00004212293  0.00006835005    0.62
curr_ratio_lag        -0.00006541045  0.00004548727   -1.44
inv_turn_lag           0.00000259316  0.00000021824   11.88
at_turn_lag            0.00022983757  0.00005742250    4.00
debt_assets_lag       -0.00156634681  0.00019189363   -8.16
ptb_lag                0.00000018713  0.00000008080    2.32
rect_turn_lag         -0.00000320898  0.00000111798   -2.87
sale_nwc_lag          -0.00000015532  0.00000008770   -1.77
divyield_lag          -0.01283288083  0.00157544984   -8.15
p_daily_change_lag    -0.00029629126  0.00059138107   -0.50
volatile_lag           0.00333887832  0.00083980693    3.98
is_annouce            -0.00019989662  0.00008378558   -2.39
red_annouce            0.00113796781  0.00019963954    5.70
no_annoucement         0.00008126963  0.00003139702    2.59
TFF                   -0.00000000552  0.00000000199   -2.77
```

```
coleman_liau          -0.00000317663  0.00000141251   -2.25
flesch                 0.00000875656  0.00000222783    3.93
fog_info               0.00002444354  0.00000689281    3.55
num_words              0.00001379325  0.00000294217    4.69
bio_complete           0.00055581130  0.00005462425   10.18
has_url               -0.00001183720  0.00006998454   -0.17
hashtags_count         0.00001370984  0.00001908170    0.72
has_hashtags          -0.00009965494  0.00006260844   -1.59
contains_company_name -0.00010768968  0.00004876074   -2.21
contains_words        -0.00040622327  0.00006149928   -6.61
custom_content                    NA             NA      NA
days_joined            0.00000006951  0.00000001708    4.07
pop_ind                0.00040493991  0.00004353854    9.30
is_mon                 0.00016958785  0.00006380023    2.66
is_tue                 0.00016548784  0.00006333292    2.61
is_wed                 0.00015781033  0.00006328140    2.49
is_thu                 0.00011431596  0.00006305006    1.81
is_fri                 0.00013705963  0.00006314965    2.17
is_sat                -0.00001759827  0.00007063166   -0.25
log(atq)              -0.00004548336  0.00001200832   -3.79
                                   Pr(>|t|)
(Intercept)                         0.05256 .
bm_lag                              0.00013 ***
pe_exi_lag                          0.60452
ps_lag                              0.00119 **
pcf_lag                             0.68577
dpr_lag                             0.57228
npm_lag                             0.00017 ***
opmbd_lag              0.00000043651852996 ***
opmad_lag                           0.03782 *
gpm_lag                             0.07456 .
roa_lag               < 0.0000000000000002 ***
roe_lag                             0.01199 *
roce_lag                            0.00049 ***
debt_at_lag           < 0.0000000000000002 ***
de_ratio_lag                        0.76511
intcov_lag                          0.60057
cash_ratio_lag                      0.12091
quick_ratio_lag                     0.53772
curr_ratio_lag                      0.15046
inv_turn_lag          < 0.0000000000000002 ***
at_turn_lag            0.00006303365285236 ***
debt_assets_lag        0.00000000000000036 ***
ptb_lag                             0.02057 *
rect_turn_lag                       0.00411 **
sale_nwc_lag                        0.07657 .
divyield_lag           0.00000000000000041 ***
p_daily_change_lag                  0.61637
volatile_lag           0.00007056199287447 ***
is_annouce                          0.01706 *
red_annouce            0.00000001225393383 ***
no_annoucement                      0.00965 **
TFF                                 0.00553 **
coleman_liau                        0.02454 *
```

```
flesch                    0.00008523037022712 ***
fog_info                              0.00039 ***
num_words                 0.00000278752101384 ***
bio_complete            < 0.0000000000000002 ***
has_url                               0.86569
hashtags_count                        0.47247
has_hashtags                          0.11147
contains_company_name                 0.02723 *
contains_words            0.00000000004133100 ***
custom_content                             NA
days_joined               0.00004729893273680 ***
pop_ind                 < 0.0000000000000002 ***
is_mon                                0.00787 **
is_tue                                0.00899 **
is_wed                                0.01265 *
is_thu                                0.06984 .
is_fri                                0.03000 *
is_sat                                0.80324
log(atq)                              0.00015 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00161 on 12043 degrees of freedom
Multiple R-squared:  0.0745,    Adjusted R-squared:  0.0707
F-statistic: 19.4 on 50 and 12043 DF,  p-value: <0.0000000000000002
```

```r
#run regression for combined_Mod3
Model18 <- lm(Mod18_formula, data = Train)
summary(Model18)
```

```
Call:
lm(formula = Mod18_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.03509 -0.00048 -0.00017  0.00020  0.04820

Coefficients: (1 not defined because of singularities)
                          Estimate     Std. Error t value
(Intercept)           -0.00011080225  0.00023740149   -0.47
p_daily_change         0.00000527934  0.00060105927    0.01
volatile               0.00198569874  0.00106112894    1.87
p_daily_change_lag    -0.00030670029  0.00061617184   -0.50
volatile_lag           0.00141681596  0.00103791860    1.37
bm                     0.00026973449  0.00006643863    4.06
pe_exi                 0.00000007241  0.00000008792    0.82
pcf                    0.00000036995  0.00000074675    0.50
dpr                   -0.00000779045  0.00000753906   -1.03
opmbd                  0.00260451033  0.00023093631   11.28
gpm                   -0.00004532516  0.00011519654   -0.39
roa                   -0.00311648251  0.00026750824  -11.65
roe                   -0.00000247373  0.00000189856   -1.30
```

```
debt_at                0.00110777115  0.00014140678    7.83
de_ratio              -0.00000037273  0.00000093484   -0.40
intcov                -0.00000002817  0.00000003583   -0.79
curr_ratio            -0.00001233791  0.00001907166   -0.65
inv_turn               0.00000291526  0.00000023072   12.64
at_turn                0.00036050288  0.00004906440    7.35
debt_assets           -0.00088325548  0.00013197010   -6.69
rect_turn             -0.00000309727  0.00000095279   -3.25
sale_nwc              -0.00000014936  0.00000007361   -2.03
divyield              -0.00942861285  0.00132244288   -7.13
is_annouce            -0.00019933983  0.00008382843   -2.38
red_annouce            0.00112826950  0.00019991295    5.64
no_annoucement         0.00007280304  0.00003145039    2.31
TFF                   -0.00000000264  0.00000000185   -1.43
coleman_liau          -0.00000234946  0.00000140297   -1.67
flesch                 0.00000586247  0.00000208024    2.82
fog_info               0.00001676780  0.00000656176    2.56
num_words              0.00001523880  0.00000275904    5.52
bio_complete           0.00048866418  0.00005073942    9.63
has_url                0.00004583168  0.00006863486    0.67
hashtags_count         0.00000305008  0.00001779428    0.17
has_hashtags          -0.00001046527  0.00005758248   -0.18
contains_company_name -0.00004228084  0.00004660862   -0.91
contains_words        -0.00031111689  0.00005850022   -5.32
custom_content                    NA             NA      NA
days_joined            0.00000007286  0.00000001628    4.48
pop_ind                0.00036723386  0.00004123876    8.91
is_mon                 0.00017581833  0.00006388128    2.75
is_tue                 0.00017221718  0.00006340889    2.72
is_wed                 0.00016575397  0.00006335186    2.62
is_thu                 0.00012312777  0.00006311443    1.95
is_fri                 0.00014545830  0.00006322098    2.30
is_sat                -0.00001242264  0.00007075456   -0.18
log(atq)              -0.00004441053  0.00001176465   -3.77
                                  Pr(>|t|)
(Intercept)                        0.64070
p_daily_change                     0.99299
volatile                           0.06133 .
p_daily_change_lag                 0.61867
volatile_lag                       0.17226
bm                      0.0000494031911758 ***
pe_exi                             0.41017
pcf                                0.62032
dpr                                0.30146
opmbd                 < 0.0000000000000002 ***
gpm                                0.69399
roa                   < 0.0000000000000002 ***
roe                                0.19262
debt_at                 0.0000000000000051 ***
de_ratio                           0.69012
intcov                             0.43180
curr_ratio                         0.51769
inv_turn              < 0.0000000000000002 ***
at_turn                 0.0000000000002149 ***
```

```
debt_assets            0.0000000000228574 ***
rect_turn                         0.00115 **
sale_nwc                          0.04248 *
divyield               0.0000000000010634 ***
is_annouce                        0.01742 *
red_annouce            0.0000000170087581 ***
no_annoucement                    0.02064 *
TFF                               0.15351
coleman_liau                      0.09403 .
flesch                            0.00484 **
fog_info                          0.01062 *
num_words              0.0000000339715921 ***
bio_complete         < 0.0000000000000002 ***
has_url                           0.50430
hashtags_count                    0.86391
has_hashtags                      0.85579
contains_company_name             0.36435
contains_words         0.0000001066633380 ***
custom_content                         NA
days_joined            0.0000077019684285 ***
pop_ind              < 0.0000000000000002 ***
is_mon                            0.00593 **
is_tue                            0.00662 **
is_wed                            0.00890 **
is_thu                            0.05110 .
is_fri                            0.02142 *
is_sat                            0.86063
log(atq)                          0.00016 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00161 on 12048 degrees of freedom
Multiple R-squared:  0.0709,    Adjusted R-squared:  0.0674
F-statistic: 20.4 on 45 and 12048 DF,  p-value: <0.0000000000000002
```

```r
#run combined_Mod1 + industry FE + firm FE
Mod19_formula <- update(Mod17_formula, .~. + factor(gvkey) + factor(sic))
Model19 <- lm(Mod19_formula, data = Train)
summary(Model19)
```

```
Call:
lm(formula = Mod19_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.03654 -0.00030 -0.00005  0.00017  0.04678

Coefficients: (74 not defined because of singularities)
                          Estimate    Std. Error t value             Pr(>|t|)
(Intercept)           0.0246521104  0.0033743085    7.31   0.0000000000002931
bm_lag                0.0005367015  0.0003801188    1.41              0.15800
pe_exi_lag            0.0000000603  0.0000001158    0.52              0.60232
```

ps_lag	-0.0000113557	0.0000306973	-0.37	0.71145
pcf_lag	-0.0000008644	0.0000009338	-0.93	0.35466
dpr_lag	-0.0000125554	0.0000199984	-0.63	0.53013
npm_lag	0.0018510680	0.0010312893	1.79	0.07269
opmbd_lag	-0.0073305362	0.0021972790	-3.34	0.00085
opmad_lag	0.0043397142	0.0017177552	2.53	0.01154
gpm_lag	-0.0003873199	0.0008517616	-0.45	0.64931
roa_lag	-0.0013882278	0.0026054281	-0.53	0.59417
roe_lag	-0.0000126470	0.0000112935	-1.12	0.26280
roce_lag	0.0001942063	0.0009570399	0.20	0.83920
debt_at_lag	-0.0016819652	0.0011095219	-1.52	0.12956
de_ratio_lag	-0.0000002465	0.0000009421	-0.26	0.79363
intcov_lag	0.0000000138	0.0000000493	0.28	0.77965
cash_ratio_lag	-0.0005622834	0.0002362854	-2.38	0.01734
quick_ratio_lag	0.0002659181	0.0004755763	0.56	0.57607
curr_ratio_lag	0.0003774921	0.0003903897	0.97	0.33358
inv_turn_lag	0.0000004908	0.0000003759	1.31	0.19173
at_turn_lag	-0.0002867298	0.0004109635	-0.70	0.48538
debt_assets_lag	0.0020572477	0.0009934115	2.07	0.03839
ptb_lag	0.0000001002	0.0000000921	1.09	0.27646
rect_turn_lag	0.0000030501	0.0000045158	0.68	0.49942
sale_nwc_lag	0.0000000560	0.0000001085	0.52	0.60573
divyield_lag	0.0058275359	0.0055363117	1.05	0.29254
p_daily_change_lag	-0.0002683089	0.0005613108	-0.48	0.63266
volatile_lag	0.0015794959	0.0009051635	1.74	0.08101
is_annouce	-0.0001926855	0.0000798103	-2.41	0.01578
red_annouce	0.0012967761	0.0001899318	6.83	0.0000000000090528
no_annoucement	0.0001009191	0.0000299181	3.37	0.00075
TFF	0.0000018573	0.0000004387	4.23	0.0000231860554265
coleman_liau	-0.0007687577	0.0001019833	-7.54	0.0000000000000511
flesch	-0.0001051873	0.0000269332	-3.91	0.0000945518153590
fog_info	-0.0004364105	0.0000863498	-5.05	0.0000004390792472
num_words	-0.0000514837	0.0000488618	-1.05	0.29206
bio_complete	0.0010683554	0.0014758109	0.72	0.46913
has_url	0.0010299409	0.0010583447	0.97	0.33049
hashtags_count	-0.0074737819	0.0013574918	-5.51	0.0000000375538816
has_hashtags	-0.0199092409	0.0030919642	-6.44	0.0000000001248415
contains_company_name	0.0116034859	0.0019134034	6.06	0.0000000013651152
contains_words	0.0141764675	0.0038301264	3.70	0.00022
custom_content	NA	NA	NA	NA
days_joined	-0.0000010881	0.0000003049	-3.57	0.00036
pop_ind	0.0213422608	0.0032659687	6.53	0.0000000000663047
is_mon	0.0001224505	0.0000609014	2.01	0.04439
is_tue	0.0000923248	0.0000604943	1.53	0.12699
is_wed	0.0000818267	0.0000604606	1.35	0.17596
is_thu	0.0000517828	0.0000602348	0.86	0.38998
is_fri	0.0000787887	0.0000603093	1.31	0.19144
is_sat	-0.0000069972	0.0000668650	-0.10	0.91666
log(atq)	0.0000117711	0.0000449765	0.26	0.79354
factor(gvkey)1447	-0.0133112821	0.0019412224	-6.86	0.0000000000073695
factor(gvkey)1487	-0.0185325093	0.0039536604	-4.69	0.0000027970923088
factor(gvkey)1659	-0.0014669815	0.0008029076	-1.83	0.06771
factor(gvkey)1878	0.0204059568	0.0039091407	5.22	0.0000001818588429
factor(gvkey)1919	0.0262679072	0.0043840930	5.99	0.0000000021370912

factor(gvkey)1920	-0.0059766161	0.0027963504	-2.14	0.03259
factor(gvkey)3170	0.0200430337	0.0027497954	7.29	0.0000000000003321
factor(gvkey)3226	-0.0144422358	0.0026024847	-5.55	0.0000000292743983
factor(gvkey)3336	-0.0182402175	0.0043004706	-4.24	0.0000223753396609
factor(gvkey)3708	-0.0116844767	0.0034689797	-3.37	0.00076
factor(gvkey)3814	0.0036523408	0.0015000090	2.43	0.01491
factor(gvkey)4060	-0.0090863497	0.0013139921	-6.92	0.0000000000049141
factor(gvkey)4201	-0.0360555063	0.0048942239	-7.37	0.0000000000001860
factor(gvkey)4321	-0.0159962652	0.0021578749	-7.41	0.0000000000001318
factor(gvkey)4839	-0.1256077648	0.0341423459	-3.68	0.00024
factor(gvkey)4988	-0.0098136340	0.0018396441	-5.33	0.0000000975390255
factor(gvkey)5492	0.0174771097	0.0034295982	5.10	0.0000003522622755
factor(gvkey)5742	0.0185981700	0.0029007070	6.41	0.0000000001494171
factor(gvkey)6435	0.0064361796	0.0017234704	3.73	0.00019
factor(gvkey)6788	-0.0066072496	0.0018856194	-3.50	0.00046
factor(gvkey)6829	0.0409299598	0.0070746230	5.79	0.0000000074124896
factor(gvkey)7163	0.0295265843	0.0045540796	6.48	0.0000000000931141
factor(gvkey)8113	0.0204607481	0.0025697377	7.96	0.0000000000000018
factor(gvkey)8358	-0.0116969926	0.0020230026	-5.78	0.0000000075664394
factor(gvkey)8402	-0.0400384695	0.0058095480	-6.89	0.0000000000057840
factor(gvkey)8463	0.0074247627	0.0017934512	4.14	0.0000349793304022
factor(gvkey)8479	-0.0143584318	0.0024979370	-5.75	0.0000000092446995
factor(gvkey)8549	-0.0125582369	0.0023980410	-5.24	0.0000001660894441
factor(gvkey)9599	0.0341546342	0.0054261369	6.29	0.0000000003192418
factor(gvkey)9667	0.0142906225	0.0039039946	3.66	0.00025
factor(gvkey)10247	0.0197248395	0.0044096940	4.47	0.0000077817684699
factor(gvkey)10983	0.0118670111	0.0025544474	4.65	0.0000034265895994
factor(gvkey)11115	-0.0234257157	0.0042818491	-5.47	0.0000000456674552
factor(gvkey)11264	0.0031874831	0.0035255992	0.90	0.36596
factor(gvkey)11304	0.0022468822	0.0017300979	1.30	0.19407
factor(gvkey)11584	-0.0331013564	0.0043043578	-7.69	0.0000000000000158
factor(gvkey)11669	0.0427417042	0.0065681310	6.51	0.0000000000794933
factor(gvkey)11770	-0.0267792219	0.0037828829	-7.08	0.0000000000015326
factor(gvkey)12441	-0.0977842028	0.0163712068	-5.97	0.0000000023963714
factor(gvkey)13092	-0.0018706206	0.0018614535	-1.00	0.31495
factor(gvkey)14412	0.0148409345	0.0028078793	5.29	0.0000001275935674
factor(gvkey)17035	-0.0085904195	0.0007317892	-11.74	< 0.0000000000000002
factor(gvkey)23225	-0.0016832018	0.0007122273	-2.36	0.01813
factor(gvkey)23252	0.0023863192	0.0011577967	2.06	0.03932
factor(gvkey)24468	-0.0178709251	0.0029032622	-6.16	0.0000000007725307
factor(gvkey)24800	-0.0277225946	0.0046767799	-5.93	0.0000000031565617
factor(gvkey)25124	-0.0139455208	0.0023610888	-5.91	0.0000000035917979
factor(gvkey)25340	0.0136527119	0.0051936078	2.63	0.00858
factor(gvkey)26011	0.0102744651	0.0020039872	5.13	0.0000002989747767
factor(gvkey)27794	0.0148849184	0.0031794735	4.68	0.0000028781671585
factor(gvkey)27914	0.0154841836	0.0031095710	4.98	0.0000006462950906
factor(gvkey)28034	0.0253891489	0.0037496248	6.77	0.0000000000133767
factor(gvkey)28216	-0.0116090463	0.0025557553	-4.54	0.0000056183719679
factor(gvkey)28303	0.0189087463	0.0041086663	4.60	0.0000042242768903
factor(gvkey)28629	-0.0131838400	0.0031182856	-4.23	0.0000237621099598
factor(gvkey)28924	0.0065450931	0.0026013800	2.52	0.01188
factor(gvkey)29011	0.0191205628	0.0020387437	9.38	< 0.0000000000000002
factor(gvkey)29241	-0.0476271639	0.0084290961	-5.65	0.0000000163796964
factor(gvkey)29612	-0.0040544852	0.0020530852	-1.97	0.04831

factor(gvkey)29710	0.0020274119	0.0012606308	1.61	0.10781
factor(gvkey)29751	0.0005591389	0.0027347319	0.20	0.83800
factor(gvkey)30138	0.0176132178	0.0026962741	6.53	0.0000000000673333
factor(gvkey)61494	0.0206993454	0.0038547240	5.37	0.0000000802695213
factor(gvkey)65772	-0.0154995836	0.0025907720	-5.98	0.0000000022586444
factor(gvkey)116504	0.0104995533	0.0024493595	4.29	0.0000182805853323
factor(gvkey)118502	0.0853594033	0.0118799583	7.19	0.0000000000007111
factor(gvkey)121718	-0.0432022884	0.0066762233	-6.47	0.0000000001011102
factor(gvkey)122841	-0.0313559106	0.0047370404	-6.62	0.0000000000376306
factor(gvkey)126554	0.0143291679	0.0023353392	6.14	0.0000000008740986
factor(gvkey)136725	-0.0079844836	0.0023911890	-3.34	0.00084
factor(gvkey)137131	0.0093521019	0.0020808533	4.49	0.0000070434747711
factor(gvkey)141384	0.0262668725	0.0040310173	6.52	0.0000000000750086
factor(gvkey)145049	-0.0285452220	0.0048347379	-5.90	0.0000000036399509
factor(gvkey)148470	NA	NA	NA	NA
factor(gvkey)150139	0.0064299627	0.0021986386	2.92	0.00346
factor(gvkey)156617	-0.0289378422	0.0057533678	-5.03	0.0000004983106393
factor(gvkey)160211	-0.0056470714	0.0023758026	-2.38	0.01747
factor(gvkey)164416	NA	NA	NA	NA
factor(gvkey)164664	NA	NA	NA	NA
factor(gvkey)165746	NA	NA	NA	NA
factor(gvkey)170841	NA	NA	NA	NA
factor(gvkey)174317	NA	NA	NA	NA
factor(gvkey)177376	NA	NA	NA	NA
factor(gvkey)178493	NA	NA	NA	NA
factor(gvkey)179666	NA	NA	NA	NA
factor(gvkey)180402	NA	NA	NA	NA
factor(gvkey)180646	NA	NA	NA	NA
factor(gvkey)184500	NA	NA	NA	NA
factor(sic)1531	NA	NA	NA	NA
factor(sic)1623	NA	NA	NA	NA
factor(sic)2080	NA	NA	NA	NA
factor(sic)2300	NA	NA	NA	NA
factor(sic)2320	NA	NA	NA	NA
factor(sic)2510	NA	NA	NA	NA
factor(sic)2621	NA	NA	NA	NA
factor(sic)2810	NA	NA	NA	NA
factor(sic)2820	NA	NA	NA	NA
factor(sic)2836	NA	NA	NA	NA
factor(sic)2844	NA	NA	NA	NA
factor(sic)2851	NA	NA	NA	NA
factor(sic)2911	NA	NA	NA	NA
factor(sic)3490	NA	NA	NA	NA
factor(sic)3561	NA	NA	NA	NA
factor(sic)3576	NA	NA	NA	NA
factor(sic)3674	NA	NA	NA	NA
factor(sic)3711	NA	NA	NA	NA
factor(sic)3724	NA	NA	NA	NA
factor(sic)3812	NA	NA	NA	NA
factor(sic)3823	NA	NA	NA	NA
factor(sic)3826	NA	NA	NA	NA
factor(sic)3843	NA	NA	NA	NA
factor(sic)3845	NA	NA	NA	NA
factor(sic)3990	NA	NA	NA	NA

```
factor(sic)4213                 NA            NA      NA                    NA
factor(sic)4833                 NA            NA      NA                    NA
factor(sic)4841                 NA            NA      NA                    NA
factor(sic)4911                 NA            NA      NA                    NA
factor(sic)4931                 NA            NA      NA                    NA
factor(sic)5047                 NA            NA      NA                    NA
factor(sic)5065                 NA            NA      NA                    NA
factor(sic)5140                 NA            NA      NA                    NA
factor(sic)5150                 NA            NA      NA                    NA
factor(sic)5211                 NA            NA      NA                    NA
factor(sic)5661                 NA            NA      NA                    NA
factor(sic)5734                 NA            NA      NA                    NA
factor(sic)5812                 NA            NA      NA                    NA
factor(sic)5912                 NA            NA      NA                    NA
factor(sic)5940                 NA            NA      NA                    NA
factor(sic)5944                 NA            NA      NA                    NA
factor(sic)5961                 NA            NA      NA                    NA
factor(sic)6141                 NA            NA      NA                    NA
factor(sic)6200                 NA            NA      NA                    NA
factor(sic)6211                 NA            NA      NA                    NA
factor(sic)6282                 NA            NA      NA                    NA
factor(sic)6311                 NA            NA      NA                    NA
factor(sic)6324                 NA            NA      NA                    NA
factor(sic)6331                 NA            NA      NA                    NA
factor(sic)6798                 NA            NA      NA                    NA
factor(sic)7310                 NA            NA      NA                    NA
factor(sic)7323                 NA            NA      NA                    NA
factor(sic)7370                 NA            NA      NA                    NA
factor(sic)7372                 NA            NA      NA                    NA
factor(sic)7389                 NA            NA      NA                    NA
factor(sic)7900                 NA            NA      NA                    NA
factor(sic)7990                 NA            NA      NA                    NA
factor(sic)8700                 NA            NA      NA                    NA
factor(sic)8721                 NA            NA      NA                    NA
factor(sic)8731                 NA            NA      NA                    NA
factor(sic)8742                 NA            NA      NA                    NA

(Intercept)           ***
bm_lag
pe_exi_lag
ps_lag
pcf_lag
dpr_lag
npm_lag               .
opmbd_lag             ***
opmad_lag             *
gpm_lag
roa_lag
roe_lag
roce_lag
debt_at_lag
de_ratio_lag
intcov_lag
cash_ratio_lag        *
```

```
quick_ratio_lag
curr_ratio_lag
inv_turn_lag
at_turn_lag
debt_assets_lag       *
ptb_lag
rect_turn_lag
sale_nwc_lag
divyield_lag
p_daily_change_lag
volatile_lag          .
is_annouce            *
red_annouce           ***
no_annoucement        ***
TFF                   ***
coleman_liau          ***
flesch                ***
fog_info              ***
num_words
bio_complete
has_url
hashtags_count        ***
has_hashtags          ***
contains_company_name ***
contains_words        ***
custom_content
days_joined           ***
pop_ind               ***
is_mon                *
is_tue
is_wed
is_thu
is_fri
is_sat
log(atq)
factor(gvkey)1447     ***
factor(gvkey)1487     ***
factor(gvkey)1659     .
factor(gvkey)1878     ***
factor(gvkey)1919     ***
factor(gvkey)1920     *
factor(gvkey)3170     ***
factor(gvkey)3226     ***
factor(gvkey)3336     ***
factor(gvkey)3708     ***
factor(gvkey)3814     *
factor(gvkey)4060     ***
factor(gvkey)4201     ***
factor(gvkey)4321     ***
factor(gvkey)4839     ***
factor(gvkey)4988     ***
factor(gvkey)5492     ***
factor(gvkey)5742     ***
factor(gvkey)6435     ***
```

factor(gvkey)6788	***
factor(gvkey)6829	***
factor(gvkey)7163	***
factor(gvkey)8113	***
factor(gvkey)8358	***
factor(gvkey)8402	***
factor(gvkey)8463	***
factor(gvkey)8479	***
factor(gvkey)8549	***
factor(gvkey)9599	***
factor(gvkey)9667	***
factor(gvkey)10247	***
factor(gvkey)10983	***
factor(gvkey)11115	***
factor(gvkey)11264	
factor(gvkey)11304	
factor(gvkey)11584	***
factor(gvkey)11669	***
factor(gvkey)11770	***
factor(gvkey)12441	***
factor(gvkey)13092	
factor(gvkey)14412	***
factor(gvkey)17035	***
factor(gvkey)23225	*
factor(gvkey)23252	*
factor(gvkey)24468	***
factor(gvkey)24800	***
factor(gvkey)25124	***
factor(gvkey)25340	**
factor(gvkey)26011	***
factor(gvkey)27794	***
factor(gvkey)27914	***
factor(gvkey)28034	***
factor(gvkey)28216	***
factor(gvkey)28303	***
factor(gvkey)28629	***
factor(gvkey)28924	*
factor(gvkey)29011	***
factor(gvkey)29241	***
factor(gvkey)29612	*
factor(gvkey)29710	
factor(gvkey)29751	
factor(gvkey)30138	***
factor(gvkey)61494	***
factor(gvkey)65772	***
factor(gvkey)116504	***
factor(gvkey)118502	***
factor(gvkey)121718	***
factor(gvkey)122841	***
factor(gvkey)126554	***
factor(gvkey)136725	***
factor(gvkey)137131	***
factor(gvkey)141384	***
factor(gvkey)145049	***

```
factor(gvkey)148470
factor(gvkey)150139   **
factor(gvkey)156617   ***
factor(gvkey)160211   *
factor(gvkey)164416
factor(gvkey)164664
factor(gvkey)165746
factor(gvkey)170841
factor(gvkey)174317
factor(gvkey)177376
factor(gvkey)178493
factor(gvkey)179666
factor(gvkey)180402
factor(gvkey)180646
factor(gvkey)184500
factor(sic)1531
factor(sic)1623
factor(sic)2080
factor(sic)2300
factor(sic)2320
factor(sic)2510
factor(sic)2621
factor(sic)2810
factor(sic)2820
factor(sic)2836
factor(sic)2844
factor(sic)2851
factor(sic)2911
factor(sic)3490
factor(sic)3561
factor(sic)3576
factor(sic)3674
factor(sic)3711
factor(sic)3724
factor(sic)3812
factor(sic)3823
factor(sic)3826
factor(sic)3843
factor(sic)3845
factor(sic)3990
factor(sic)4213
factor(sic)4833
factor(sic)4841
factor(sic)4911
factor(sic)4931
factor(sic)5047
factor(sic)5065
factor(sic)5140
factor(sic)5150
factor(sic)5211
factor(sic)5661
factor(sic)5734
factor(sic)5812
factor(sic)5912
```

```
factor(sic)5940
factor(sic)5944
factor(sic)5961
factor(sic)6141
factor(sic)6200
factor(sic)6211
factor(sic)6282
factor(sic)6311
factor(sic)6324
factor(sic)6331
factor(sic)6798
factor(sic)7310
factor(sic)7323
factor(sic)7370
factor(sic)7372
factor(sic)7389
factor(sic)7900
factor(sic)7990
factor(sic)8700
factor(sic)8721
factor(sic)8731
factor(sic)8742
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00152 on 11967 degrees of freedom
Multiple R-squared:  0.178, Adjusted R-squared:  0.17
F-statistic: 20.6 on 126 and 11967 DF,  p-value: <0.0000000000000002
```

```r
#run combined_Mod2 + industry FE + firm FE
Mod20_formula <- update(Mod18_formula, .~. + factor(gvkey) + factor(sic))
Model20 <- lm(Mod20_formula, data = Train)
summary(Model20)
```

```
Call:
lm(formula = Mod20_formula, data = Train)

Residuals:
     Min       1Q   Median       3Q      Max
-0.03674 -0.00031 -0.00005  0.00017  0.04657

Coefficients: (74 not defined because of singularities)
                        Estimate     Std. Error t value
(Intercept)           0.02271722366  0.00223662589   10.16
p_daily_change        0.00032995147  0.00058555494    0.56
volatile              0.00136536005  0.00105085487    1.30
p_daily_change_lag   -0.00029662020  0.00058236933   -0.51
volatile_lag          0.00098725739  0.00101883107    0.97
bm                    0.00128665664  0.00032180846    4.00
pe_exi                0.00000003715  0.00000011067    0.34
pcf                  -0.00000029235  0.00000082080   -0.36
dpr                   0.00000542895  0.00000878047    0.62
```

opmbd	-0.00165826639	0.00092005716	-1.80
gpm	-0.00135443210	0.00076855361	-1.76
roa	-0.00025883881	0.00130255621	-0.20
roe	-0.00000000589	0.00000212081	0.00
debt_at	-0.00301863217	0.00096036379	-3.14
de_ratio	0.00000003599	0.00000093129	0.04
intcov	0.00000004241	0.00000004539	0.93
curr_ratio	0.00035129686	0.00010058337	3.49
inv_turn	0.00000051287	0.00000041046	1.25
at_turn	-0.00017765802	0.00022467133	-0.79
debt_assets	0.00287305170	0.00072602081	3.96
rect_turn	0.00000067494	0.00000376475	0.18
sale_nwc	0.00000009701	0.00000009768	0.99
divyield	0.00107736431	0.00358503754	0.30
is_annouce	-0.00018784357	0.00007979848	-2.35
red_annouce	0.00129688407	0.00018985309	6.83
no_annoucement	0.00009505939	0.00002993178	3.18
TFF	0.00000196987	0.00000040439	4.87
coleman_liau	-0.00074638805	0.00006714274	-11.12
flesch	-0.00008750096	0.00001840792	-4.75
fog_info	-0.00041787258	0.00007988363	-5.23
num_words	-0.00005749427	0.00003165634	-1.82
bio_complete	-0.00005977541	0.00115106036	-0.05
has_url	0.00149491914	0.00092632409	1.61
hashtags_count	-0.00710047992	0.00087015166	-8.16
has_hashtags	-0.01862452829	0.00255223794	-7.30
contains_company_name	0.01099781867	0.00128390794	8.57
contains_words	0.01395113746	0.00249411847	5.59
custom_content	NA	NA	NA
days_joined	-0.00000094341	0.00000029055	-3.25
pop_ind	0.02000750116	0.00198006362	10.10
is_mon	0.00012438637	0.00006088030	2.04
is_tue	0.00009649072	0.00006046423	1.60
is_wed	0.00008481292	0.00006043398	1.40
is_thu	0.00005701578	0.00006020302	0.95
is_fri	0.00008162869	0.00006028950	1.35
is_sat	-0.00000550438	0.00006684846	-0.08
log(atq)	0.00000735379	0.00004150650	0.18
factor(gvkey)1447	-0.01283777389	0.00162404892	-7.90
factor(gvkey)1487	-0.01893607116	0.00310036504	-6.11
factor(gvkey)1659	-0.00219968607	0.00070398960	-3.12
factor(gvkey)1878	0.01852253023	0.00314726533	5.89
factor(gvkey)1919	0.02523436645	0.00268459849	9.40
factor(gvkey)1920	-0.00511764299	0.00236398755	-2.16
factor(gvkey)3170	0.01935470674	0.00209415225	9.24
factor(gvkey)3226	-0.01519174550	0.00228840536	-6.64
factor(gvkey)3336	-0.01689976350	0.00332205426	-5.09
factor(gvkey)3708	-0.01244110765	0.00317243111	-3.92
factor(gvkey)3814	0.00333836897	0.00119557758	2.79
factor(gvkey)4060	-0.00859536169	0.00105105969	-8.18
factor(gvkey)4201	-0.03435641775	0.00299319507	-11.48
factor(gvkey)4321	-0.01520064586	0.00142152859	-10.69
factor(gvkey)4839	-0.13445474259	0.03141485889	-4.28
factor(gvkey)4988	-0.00895294136	0.00124846881	-7.17

factor(gvkey)5492	0.01740442928	0.00273449796	6.36
factor(gvkey)5742	0.01672720661	0.00176705135	9.47
factor(gvkey)6435	0.00616975149	0.00114844658	5.37
factor(gvkey)6788	-0.00870612117	0.00093853380	-9.28
factor(gvkey)6829	0.03881459950	0.00416461181	9.32
factor(gvkey)7163	0.02849213128	0.00286473005	9.95
factor(gvkey)8113	0.01913642049	0.00168790345	11.34
factor(gvkey)8358	-0.01066329339	0.00177696508	-6.00
factor(gvkey)8402	-0.03812665963	0.00385736354	-9.88
factor(gvkey)8463	0.00726051625	0.00114468014	6.34
factor(gvkey)8479	-0.01340837933	0.00154358887	-8.69
factor(gvkey)8549	-0.01458080023	0.00173346575	-8.41
factor(gvkey)9599	0.03370346470	0.00297976908	11.31
factor(gvkey)9667	0.01283698076	0.00315096219	4.07
factor(gvkey)10247	0.01877236922	0.00251597947	7.46
factor(gvkey)10983	0.01083568217	0.00153342702	7.07
factor(gvkey)11115	-0.02239794619	0.00246538247	-9.08
factor(gvkey)11264	0.00173247773	0.00283762426	0.61
factor(gvkey)11304	0.00127258029	0.00134774978	0.94
factor(gvkey)11584	-0.03195211390	0.00346434658	-9.22
factor(gvkey)11669	0.03961213899	0.00452705322	8.75
factor(gvkey)11770	-0.02535712596	0.00275665193	-9.20
factor(gvkey)12441	-0.09959081646	0.01462855537	-6.81
factor(gvkey)13092	-0.00208890452	0.00120438988	-1.73
factor(gvkey)14412	0.01420238422	0.00168415016	8.43
factor(gvkey)17035	-0.00860509235	0.00062296330	-13.81
factor(gvkey)23225	-0.00232386337	0.00055725305	-4.17
factor(gvkey)23252	0.00233270527	0.00099033974	2.36
factor(gvkey)24468	-0.01689274066	0.00194462369	-8.69
factor(gvkey)24800	-0.02616226241	0.00271567884	-9.63
factor(gvkey)25124	-0.01331631504	0.00170816021	-7.80
factor(gvkey)25340	0.01160946041	0.00403708524	2.88
factor(gvkey)26011	0.00912609066	0.00185692573	4.91
factor(gvkey)27794	0.01339290989	0.00257878075	5.19
factor(gvkey)27914	0.01508991755	0.00170132675	8.87
factor(gvkey)28034	0.02340910081	0.00229912858	10.18
factor(gvkey)28216	-0.01059835243	0.00211062887	-5.02
factor(gvkey)28303	0.01765476407	0.00230636456	7.65
factor(gvkey)28629	-0.01171275300	0.00238152464	-4.92
factor(gvkey)28924	0.00609919191	0.00112877316	5.40
factor(gvkey)29011	0.01794850627	0.00170367173	10.54
factor(gvkey)29241	-0.04627647080	0.00495881693	-9.33
factor(gvkey)29612	-0.00537589553	0.00119374230	-4.50
factor(gvkey)29710	0.00076427065	0.00114312775	0.67
factor(gvkey)29751	0.00088836493	0.00224627855	0.40
factor(gvkey)30138	0.01634251107	0.00211009937	7.74
factor(gvkey)61494	0.01976288858	0.00274217024	7.21
factor(gvkey)65772	-0.01317876259	0.00195046238	-6.76
factor(gvkey)116504	0.00932623771	0.00154085787	6.05
factor(gvkey)118502	0.08256506512	0.00756288622	10.92
factor(gvkey)121718	-0.04112146982	0.00405253312	-10.15
factor(gvkey)122841	-0.02921904948	0.00291306885	-10.03
factor(gvkey)126554	0.01304075494	0.00149409951	8.73
factor(gvkey)136725	-0.00713889692	0.00157793493	-4.52

factor(gvkey)137131	0.00974177578	0.00186372007	5.23
factor(gvkey)141384	0.02495430002	0.00264661704	9.43
factor(gvkey)145049	-0.02714167554	0.00292045359	-9.29
factor(gvkey)148470	NA	NA	NA
factor(gvkey)150139	0.00525965772	0.00123532059	4.26
factor(gvkey)156617	-0.02806547465	0.00376083220	-7.46
factor(gvkey)160211	-0.00492364516	0.00179173350	-2.75
factor(gvkey)164416	NA	NA	NA
factor(gvkey)164664	NA	NA	NA
factor(gvkey)165746	NA	NA	NA
factor(gvkey)170841	NA	NA	NA
factor(gvkey)174317	NA	NA	NA
factor(gvkey)177376	NA	NA	NA
factor(gvkey)178493	NA	NA	NA
factor(gvkey)179666	NA	NA	NA
factor(gvkey)180402	NA	NA	NA
factor(gvkey)180646	NA	NA	NA
factor(gvkey)184500	NA	NA	NA
factor(sic)1531	NA	NA	NA
factor(sic)1623	NA	NA	NA
factor(sic)2080	NA	NA	NA
factor(sic)2300	NA	NA	NA
factor(sic)2320	NA	NA	NA
factor(sic)2510	NA	NA	NA
factor(sic)2621	NA	NA	NA
factor(sic)2810	NA	NA	NA
factor(sic)2820	NA	NA	NA
factor(sic)2836	NA	NA	NA
factor(sic)2844	NA	NA	NA
factor(sic)2851	NA	NA	NA
factor(sic)2911	NA	NA	NA
factor(sic)3490	NA	NA	NA
factor(sic)3561	NA	NA	NA
factor(sic)3576	NA	NA	NA
factor(sic)3674	NA	NA	NA
factor(sic)3711	NA	NA	NA
factor(sic)3724	NA	NA	NA
factor(sic)3812	NA	NA	NA
factor(sic)3823	NA	NA	NA
factor(sic)3826	NA	NA	NA
factor(sic)3843	NA	NA	NA
factor(sic)3845	NA	NA	NA
factor(sic)3990	NA	NA	NA
factor(sic)4213	NA	NA	NA
factor(sic)4833	NA	NA	NA
factor(sic)4841	NA	NA	NA
factor(sic)4911	NA	NA	NA
factor(sic)4931	NA	NA	NA
factor(sic)5047	NA	NA	NA
factor(sic)5065	NA	NA	NA
factor(sic)5140	NA	NA	NA
factor(sic)5150	NA	NA	NA
factor(sic)5211	NA	NA	NA
factor(sic)5661	NA	NA	NA

```
factor(sic)5734                    NA              NA       NA
factor(sic)5812                    NA              NA       NA
factor(sic)5912                    NA              NA       NA
factor(sic)5940                    NA              NA       NA
factor(sic)5944                    NA              NA       NA
factor(sic)5961                    NA              NA       NA
factor(sic)6141                    NA              NA       NA
factor(sic)6200                    NA              NA       NA
factor(sic)6211                    NA              NA       NA
factor(sic)6282                    NA              NA       NA
factor(sic)6311                    NA              NA       NA
factor(sic)6324                    NA              NA       NA
factor(sic)6331                    NA              NA       NA
factor(sic)6798                    NA              NA       NA
factor(sic)7310                    NA              NA       NA
factor(sic)7323                    NA              NA       NA
factor(sic)7370                    NA              NA       NA
factor(sic)7372                    NA              NA       NA
factor(sic)7389                    NA              NA       NA
factor(sic)7900                    NA              NA       NA
factor(sic)7990                    NA              NA       NA
factor(sic)8700                    NA              NA       NA
factor(sic)8721                    NA              NA       NA
factor(sic)8731                    NA              NA       NA
factor(sic)8742                    NA              NA       NA
                                  Pr(>|t|)
(Intercept)            < 0.0000000000000002 ***
p_daily_change                      0.57312
volatile                            0.19387
p_daily_change_lag                  0.61053
volatile_lag                        0.33256
bm                       0.00006420741532630 ***
pe_exi                              0.73711
pcf                                 0.72172
dpr                                 0.53639
opmbd                               0.07152 .
gpm                                 0.07804 .
roa                                 0.84249
roe                                 0.99778
debt_at                             0.00168 **
de_ratio                            0.96917
intcov                              0.35009
curr_ratio                          0.00048 ***
inv_turn                            0.21150
at_turn                             0.42911
debt_assets              0.00007625268952497 ***
rect_turn                           0.85772
sale_nwc                            0.32064
divyield                            0.76379
is_annouce                          0.01859 *
red_annouce              0.00000000000884150 ***
no_annoucement                      0.00150 **
TFF                      0.00000112344973997 ***
coleman_liau           < 0.0000000000000002 ***
```

```
flesch                 0.00000202310754748 ***
fog_info               0.00000017142581144 ***
num_words                          0.06936 .
bio_complete                       0.95858
has_url                            0.10659
hashtags_count         0.00000000000000037 ***
has_hashtags           0.00000000000031207 ***
contains_company_name < 0.0000000000000002 ***
contains_words         0.00000002272679845 ***
custom_content                          NA
days_joined                        0.00117 **
pop_ind               < 0.0000000000000002 ***
is_mon                             0.04106 *
is_tue                             0.11055
is_wed                             0.16052
is_thu                             0.34363
is_fri                             0.17578
is_sat                             0.93438
log(atq)                           0.85938
factor(gvkey)1447      0.00000000000000292 ***
factor(gvkey)1487      0.00000000104217321 ***
factor(gvkey)1659                  0.00178 **
factor(gvkey)1878      0.00000000408029829 ***
factor(gvkey)1919     < 0.0000000000000002 ***
factor(gvkey)1920                  0.03042 *
factor(gvkey)3170     < 0.0000000000000002 ***
factor(gvkey)3226      0.00000000003304129 ***
factor(gvkey)3336      0.00000036900289236 ***
factor(gvkey)3708      0.00008844185719615 ***
factor(gvkey)3814                  0.00524 **
factor(gvkey)4060      0.00000000000000032 ***
factor(gvkey)4201     < 0.0000000000000002 ***
factor(gvkey)4321     < 0.0000000000000002 ***
factor(gvkey)4839      0.00001883703554444 ***
factor(gvkey)4988      0.00000000000078755 ***
factor(gvkey)5492      0.00000000020273447 ***
factor(gvkey)5742     < 0.0000000000000002 ***
factor(gvkey)6435      0.00000007921283721 ***
factor(gvkey)6788     < 0.0000000000000002 ***
factor(gvkey)6829     < 0.0000000000000002 ***
factor(gvkey)7163     < 0.0000000000000002 ***
factor(gvkey)8113     < 0.0000000000000002 ***
factor(gvkey)8358      0.00000000201968990 ***
factor(gvkey)8402     < 0.0000000000000002 ***
factor(gvkey)8463      0.00000000023370089 ***
factor(gvkey)8479     < 0.0000000000000002 ***
factor(gvkey)8549     < 0.0000000000000002 ***
factor(gvkey)9599     < 0.0000000000000002 ***
factor(gvkey)9667      0.00004651287915249 ***
factor(gvkey)10247     0.00000000000009162 ***
factor(gvkey)10983     0.00000000000167924 ***
factor(gvkey)11115    < 0.0000000000000002 ***
factor(gvkey)11264                 0.54152
factor(gvkey)11304                 0.34507
```

```
factor(gvkey)11584    < 0.0000000000000002 ***
factor(gvkey)11669    < 0.0000000000000002 ***
factor(gvkey)11770    < 0.0000000000000002 ***
factor(gvkey)12441     0.00000000001037108 ***
factor(gvkey)13092                 0.08287 .
factor(gvkey)14412    < 0.0000000000000002 ***
factor(gvkey)17035    < 0.0000000000000002 ***
factor(gvkey)23225     0.00003064605171836 ***
factor(gvkey)23252                 0.01852 *
factor(gvkey)24468    < 0.0000000000000002 ***
factor(gvkey)24800    < 0.0000000000000002 ***
factor(gvkey)25124     0.00000000000000693 ***
factor(gvkey)25340                 0.00404 **
factor(gvkey)26011     0.00000090131479516 ***
factor(gvkey)27794     0.00000020975865727 ***
factor(gvkey)27914    < 0.0000000000000002 ***
factor(gvkey)28034    < 0.0000000000000002 ***
factor(gvkey)28216     0.00000052029479640 ***
factor(gvkey)28303     0.00000000000002085 ***
factor(gvkey)28629     0.00000088515440430 ***
factor(gvkey)28924     0.00000006664882414 ***
factor(gvkey)29011    < 0.0000000000000002 ***
factor(gvkey)29241    < 0.0000000000000002 ***
factor(gvkey)29612     0.00000675080884027 ***
factor(gvkey)29710                 0.50378
factor(gvkey)29751                 0.69249
factor(gvkey)30138     0.00000000000001034 ***
factor(gvkey)61494     0.00000000000060622 ***
factor(gvkey)65772     0.00000000001476735 ***
factor(gvkey)116504    0.00000000146766036 ***
factor(gvkey)118502   < 0.0000000000000002 ***
factor(gvkey)121718   < 0.0000000000000002 ***
factor(gvkey)122841   < 0.0000000000000002 ***
factor(gvkey)126554   < 0.0000000000000002 ***
factor(gvkey)136725    0.00000612071383546 ***
factor(gvkey)137131    0.00000017512538128 ***
factor(gvkey)141384   < 0.0000000000000002 ***
factor(gvkey)145049   < 0.0000000000000002 ***
factor(gvkey)148470                     NA
factor(gvkey)150139    0.00002080910379862 ***
factor(gvkey)156617    0.00000000000009072 ***
factor(gvkey)160211                0.00601 **
factor(gvkey)164416                     NA
factor(gvkey)164664                     NA
factor(gvkey)165746                     NA
factor(gvkey)170841                     NA
factor(gvkey)174317                     NA
factor(gvkey)177376                     NA
factor(gvkey)178493                     NA
factor(gvkey)179666                     NA
factor(gvkey)180402                     NA
factor(gvkey)180646                     NA
factor(gvkey)184500                     NA
factor(sic)1531                         NA
```

factor(sic)1623	NA
factor(sic)2080	NA
factor(sic)2300	NA
factor(sic)2320	NA
factor(sic)2510	NA
factor(sic)2621	NA
factor(sic)2810	NA
factor(sic)2820	NA
factor(sic)2836	NA
factor(sic)2844	NA
factor(sic)2851	NA
factor(sic)2911	NA
factor(sic)3490	NA
factor(sic)3561	NA
factor(sic)3576	NA
factor(sic)3674	NA
factor(sic)3711	NA
factor(sic)3724	NA
factor(sic)3812	NA
factor(sic)3823	NA
factor(sic)3826	NA
factor(sic)3843	NA
factor(sic)3845	NA
factor(sic)3990	NA
factor(sic)4213	NA
factor(sic)4833	NA
factor(sic)4841	NA
factor(sic)4911	NA
factor(sic)4931	NA
factor(sic)5047	NA
factor(sic)5065	NA
factor(sic)5140	NA
factor(sic)5150	NA
factor(sic)5211	NA
factor(sic)5661	NA
factor(sic)5734	NA
factor(sic)5812	NA
factor(sic)5912	NA
factor(sic)5940	NA
factor(sic)5944	NA
factor(sic)5961	NA
factor(sic)6141	NA
factor(sic)6200	NA
factor(sic)6211	NA
factor(sic)6282	NA
factor(sic)6311	NA
factor(sic)6324	NA
factor(sic)6331	NA
factor(sic)6798	NA
factor(sic)7310	NA
factor(sic)7323	NA
factor(sic)7370	NA
factor(sic)7372	NA
factor(sic)7389	NA

```
factor(sic)7900                        NA
factor(sic)7990                        NA
factor(sic)8700                        NA
factor(sic)8721                        NA
factor(sic)8731                        NA
factor(sic)8742                        NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.00152 on 11972 degrees of freedom
Multiple R-squared:  0.178, Adjusted R-squared:  0.17
F-statistic: 21.5 on 121 and 11972 DF,  p-value: <0.0000000000000002
```

7.2 In Sample Evaluation

We then predict the growth rate and the number of followers using the fitted linear regression models (`combined_Mod1` to `combined_Mod5`). It then computes and prints the root mean squared error (RMSE) between the actual and predicted followers for each model, which can be used to evaluate the in-sample predictive accuracy of the models.

```
#predict growth rate using Training data
Train$Pre_rate_16 <- predict(Model16, Train)
Train$Pre_rate_17 <- predict(Model17, Train)
Train$Pre_rate_18 <- predict(Model18, Train)
Train$Pre_rate_19 <- predict(Model19, Train)
```

```
Warning in predict.lm(Model19, Train): prediction from rank-deficient fit;
attr(*, "non-estim") has doubtful cases
```

```
Train$Pre_rate_20 <- predict(Model20, Train)
```

```
Warning in predict.lm(Model20, Train): prediction from rank-deficient fit;
attr(*, "non-estim") has doubtful cases
```

```
#compute predicted followers using predicted growth rate
Train$Followers_M16 <- (1+Train$Pre_rate_16)*Train$followers_lag
Train$Followers_M17 <- (1+Train$Pre_rate_17)*Train$followers_lag
Train$Followers_M18 <- (1+Train$Pre_rate_18)*Train$followers_lag
Train$Followers_M19 <- (1+Train$Pre_rate_19)*Train$followers_lag
Train$Followers_M20 <- (1+Train$Pre_rate_20)*Train$followers_lag

#rmse formulae
rmse <- function(v1, v2) {
  sqrt(mean((v1 - v2)^2, na.rm = T))
}

#compute rmse to check in sample accuracy
rmse <- c(rmse(Train$followers,Train$Followers_M16),
          rmse(Train$followers,Train$Followers_M17),
          rmse(Train$followers,Train$Followers_M18),
          rmse(Train$followers,Train$Followers_M19),
          rmse(Train$followers,Train$Followers_M20))
```

```
#list down the rmse for all 5 combined models
names(rmse) <- c("Model16", "Model17", "Model18", "Model19", "Model20")
rmse
```

```
Model16 Model17 Model18 Model19 Model20
  917.2   918.7   918.9   900.4   896.0
```

7.3 Prediction of Test Followers

We obtain the last day's number of followers from the training data and merges it with the test data. Then we predict the growth rate for the test data using the `combined_Mod5` model, and fills in any missing values in the `followers` column of the test data using the predicted growth rate and the lagged values of `followers`.

```
Test16 <- predict_test(last_day_followers, Test, Model16)
Test17 <- predict_test(last_day_followers, Test, Model17)
Test18 <- predict_test(last_day_followers, Test, Model18)
Test19 <- predict_test(last_day_followers, Test, Model19)
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `Pre_rate_1 = predict(Model_no, Test_temp)`.
Caused by warning in `predict.lm()`:
! prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

```
Test20 <- predict_test(last_day_followers, Test, Model20)
```

```
Warning: There was 1 warning in `mutate()`.
ℹ In argument: `Pre_rate_1 = predict(Model_no, Test_temp)`.
Caused by warning in `predict.lm()`:
! prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases
```

We merge the original **`Test`** dataset with the derived **`Test_6`** dataset containing calculated follower counts for the specified companies.

Following the merge, the dataset is filtered to retain only observations from 1 July 2017 onwards. Subsequently, we round follower counts and create a subset of the dataset containing only the **`ID`** and **`followers`** columns.

Finally, we saved it as "submission_MXX.csv". XX denotes the submission for a particular model, for e.g. the submission for Model5 out of sample prediction will be saved as "submission_M5.csv"

```
# merge the six companies with other companies 6492
Test_submission16 <- Test16 %>% select(gvkey, date, followers, ID)
Test_submission16 <- rbind(Test_submission16, Test_6)
Test_submission16 <- Test_submission16 %>% filter(date>=20170701)
Test_submission16$followers <- round(Test_submission16$followers)
Followers_submisson_M16 <- Test_submission16 %>% select(ID, followers)

# merge the six companies with other companies 6492
Test_submission17 <- Test17 %>% select(gvkey, date, followers, ID)
```

```
Test_submission17 <- rbind(Test_submission17, Test_6)
Test_submission17 <- Test_submission17 %>% filter(date>=20170701)
Test_submission17$followers <- round(Test_submission17$followers)
Followers_submisson_M17 <- Test_submission17 %>% select(ID, followers)

# merge the six companies with other companies 6492
Test_submission18 <- Test18 %>% select(gvkey, date, followers, ID)
Test_submission18 <- rbind(Test_submission18, Test_6)
Test_submission18 <- Test_submission18 %>% filter(date>=20170701)
Test_submission18$followers <- round(Test_submission18$followers)
Followers_submisson_M18 <- Test_submission18 %>% select(ID, followers)

# merge the six companies with other companies 6492
Test_submission19 <- Test19 %>% select(gvkey, date, followers, ID)
Test_submission19 <- rbind(Test_submission19, Test_6)
Test_submission19 <- Test_submission19 %>% filter(date>=20170701)
Test_submission19$followers <- round(Test_submission19$followers)
Followers_submisson_M19 <- Test_submission19 %>% select(ID, followers)

# merge the six companies with other companies 6492
Test_submission20 <- Test20 %>% select(gvkey, date, followers, ID)
Test_submission20 <- rbind(Test_submission20, Test_6)
Test_submission20 <- Test_submission20 %>% filter(date>=20170701)
Test_submission20$followers <- round(Test_submission20$followers)
Followers_submisson_M20 <- Test_submission20 %>% select(ID, followers)

# save the sample of submission
write.csv(Followers_submisson_M16, "submission_M16.csv", row.names = FALSE)
write.csv(Followers_submisson_M17, "submission_M17.csv", row.names = FALSE)
write.csv(Followers_submisson_M18, "submission_M18.csv", row.names = FALSE)
write.csv(Followers_submisson_M19, "submission_M19.csv", row.names = FALSE)
write.csv(Followers_submisson_M20, "submission_M20.csv", row.names = FALSE)
```

7.4 Out of Sample Evaluation

Results for combined_model

submission_M20.csv Complete · now	**5003.6044**
submission_M19.csv Complete · 20s ago	**5125.92519**
submission_M18.csv Complete · 36s ago	**4483.74433**
submission_M17.csv Complete · 1m ago	**4380.31703**
submission_M16.csv Complete · 2m ago	**4464.32719**

8. LASSO

8.1 Model Selection for LASSO, XGBoost and Random Forest

The model we have selected for performing LASSO (regularisation), XGBoost and Random Forest (Ensembling) is as follows:

1. ML_Mod21 : stock price and financial ratios model
2. ML_Mod22 : stock price (lag), financial ratios (lag) model
3. ML_Mod23 : announcement and profile variables model
4. ML_Mod24 : financial ratios, stock price , announcement, profile variables combined
5. ML_Mod25 : financial ratios (lag), stock price (lag), announcement, profile variables combined

```
#-------------------mutate log(at),factor(gvkey),factor(sic)-----------------------------#

#mutate log(at) into Train dataset
Train <- Train %>% mutate (log_atq = log(atq), factor_gvkey = factor(gvkey), factor_sic = fact

#mutate log(at) into Test dataset
Test <- Test %>% mutate (log_atq = log(atq), factor_gvkey = factor(gvkey), factor_sic = factor

#------------------create formula for LASSO, XGBoost and Random Forest-----------------------

#ML_Mod21 : stock price and financial ratios model
ML_Mod21_formula <- as.formula("growth_rate ~ bm + pe_exi + ps + pcf + dpr + npm + opmbd + opm

#ML_Mod22 : stock price (lag), financial ratios (lag) model
ML_Mod22_formula <- as.formula("growth_rate~bm_lag + pe_exi_lag + ps_lag + pcf_lag + dpr_lag +

#ML_Mod23 : profile variables model
ML_Mod23_formula<- as.formula("growth_rate ~ is_annouce + red_annouce + no_annoucement + TFF +

#ML_Mod24 : financial ratios, stock price , announcement, profile variables combined
ML_Mod24_formula <- as.formula("growth_rate ~ p_daily_change + volatile + p_daily_change_lag +
ptb + rect_turn + sale_nwc + divyield + is_annouce + red_annouce + no_annoucement + TFF + cole

#ML_Mod25 : financial ratios (lag), stock price (lag), announcement, profile variables combine
ML_Mod25_formula <- as.formula("growth_rate ~ bm_lag + pe_exi_lag + ps_lag + pcf_lag + dpr_lag


# create x-matrics for each model and y-dataframe as input variables for LASSO/XGBoost
x21 <- model.matrix(ML_Mod21_formula, data = Train)[, -1] #remove y = growth rate
x22 <- model.matrix(ML_Mod22_formula, data = Train)[, -1] #remove y = growth rate
x23 <- model.matrix(ML_Mod23_formula, data = Train)[, -1] #remove y = growth rate
x24 <- model.matrix(ML_Mod24_formula, data = Train)[, -1] #remove y = growth rate
x25 <- model.matrix(ML_Mod25_formula, data = Train)[, -1] #remove y = growth rate

y21 <- model.frame(ML_Mod21_formula, data = Train)[ , "growth_rate"]
y22 <- model.frame(ML_Mod22_formula, data = Train)[ , "growth_rate"]
y23 <- model.frame(ML_Mod23_formula, data = Train)[ , "growth_rate"]
y24 <- model.frame(ML_Mod24_formula, data = Train)[ , "growth_rate"]
```

```
y25 <- model.frame(ML_Mod25_formula, data = Train)[ , "growth_rate"]

#create Twitter follower growth_rate column for Test data for xvals and yvals
Test <- Test %>%
  mutate(growth_rate = 0)

# create matrices for testing data
xvals21 <- model.matrix(ML_Mod21_formula, data = Test,  na.action = 'na.pass')[, -1]
xvals22 <- model.matrix(ML_Mod22_formula, data = Test,  na.action = 'na.pass')[, -1]
xvals23 <- model.matrix(ML_Mod23_formula, data = Test,  na.action = 'na.pass')[, -1]
xvals24 <- model.matrix(ML_Mod24_formula, data = Test,  na.action = 'na.pass')[, -1]
xvals25 <- model.matrix(ML_Mod25_formula, data = Test,  na.action = 'na.pass')[, -1]

yvals21 <- model.frame(ML_Mod21_formula, data = Test)[ , "growth_rate"]
yvals22 <- model.frame(ML_Mod22_formula, data = Test)[ , "growth_rate"]
yvals23 <- model.frame(ML_Mod23_formula, data = Test)[ , "growth_rate"]
yvals24 <- model.frame(ML_Mod24_formula, data = Test)[ , "growth_rate"]
yvals25 <- model.frame(ML_Mod25_formula, data = Test)[ , "growth_rate"]
```

8.2 LASSO Feature Selection

After selecting the 5 models, we employ the Least Absolute Shrinkage and Selection Operator (LASSO) technique to apply regularization. LASSO helps in eliminating less important features by shrinking their coefficients towards zero. This process aids in reducing the error terms within the regression model and prevents overfitting, thereby enhancing the accuracy of predictions.

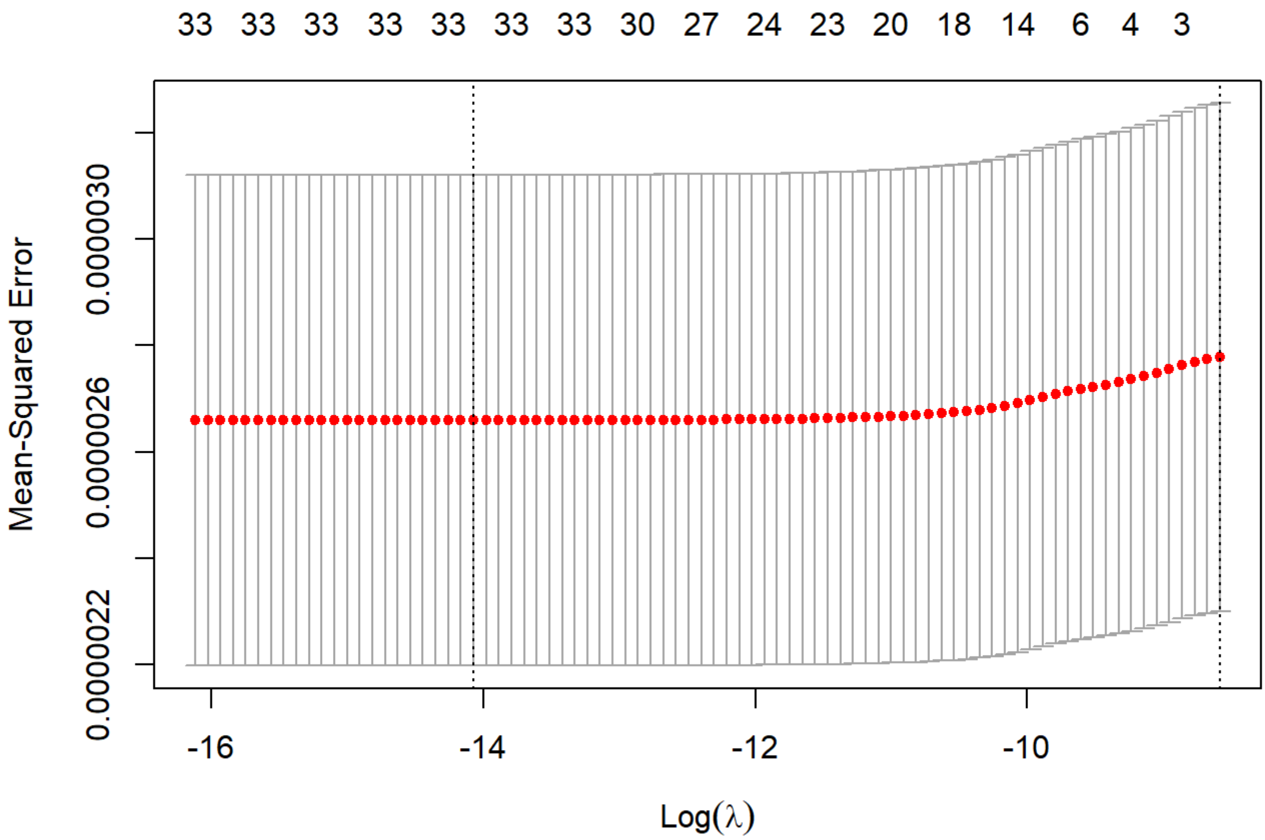
```
#--------------------- LASSO Regularisation method for feature selection-------------------#

# using LASSO for feature selection
library(glmnet)

# run cross validation LASSO (regularisation) model
set.seed(2021)
cvfit21 = cv.glmnet(x = x21, y = y21, family = "gaussian", alpha = 1, type.measure = "mse")
cvfit22 = cv.glmnet(x = x22, y = y22, family = "gaussian", alpha = 1, type.measure = "mse")
cvfit23 = cv.glmnet(x = x23, y = y23, family = "gaussian", alpha = 1, type.measure = "mse")
cvfit24 = cv.glmnet(x = x24, y = y24, family = "gaussian", alpha = 1, type.measure = "mse")
cvfit25 = cv.glmnet(x = x25, y = y25, family = "gaussian", alpha = 1, type.measure = "mse")

# lambda for best performance model(min) and most regularised model(1se)
cvfit21$lambda.min
```

```
[1] 0.0000007735
```

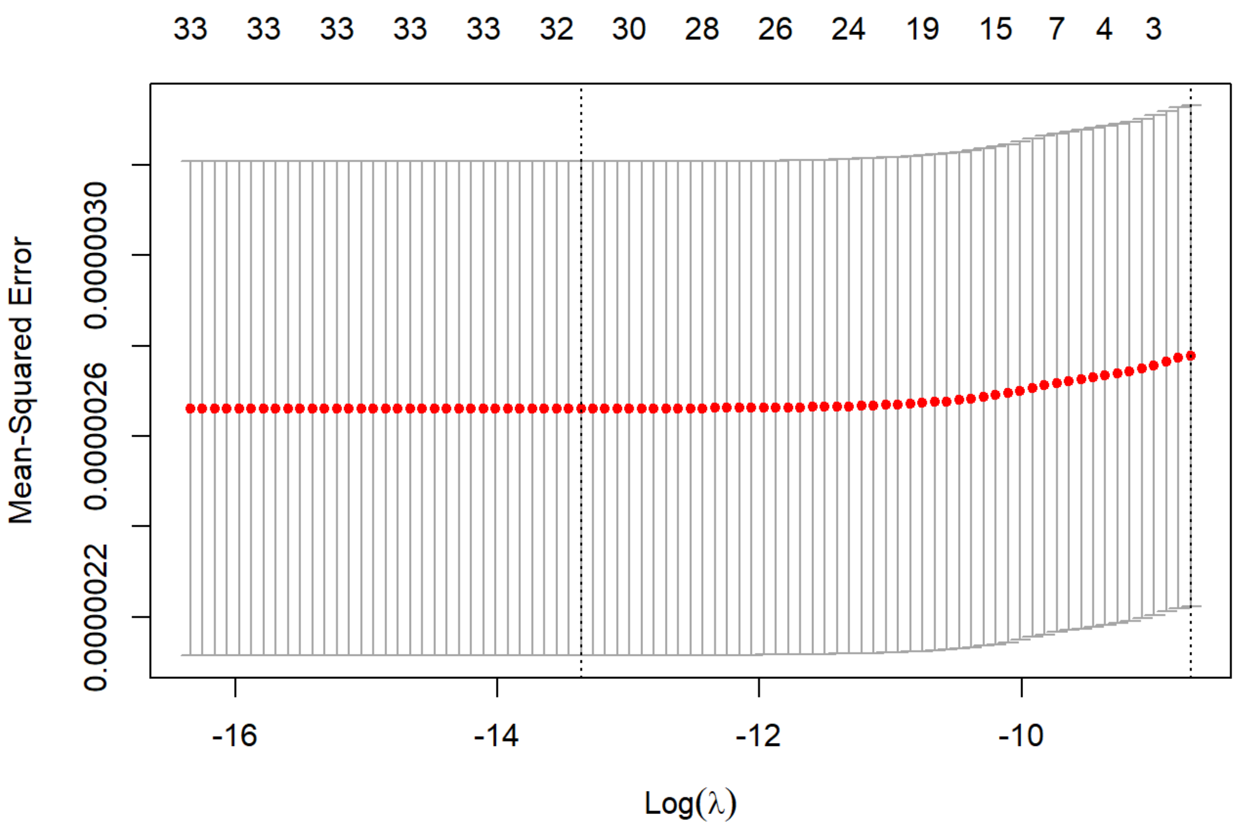
```
cvfit21$lambda.1se
```

```
[1] 0.0001872
```

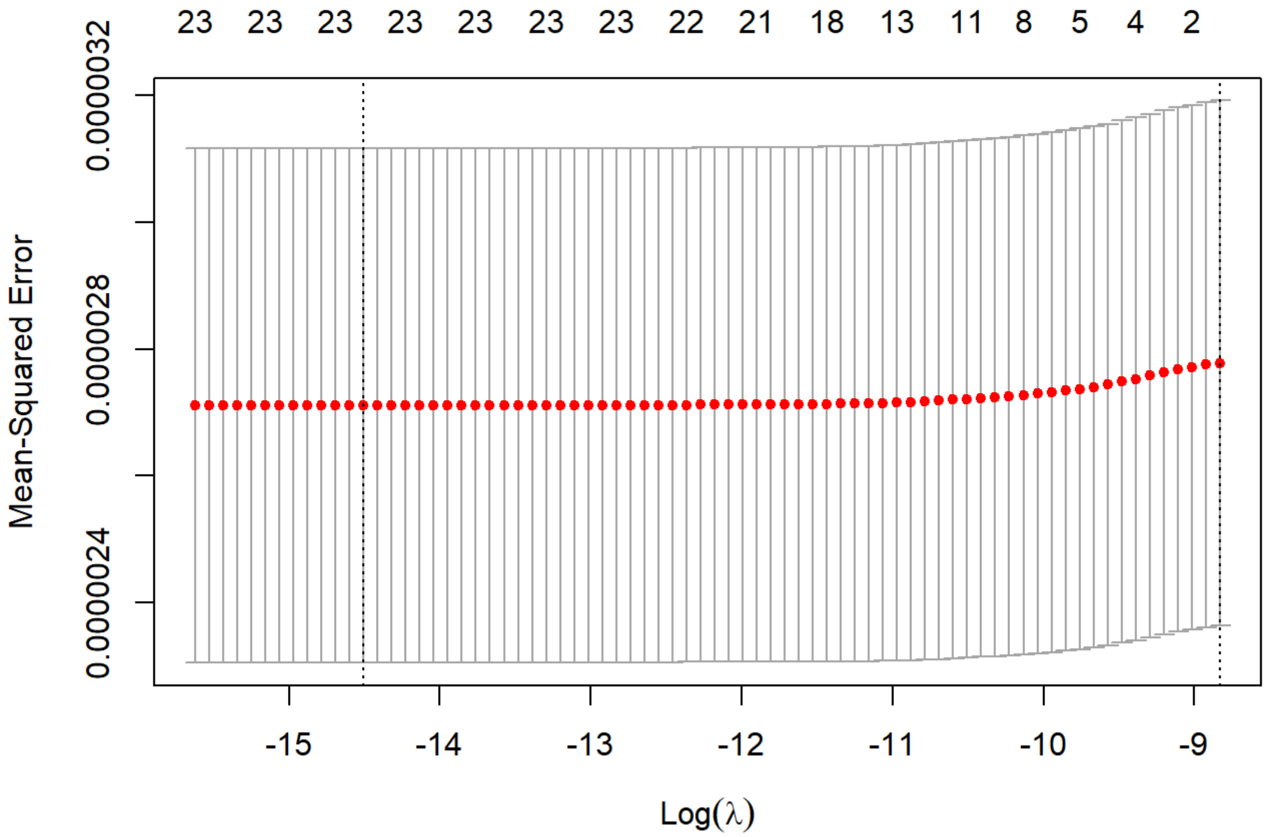
```
# plot cvfit graph
plot(cvfit21)
```



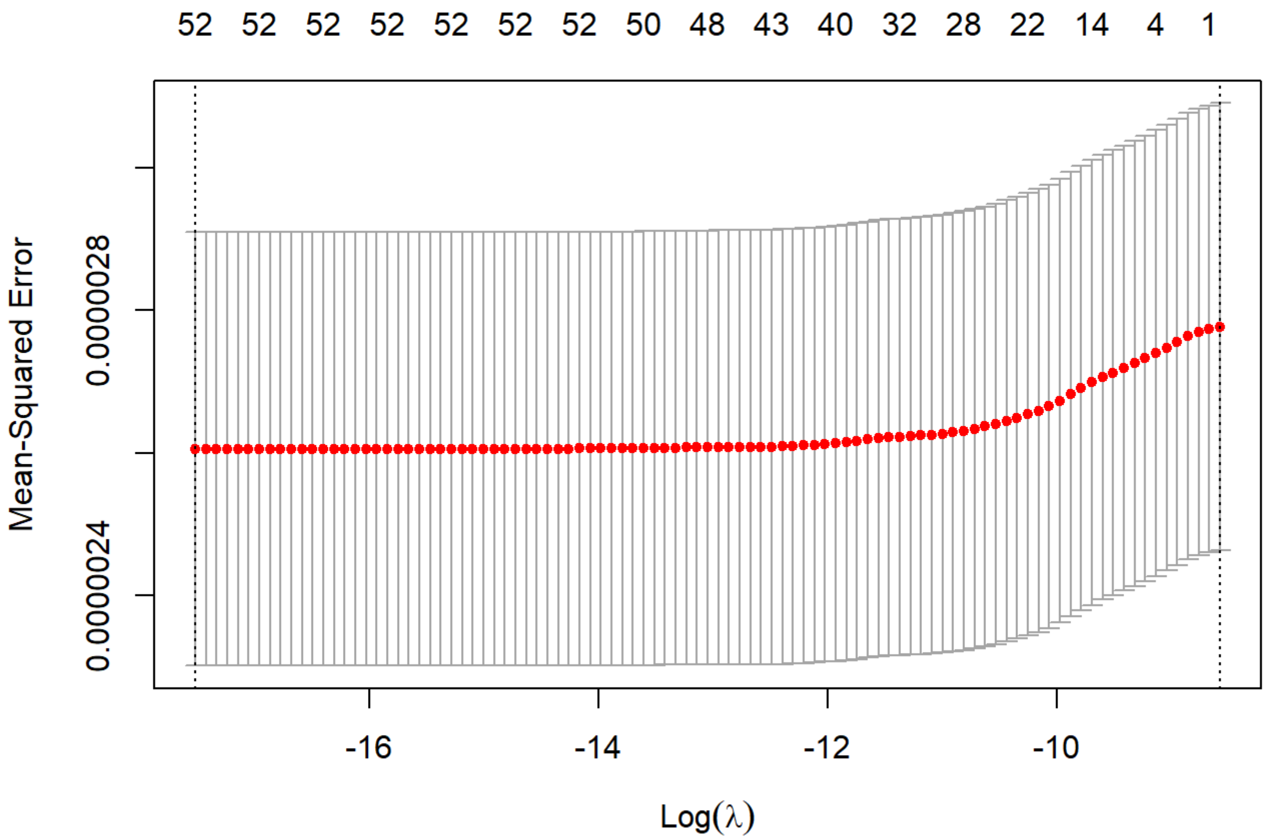

```
plot(cvfit22)
```



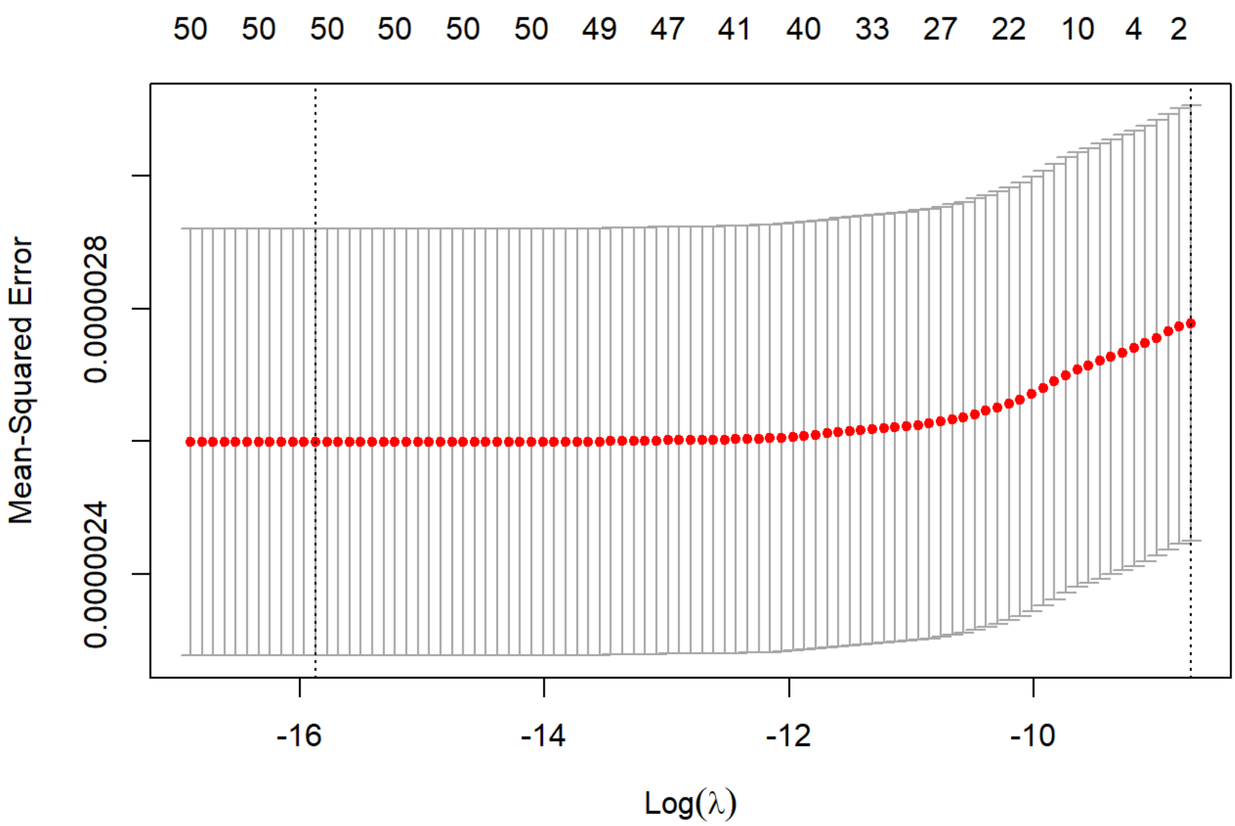
```r
plot(cvfit23)
```




```r
plot(cvfit24)
```




```
plot(cvfit25)
```

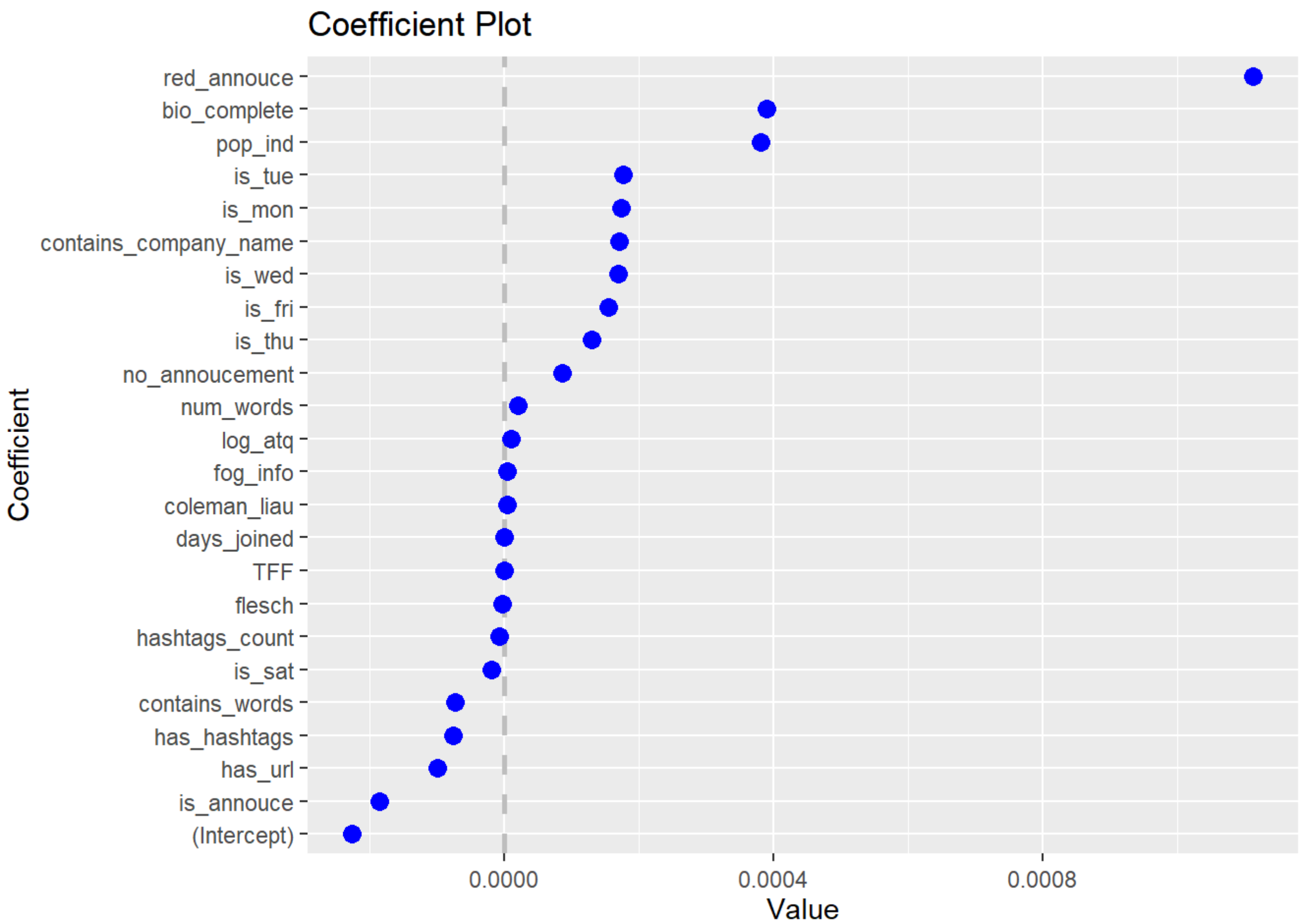


8.3 Lamda.min & Lamda.1SE

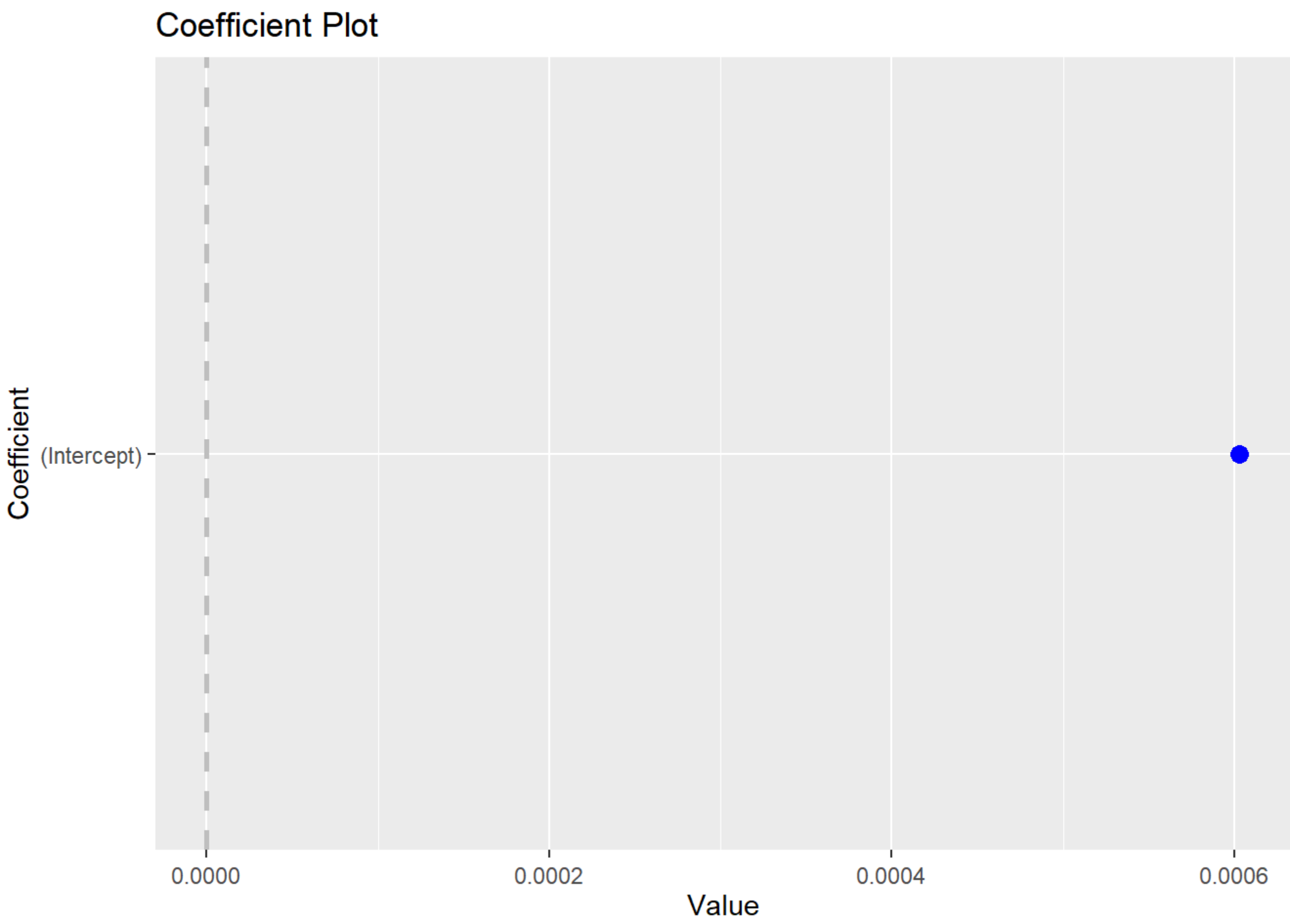
Based on the cvfit plot for all 5 models, it is interesting to note that that the MSE did not change much even with the removal of features (shrinkage of coefficients) using LASSO. This means that the incremental explanatory power of each feature is minimal. In other words, the features used in our model has explanatory power on its own and when added together as a whole. However, it has no incremental added explanatory power if we were to consider feature by feature addition.

Hence, LASSO lambda.1SE (most regularised) suggest the removal of all features for all 5 model, retaining only the y-intercept. LASSO lambda.1SE regularised technique seems to regard all features as unimportant if the features do not incrementally changes the MSE much.

```r
coefplot(cvfit23, lambda = 'lambda.min', sort = 'magnitude')
```




```r
coefplot(cvfit23, lambda = 'lambda.1se', sort = 'magnitude')
```



8.4 In Sample Evaluation

Next we performed in sample cross validation prediction using lambda.min and lambda.1SE. We then compute the respective RMSE for the 5 model which regularisation is perform on.

```
# predict using cvfit and lambda.min of the model (best performance model)
Train$LASSO_rate_M21 <- predict(cvfit21, x21, s = "lambda.min")
Train$LASSO_rate_M22 <- predict(cvfit22, x22, s = "lambda.min")
Train$LASSO_rate_M23 <- predict(cvfit23, x23, s = "lambda.min")
Train$LASSO_rate_M24 <- predict(cvfit24, x24, s = "lambda.min")
Train$LASSO_rate_M25 <- predict(cvfit25, x25, s = "lambda.min")

# predict using cvfit and lambda.1se of the model (most regularised model)
Train$LASSO_rate_SE21 <- predict(cvfit21, x21, s = "lambda.1se")
Train$LASSO_rate_SE22 <- predict(cvfit22, x22, s = "lambda.1se")
Train$LASSO_rate_SE23 <- predict(cvfit23, x23, s = "lambda.1se")
Train$LASSO_rate_SE24 <- predict(cvfit24, x24, s = "lambda.1se")
Train$LASSO_rate_SE25 <- predict(cvfit25, x25, s = "lambda.1se")

# compute predicted followers predicted growth rate obtained from lamda.min (best performance 
Train$Follower_LA_M21 <- (1+Train$LASSO_rate_M21)*Train$followers_lag
Train$Follower_LA_M22 <- (1+Train$LASSO_rate_M22)*Train$followers_lag
Train$Follower_LA_M23 <- (1+Train$LASSO_rate_M23)*Train$followers_lag
Train$Follower_LA_M24 <- (1+Train$LASSO_rate_M24)*Train$followers_lag
Train$Follower_LA_M25 <- (1+Train$LASSO_rate_M25)*Train$followers_lag

# compute predicted followers predicted growth rate obtained from lamda.1se (most regularised 
```

```
Train$Follower_LA_SE21 <- (1+Train$LASSO_rate_SE21)*Train$followers_lag
Train$Follower_LA_SE22 <- (1+Train$LASSO_rate_SE22)*Train$followers_lag
Train$Follower_LA_SE23 <- (1+Train$LASSO_rate_SE23)*Train$followers_lag
Train$Follower_LA_SE24 <- (1+Train$LASSO_rate_SE24)*Train$followers_lag
Train$Follower_LA_SE25 <- (1+Train$LASSO_rate_SE25)*Train$followers_lag

# compute rmse to measure in sample accuracy
rmse <- function(v1, v2) {
  sqrt(mean((v1 - v2)^2, na.rm = T))
}

RMSE <- c(rmse(Train$followers,Train$Follower_LA_M21),
          rmse(Train$followers,Train$Follower_LA_M22),
          rmse(Train$followers,Train$Follower_LA_M23),
          rmse(Train$followers,Train$Follower_LA_M24),
          rmse(Train$followers,Train$Follower_LA_M25),
          rmse(Train$followers,Train$Follower_LA_SE21),
          rmse(Train$followers,Train$Follower_LA_SE22),
          rmse(Train$followers,Train$Follower_LA_SE23),
          rmse(Train$followers,Train$Follower_LA_SE24),
          rmse(Train$followers,Train$Follower_LA_SE25))

# obtain RMSE for all 6 models derived from best performance (lambda.min)
names(RMSE) <- c("LASSO M21","LASSO M22","LASSO M23","LASSO M24","LASSO M25",
                 "LASSO SE21","LASSO SE22", "LASSO SE23","LASSO SE24","LASSO SE25")
RMSE
```

```
 LASSO M21  LASSO M22  LASSO M23  LASSO M24  LASSO M25 LASSO SE21 LASSO SE22 
     937.4      942.3      984.2      917.4      919.2     1016.6     1016.6 
LASSO SE23 LASSO SE24 LASSO SE25 
    1016.6     1016.6     1016.6 
```

Based on the computed RMSE, it is evident that lamda.min returns a lower in sample RMSE (i.e. LASSO M21 - M25) while lamba.1SE returns a higher in sample RMSE. However, lamda.1SE results are not meaningful since it retains only the y-intercept for prediction and we will not use it for out of sample prediction.

LASSO M24 (financial and non financial combined model) returns the prediction associated with the lowest in sample RMSE of 917.24. While LASSO M23 (non financial model) returns the prediction associated with the highest in sample RMSE of 984.2. Due to the bias and variance trade off, we will expect to see a more accurate out of sample prediction using LASSO M23 (non financial model)

```
# coefficient for best performance model(min)
coef(cvfit24, s = "lambda.min")
```

```
54 x 1 sparse Matrix of class "dgCMatrix"
                            s1
(Intercept)        0.000138336865
p_daily_change    -0.000026422051
volatile           0.002462704410
p_daily_change_lag -0.000327037609
```

```
volatile_lag            0.001735494962
bm                      0.000255222837
pe_exi                  0.000000027110
ps                     -0.000040819255
pcf                     0.000001511255
dpr                    -0.000000434253
npm                     0.002290632094
opmbd                   0.001842453743
opmad                  -0.001292735824
gpm                     0.000206947368
roa                    -0.003781025218
roe                    -0.000022170621
roce                    0.000712612951
debt_at                 0.001462015958
de_ratio               -0.000000059565
intcov                 -0.000000016335
cash_ratio             -0.000080964501
quick_ratio             0.000053116889
curr_ratio             -0.000057936231
inv_turn                0.000002861800
at_turn                 0.000197917487
debt_assets            -0.001256044370
ptb                     0.000000173423
rect_turn              -0.000003326499
sale_nwc               -0.000000150651
divyield               -0.011188860851
is_annouce             -0.000195190197
red_annouce             0.001126065799
no_annoucement          0.000074872239
TFF                    -0.000000005983
coleman_liau           -0.000003299681
flesch                  0.000009964468
fog_info                0.000028938191
num_words               0.000012299640
bio_complete            0.000537008270
has_url                 0.000033548725
hashtags_count          0.000016952352
has_hashtags           -0.000076081276
contains_company_name  .
contains_words         -0.000312255386
custom_content          0.000067014322
days_joined             0.000000066488
pop_ind                 0.000363810259
is_mon                  0.000174849656
is_tue                  0.000168050070
is_wed                  0.000161876927
is_thu                  0.000118963180
is_fri                  0.000141098865
is_sat                 -0.000013594336
log_atq                -0.000044862462
```

```
# coefficient for most regularised model(1se)
coef(cvfit24, s = "lambda.1se")
```

```
54 x 1 sparse Matrix of class "dgCMatrix"
                            s1
(Intercept)           0.000603
p_daily_change        .
volatile              .
p_daily_change_lag    .
volatile_lag          .
bm                    .
pe_exi                .
ps                    .
pcf                   .
dpr                   .
npm                   .
opmbd                 .
opmad                 .
gpm                   .
roa                   .
roe                   .
roce                  .
debt_at               .
de_ratio              .
intcov                .
cash_ratio            .
quick_ratio           .
curr_ratio            .
inv_turn              .
at_turn               .
debt_assets           .
ptb                   .
rect_turn             .
sale_nwc              .
divyield              .
is_annouce            .
red_annouce           .
no_annoucement        .
TFF                   .
coleman_liau          .
flesch                .
fog_info              .
num_words             .
bio_complete          .
has_url               .
hashtags_count        .
has_hashtags          .
contains_company_name .
contains_words        .
custom_content        .
days_joined           .
pop_ind               .
is_mon                .
is_tue                .
is_wed                .
is_thu                .
is_fri                .
```

```
is_sat                .
log_atq               .
```

8.5 Prediction of Test followers

Based on our domain knowledge, it does not makes logical sense to predict Twitter followers growth using y-intercept only. Hence, in our case, lambda.1se is ineffective in aiding us perform feature selection. We will thus rely only on lambda.min to help us perform feature selection for our out of sample prediction. However, it is important to note that because lambda.min typically have a lower in sample bias, it will result in a higher out of sample variance (bias vs variance trade off).

```
#---------------------- out of sample prediction-----------------------------------------

# predict twitter growth rate using lambda min regularised model on Test data
Test21 <- Test
Test21$LASSO_rate_M21 <- predict(cvfit21, xvals21, s = "lambda.min")
Test22 <- Test
Test22$LASSO_rate_M22 <- predict(cvfit22, xvals22, s = "lambda.min")
Test23 <- Test
Test23$LASSO_rate_M23 <- predict(cvfit23, xvals23, s = "lambda.min")
Test24 <- Test
Test24$LASSO_rate_M24 <- predict(cvfit24, xvals24, s = "lambda.min")
Test25 <- Test
Test25$LASSO_rate_M25 <- predict(cvfit25, xvals25, s = "lambda.min")

#merge Train data last day followers to Test data and create 6 Test sets for prediction
Test21 <- full_join(last_day_followers, Test21, by = join_by(gvkey, date))
Test21 <- Test21 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test22 <- full_join(last_day_followers, Test22, by = join_by(gvkey, date))
Test22 <- Test22 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test23 <- full_join(last_day_followers, Test23, by = join_by(gvkey, date))
Test23 <- Test23 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test24 <- full_join(last_day_followers, Test24, by = join_by(gvkey, date))
Test24 <- Test24 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
```

```
  rename(`ID` = ID.y)

Test25 <- full_join(last_day_followers, Test25, by = join_by(gvkey, date))
Test25 <- Test25 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)


while (any(is.na(Test21$followers))) {
  Test21 <- Test21 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test21$LASSO_rate_M21), fo
}

while (any(is.na(Test22$followers))) {
  Test22 <- Test22 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test22$LASSO_rate_M22), fo
}

while (any(is.na(Test23$followers))) {
  Test23 <- Test23 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test23$LASSO_rate_M23), fo
}

while (any(is.na(Test24$followers))) {
  Test24 <- Test24 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test24$LASSO_rate_M24), fo
}

while (any(is.na(Test25$followers))) {
  Test25 <- Test25 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test25$LASSO_rate_M25), fo
}
```

We merge the six specific companies with the rest of the dataset. The merged data is filtered to include only records from 1 July 2017 onwards. The follower counts are rounded to ensure integer values. From the resulting dataset, only the columns for ID and followers are selected, representing the necessary data for submission. This subset of data is saved as a CSV file named “submission_M1_fin.csv”, excluding row numbers.

```
pacman::p_load(tidyverse)
# merge the six companies with other companies
Test_submission21 <- Test21 %>% select(gvkey, date, followers, ID)
Test_submission21 <- rbind(Test_submission21, Test_6)
Test_submission21 <- Test_submission21 %>% filter(date>=20170701)
Test_submission21$followers <- round(Test_submission21$followers)
Followers_submisson_M21 <- Test_submission21 %>% select(ID, followers)

Test_submission22 <- Test22 %>% select(gvkey, date, followers, ID)
Test_submission22 <- rbind(Test_submission22, Test_6)
Test_submission22 <- Test_submission22 %>% filter(date>=20170701)
Test_submission22$followers <- round(Test_submission22$followers)
```

```
Followers_submisson_M22 <- Test_submission22 %>% select(ID, followers)

Test_submission23 <- Test23 %>% select(gvkey, date, followers, ID)
Test_submission23 <- rbind(Test_submission23, Test_6)
Test_submission23 <- Test_submission23 %>% filter(date>=20170701)
Test_submission23$followers <- round(Test_submission23$followers)
Followers_submisson_M23 <- Test_submission23 %>% select(ID, followers)

Test_submission24 <- Test24 %>% select(gvkey, date, followers, ID)
Test_submission24 <- rbind(Test_submission24, Test_6)
Test_submission24 <- Test_submission24 %>% filter(date>=20170701)
Test_submission24$followers <- round(Test_submission24$followers)
Followers_submisson_M24 <- Test_submission24 %>% select(ID, followers)

Test_submission25 <- Test25 %>% select(gvkey, date, followers, ID)
Test_submission25 <- rbind(Test_submission25, Test_6)
Test_submission25 <- Test_submission25 %>% filter(date>=20170701)
Test_submission25$followers <- round(Test_submission25$followers)
Followers_submisson_M25 <- Test_submission25 %>% select(ID, followers)

# save the sample of submission
write.csv(Followers_submisson_M21, "submission_LAM21.csv", row.names = FALSE)
write.csv(Followers_submisson_M22, "submission_LAM22.csv", row.names = FALSE)
write.csv(Followers_submisson_M23, "submission_LAM23.csv", row.names = FALSE)
write.csv(Followers_submisson_M24, "submission_LAM24.csv", row.names = FALSE)
write.csv(Followers_submisson_M25, "submission_LAM25.csv", row.names = FALSE)
```

8.6 Out of Sample Evaluation

results for the Lasso model using the min lambda are shown below. Feature selection by LASSO is not effective in improving our out of sample prediction due to the small incremental effect of our features on MSE. Domain knowledge model, on the other hand, uses all features and helps to retain explanatory power of each features when added as a whole. This in turn, make better out of sample prediction compared to LASSO regularised models.

submission_M25.csv Complete · now	**4356.33031**
submission_M24.csv Complete · 18s ago	**4449.15789**
submission_M23.csv Complete · 32s ago	**3108.71325**
submission_M22.csv Complete · 1m ago	**3357.82945**
submission_M21.csv Complete · 1m ago	**3312.54099**

9. XGBoost

9.1 Bayesian Optimisation Hyperparameter Tuning

XGBoost comes with a wide range of hyperparameters that control various aspects of the algorithm's behavior, such as learning rate, tree depth and regularization. Tuning these hyperparameters allows us to optimize the model's performance by finding the combination that minimizes the loss function and maximizes the evaluation metric.Instead of manually tuning the parameters using a for loop, we adopted the use of Bayesian Optimisation to search for the best parameter values.

Bayesian Optimization employs probabilistic models to efficiently evaluates different hyperparameter configurations,aiming to find the optimal set of hyperparameters in fewer iterations. This done so by first setting up a optimization function.

The optimization function takes in the tuning parameters as input and return the best cross validation results (i/e. the lowest rmse score for this case). In the following code, we use the XGBoost data format function xgb.DMatrix() to prepare the data.

We create 5 scoring function, each with the respective xgb.DMatrix created using the respective x and y of the 5 models (refer to section 8.1 of report) we have selected to enhance the prediction accuracy with boosting

```
pacman::p_load(xgboost)
pacman::p_load(ParBayesianOptimization)

#hyperparameter tuning with bayesian optimisation

#define an optimizing function
scoring_function <- function(
  eta, gamma, max_depth, min_child_weight, subsample, nfold) {

  dtrain <- xgb.DMatrix(x21, label = y21, missing = NA) #financial
  # repeat the optimising function each time using only the respective x and y of selected mod
  # dtrain <- xgb.DMatrix(x22, label = y22, missing = NA) #financial(lag)
  # dtrain <- xgb.DMatrix(x23, label = y23, missing = NA) #non financial
  # dtrain <- xgb.DMatrix(x24, label = y24, missing = NA) #combined
  # dtrain <- xgb.DMatrix(x25, label = y25, missing = NA) #combined(lag)

  pars <- list(
    eta = eta,
    gamma = gamma,
    max_depth = max_depth,
    min_child_weight = min_child_weight,
    subsample = subsample,

    booster = "gbtree", #can use gbtree or gblinear to compare which derive better results
    objective = "reg:squarederror",
    eval_metric = "rmse",
    verbosity = 0
  )

  xgbcv <- xgb.cv(
    params = pars,
    data = dtrain,
```

```
    nfold = nfold,

    nrounds = 100,
    prediction = TRUE,
    showsd = TRUE,
    early_stopping_rounds = 10,
    maximize = FALSE,
    stratified = TRUE
  )

  # required by the package, the output must be a list
  # with at least one element of "Score", the measure to optimize
  # Score must start with capital S
  # For this case, we also report the best num of iteration
  return(
    list(
      Score = min(xgbcv$evaluation_log$test_rmse_mean),
      nrounds = xgbcv$best_iteration
    )
  )
}
```

9.2 Boundary values and BayesOpt()

Then, we went on to define the boundary of values for each tuning parameter. Note that we specify integer if the parameter takes in integer values only, particularly for max_depth and nfold.

```
#start the optimization process

#define the lower and upper bounds for each function (FUN) input
bounds <- list(
  eta = c(0, 1),
  gamma =c(0, 100),
  max_depth = c(2L, 10L), # L means integers
  min_child_weight = c(1, 25),
  subsample = c(0.25, 1),
  nfold = c(3L, 10L)
)
```

The optimization process is then performed by the bayesOpt() function which will maximize the optimization function using Bayesian optimization.

```
1. FUN : scoring function for optimisation
2. bounds : boundary values for all parameters
3. initPoints : initialized the optimization process by running FUN number of times, the value
must be more than the number of FUN inputs (for this case, FUN has 6 inputs, so this value
should be at least 7)
4. iters.n : times FUN to run after initialization to search for global optimal solutions. It
is to update prior belief and form posterior belief.
```

We also use system.time() to record the time consumed by running the function.

```
#record time consumed for running the function
set.seed(2021)

time_noparallel <- system.time(
  opt_obj <- bayesOpt(
    FUN = scoring_function,
    bounds = bounds,
    initPoints = 7,
    iters.n = 5,
  ))
```

9.3 XGBoost Score Summary

We then examine the output of the optimisation process to obtain the optimised values for the 6 parameters for tuning. In this case, most optimised set of parameters corresponds with the 9th iteration which returns the lowest Score (Score = min(xgbcv$evaluation_log$
test_rmse_mean)).

1. eta (learning rate): 1
2. gamma : 0
3. max_depth : 6
4. min_child_weight : 25,
5. subsample : 1
6. nfold : 3

It is worthy to note that getBestPars function did not return the parameters that corresponds to the lowest MSE Score. This is even so after we amended FUN Scoring function to OLS regression Score function with maximise set as FALSE. We have tried using the getBestPars output set of parameters to do out of sample prediction and it did not perform as well as the set of parameters outline in 1- 6 above.

Hence, we will manually input the 1-6 parameters above as the optimised parameter for training the XGBoost models. The trained models will then create predictions with the lowest RMSE within the boundaries we have set for our parameters

```
#examine the output summary
#again, have to use capital letters required by the package
opt_obj$scoreSummary
```

```
   Epoch Iteration    eta  gamma max_depth min_child_weight subsample nfold
1:     0         1 0.3877   5.83         3           23.637    0.6108     9
2:     0         2 0.2212  33.77         9           18.424    0.9222     6
3:     0         3 0.1353  28.44         9            6.793    0.4921     8
4:     0         4 0.7217  49.76         6           17.584    0.3109     3
5:     0         5 0.9144  89.25         5           12.051    0.3588     5
6:     0         6 0.7045  58.30         3            3.197    0.8748     6
7:     0         7 0.4494  77.72         7           10.302    0.7575     7
8:     1         8 1.0000 100.00         6           25.000    0.7385     3
9:     2         9 1.0000   0.00         6           25.000    1.0000     3
```

```
10:      3        10 0.6255 100.00        10           25.000    1.0000     3
11:      4        11 0.2083  65.57         2           25.000    0.2500     3
12:      5        12 0.0000 100.00        10           25.000    1.0000    10
    gpUtility acqOptimum inBounds Elapsed    Score nrounds errorMessage
 1:        NA      FALSE     TRUE    2.10 0.001630      24           NA
 2:        NA      FALSE     TRUE    6.92 0.001646      55           NA
 3:        NA      FALSE     TRUE   13.19 0.001630      91           NA
 4:        NA      FALSE     TRUE    0.56 0.001651      13           NA
 5:        NA      FALSE     TRUE    1.08 0.001647      13           NA
 6:        NA      FALSE     TRUE    1.00 0.001637      14           NA
 7:        NA      FALSE     TRUE    5.69 0.001656      41           NA
 8:    0.5754       TRUE     TRUE    1.07 0.001666      19           NA
 9:    0.4466       TRUE     TRUE    0.48 0.001545       6           NA
10:    0.5009       TRUE     TRUE    1.50 0.001659      17           NA
11:    0.4710       TRUE     TRUE    0.62 0.001647      52           NA
12:    0.3822       TRUE     TRUE    0.42 0.499400       1           NA
```

```
#obtain the optimised values for the 6 parameters we want to tune
#the built-in function output the FUN input arguments only.
getBestPars(opt_obj)
```

```
$eta
[1] 0

$gamma
[1] 100

$max_depth
[1] 10

$min_child_weight
[1] 25

$subsample
[1] 1

$nfold
[1] 10
```

```
#take the optimized values for the six parameters and train the XGBoost model
#take the optimal parameters for xgboost()
params <- list(eta = 1,
               gamma = 0,
               max_depth = 6,
               min_child_weight = 25,
               subsample = 1,
               nfold = 3,
               objective = "reg:squarederror")

#the numrounds which gives the min RMSE score
numrounds <- opt_obj$scoreSummary$nrounds[
  which(opt_obj$scoreSummary$Score
        == min(opt_obj$scoreSummary$Score))]
```

```r
fit_tuned26 <- xgboost(params = params,
                       data = x21,
                       label = y21,
                       nrounds = numrounds,
                       eval_metric = "rmse")
```

```
[1] train-rmse:0.001667
[2] train-rmse:0.001484
[3] train-rmse:0.001430
[4] train-rmse:0.001396
[5] train-rmse:0.001387
[6] train-rmse:0.001374
```

```r
fit_tuned27 <- xgboost(params = params,
                       data = x22,
                       label = y22,
                       nrounds = numrounds,
                       eval_metric = "rmse")
```

```
[1] train-rmse:0.001667
[2] train-rmse:0.001457
[3] train-rmse:0.001425
[4] train-rmse:0.001394
[5] train-rmse:0.001367
[6] train-rmse:0.001357
```

```r
fit_tuned28 <- xgboost(params = params,
                       data = x23,
                       label = y23,
                       nrounds = numrounds,
                       eval_metric = "rmse")
```

```
[1] train-rmse:0.001667
[2] train-rmse:0.001474
[3] train-rmse:0.001427
[4] train-rmse:0.001416
[5] train-rmse:0.001395
[6] train-rmse:0.001382
```

```r
fit_tuned29 <- xgboost(params = params,
                       data = x24,
                       label = y24,
                       nrounds = numrounds,
                       eval_metric = "rmse")
```

```
[1] train-rmse:0.001667
[2] train-rmse:0.001463
[3] train-rmse:0.001418
[4] train-rmse:0.001390
[5] train-rmse:0.001373
[6] train-rmse:0.001359
```

```
fit_tuned30 <- xgboost(params = params,
                       data = x25,
                       label = y25,
                       nrounds = numrounds,
                       eval_metric = "rmse")
```

```
[1] train-rmse:0.001667
[2] train-rmse:0.001465
[3] train-rmse:0.001443
[4] train-rmse:0.001417
[5] train-rmse:0.001387
[6] train-rmse:0.001376
```

9.4 In Sample Accuracy

We run the optimised parameters obtained from Bayesian Optimisation technique to train 5 models (see section 8.1 for models selected) using XGBoost. The predicted growth rate from Train data is then used to compute the actual no of Twitter followers in the respective dates.

Based on the RMSE results computed, XG28 (Non financial model on XGBoost) performs the best in terms of in sample prediction with an in sample RMSE of 640.5. On the other hand, the model that performs the worse for in sample prediction is XG27 (Financial(lag), StockPrice(lag) model on XGBoost).

```
# compute in sample predicted growth rate
Train$XG_rate_26 <- predict(fit_tuned26, x21)
Train$XG_rate_27 <- predict(fit_tuned27, x22)
Train$XG_rate_28 <- predict(fit_tuned28, x23)
Train$XG_rate_29 <- predict(fit_tuned29, x24)
Train$XG_rate_30 <- predict(fit_tuned30, x25)

# using in sample predicted growth rate, compute in sample predicted followers.
Train$Followers_XG_26<- (1+Train$XG_rate_26)*Train$followers_lag
Train$Followers_XG_27<- (1+Train$XG_rate_27)*Train$followers_lag
Train$Followers_XG_28<- (1+Train$XG_rate_28)*Train$followers_lag
Train$Followers_XG_29<- (1+Train$XG_rate_29)*Train$followers_lag
Train$Followers_XG_30<- (1+Train$XG_rate_30)*Train$followers_lag

# compute rmse to evaluate in sample prediction accuracy
rmse <- function(v1, v2) {
  sqrt(mean((v1 - v2)^2, na.rm = T))
}

RMSE <- c(rmse(Train$followers,Train$Followers_XG_26),
          rmse(Train$followers,Train$Followers_XG_27),
          rmse(Train$followers,Train$Followers_XG_28),
          rmse(Train$followers,Train$Followers_XG_29),
          rmse(Train$followers,Train$Followers_XG_30))

## To get the RMSE for all the models
names(RMSE) <- c("XG26", "XG27", "XG28", "XG29", "XG30")
RMSE
```

```
 XG26  XG27  XG28  XG29  XG30
694.7 707.9 640.5 677.0 706.4
```

9.5 Prediction of Test Followers

```
#evaluation of the tuned model
#prediction on test sample

#merge the last day of followers in Train data to Test data
Test26 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test26 <- Test26 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test27 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test27 <- Test27 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test28 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test28 <- Test28 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test29 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test29 <- Test29 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test30 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test30 <- Test30 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

pred.xgb.tuned26 <- predict(fit_tuned26, xvals21)
pred.xgb.tuned27 <- predict(fit_tuned27, xvals22)
pred.xgb.tuned28 <- predict(fit_tuned28, xvals23)
pred.xgb.tuned29 <- predict(fit_tuned29, xvals24)
pred.xgb.tuned30 <- predict(fit_tuned30, xvals25)
print(pred.xgb.tuned27)
```

```
  [1]  0.00058358  0.00058358  0.00053105  0.00053105  0.00046934  0.00053105
  [7]  0.00053105  0.00053105  0.00058358  0.00046934  0.00053105  0.00115702
```

```
 [13]  0.00053105  0.00053105  0.00053105  0.00053105  0.00053105  0.00115702
 [19]  0.00053105  0.00058358  0.00058358  0.00046934  0.00053105  0.00115702
 [25]  0.00058358  0.00053105  0.00058358  0.00053105  0.00053105  0.00115702
 [31]  0.00053105  0.00058358  0.00053105  0.00058358  0.00046934  0.00053105
 [37]  0.00115702  0.00053105  0.00053105  0.00058358  0.00046934  0.00115702
 [43]  0.00058358  0.00053105  0.00053105  0.00053105  0.00046934  0.00115702
 [49]  0.00058358  0.00053105  0.00053105  0.00053105  0.00046934  0.00053105
 [55]  0.00115702  0.00053105  0.00053105  0.00053105  0.00053105  0.00120956
 [61]  0.00053105  0.00058358  0.00053105  0.00046934  0.00053105  0.00053105
 [67]  0.00053105  0.00053105  0.00046934  0.00053105  0.00115702  0.00058358
 [73]  0.00058358  0.00053105  0.00053105  0.00046934  0.00138854  0.00173656
 [79]  0.00194721  0.00173656  0.00173656  0.00173656  0.00173656  0.00138854
 [85]  0.00173656  0.00194721  0.00173656  0.00173656  0.00173656  0.00173656
 [91]  0.00138854  0.00173656  0.00194721  0.00173656  0.00173656  0.00173656
 [97]  0.00173656  0.00138854  0.00173656  0.00194721  0.00173656  0.00173656
[103]  0.00173656  0.00173656  0.00138854  0.00173656  0.00194721  0.00173656
[109]  0.00173656  0.00173656  0.00173656  0.00138854  0.00173656  0.00194721
[115]  0.00173656  0.00173656  0.00173656  0.00173656  0.00138854  0.00173656
[121]  0.00194721  0.00173656  0.00173656  0.00173656  0.00173656  0.00138854
[127]  0.00173656  0.00194721  0.00173656  0.00173656  0.00173656  0.00173656
[133]  0.00138854  0.00173656  0.00194721  0.00173656  0.00173656  0.00173656
[139]  0.00173656  0.00138854  0.00173656  0.00194721  0.00173656  0.00173656
[145]  0.00173656  0.00173656  0.00138854  0.00173656  0.00194721  0.00173656
[151]  0.00173656  0.00173656  0.00173656  0.00138854  0.00173656  0.00194721
[157]  0.00173656  0.00173656  0.00173656  0.00173656  0.00138854  0.00173656
[163]  0.00194721  0.00173656  0.00173656  0.00173656  0.00173656  0.00138854
[169] -0.00091171  0.00031172  0.00019747  0.00019747 -0.00002744 -0.00002744
[175] -0.00091171 -0.00018655 -0.00002744 -0.00002744 -0.00002744  0.00019747
[181] -0.00091171  0.00019747  0.00003836 -0.00002744 -0.00002744 -0.00002744
[187]  0.00003836 -0.00002744 -0.00002744 -0.00002744 -0.00002744 -0.00091171
[193] -0.00002744 -0.00018655 -0.00002744  0.00008667 -0.00002744 -0.00018655
[199] -0.00002744 -0.00002744 -0.00002744 -0.00002744 -0.00091171 -0.00018655
[205] -0.00002744 -0.00002744 -0.00002744 -0.00002744 -0.00091171 -0.00018655
[211] -0.00002744 -0.00002744 -0.00002744 -0.00002744 -0.00018655 -0.00002744
[217] -0.00002744 -0.00002744 -0.00002744 -0.00091171 -0.00018655 -0.00002744
[223]  0.00008667 -0.00002744 -0.00018565  0.00042251  0.00019747 -0.00002744
[229] -0.00002744 -0.00002744 -0.00091171 -0.00002744 -0.00018655 -0.00002744
[235] -0.00002744  0.00019747 -0.00018655 -0.00002744 -0.00002744  0.00019747
[241] -0.00002744 -0.00091171  0.00098137  0.00077072  0.00077072  0.00077072
[247]  0.00098137  0.00077072  0.00070792  0.00077072  0.00077072  0.00098137
[253]  0.00077072  0.00055093  0.00077072  0.00077072  0.00091858  0.00077072
[259]  0.00077072  0.00077072  0.00077072  0.00098137  0.00077072  0.00077072
[265]  0.00077072  0.00077072  0.00098137  0.00077072  0.00077072  0.00077072
[271]  0.00077072  0.00012515  0.00098137  0.00077072  0.00077072  0.00077072
[277]  0.00070792  0.00098137  0.00070792  0.00077072  0.00077072  0.00077072
[283]  0.00098137  0.00077072  0.00077072  0.00077072  0.00077072  0.00012515
[289]  0.00077072  0.00090970  0.00077072  0.00077072  0.00091858  0.00077072
[295]  0.00077072  0.00077072  0.00077072  0.00098137  0.00077072  0.00077072
[301]  0.00077072  0.00077072  0.00098137  0.00077072  0.00077072  0.00077072
[307]  0.00077072  0.00051020  0.00057191  0.00057191 -0.00033317 -0.00033317
[313]  0.00057191  0.00057191  0.00051020  0.00057191  0.00057191  0.00057191
[319]  0.00057191  0.00057191  0.00057191  0.00051020  0.00057191  0.00057191
[325]  0.00057191  0.00057191  0.00057191  0.00057191  0.00051020  0.00057191
[331]  0.00057191  0.00057191  0.00057191  0.00057191  0.00162739  0.00051020
```

```
[337]  0.00057191  0.00090098  0.00016111  0.00016111  0.00016111  0.00016111
[343]  0.00009940  0.00016111  0.00090098  0.00016111  0.00016111  0.00016111
[349]  0.00016111  0.00009940  0.00016111  0.00090098  0.00016111  0.00016111
[355]  0.00016111  0.00009831  0.00009940  0.00016111  0.00090098  0.00016111
[361]  0.00016111  0.00286411  0.00016111  0.00042067  0.00048238  0.00122225
[367] -0.00109787  0.00016111  0.00016111  0.00016111  0.00009940  0.00016111
[373]  0.00090098  0.00016111  0.00016111  0.00016111  0.00046411  0.00009940
[379]  0.00016111  0.00090098  0.00048238  0.00016111  0.00046411  0.00016111
[385]  0.00009940  0.00016111  0.00090098  0.00016111  0.00056281  0.00016111
[391]  0.00016111  0.00050109  0.00056281  0.00056281  0.00016111  0.00016111
[397]  0.00016111  0.00016111  0.00009940  0.00097795  0.00133111  0.00072984
[403]  0.00132596  0.00132596  0.00095990  0.00132596  0.00072984  0.00072984
[409]  0.00072984  0.00038183  0.00095475  0.00095990  0.00072984  0.00192750
[415]  0.00072984  0.00132596  0.00193264  0.00132596  0.00072984  0.00132596
[421]  0.00132596  0.00193264  0.00072984  0.00072984  0.00132596  0.00132596
[427]  0.00097795  0.00133111  0.00132596  0.00072984  0.00192750  0.00204160
[433]  0.00095475  0.00095990  0.00072984  0.00132596  0.00072984  0.00192750
[439]  0.00132596  0.00133111  0.00072984  0.00072984  0.00095475  0.00072984
[445]  0.00038183  0.00073499  0.00132596  0.00072984  0.00072984  0.00097795
[451]  0.00132596  0.00133111  0.00132596  0.00192750  0.00132596  0.00132596
[457]  0.00073499  0.00072984  0.00072984  0.00072984  0.00072984  0.00073499
[463]  0.00132596  0.00132596  0.00132596  0.00132596  0.00038183  0.00072984
[469]  0.00073499  0.00095475  0.00072984  0.00132596  0.00132596  0.00033903
[475]  0.00025113  0.00033903  0.00033903  0.00033903  0.00033903  0.00033903
[481]  0.00033903  0.00047801  0.00127075  0.00096013  0.00087224  0.00033903
[487]  0.00033903  0.00033903  0.00033903  0.00027732  0.00025113  0.00033903
[493]  0.00096013  0.00033903  0.00033903 -0.00017515  0.00118285  0.00033903
[499]  0.00127075  0.00127075  0.00118285  0.00096013  0.00033903  0.00127075
[505]  0.00089842  0.00096013  0.00087224  0.00033903  0.00033903  0.00033903
[511]  0.00127075  0.00027732  0.00047801  0.00096013  0.00033903  0.00127075
[517]  0.00033903  0.00033903  0.00096013  0.00011924  0.00033903  0.00033903
[523]  0.00033903  0.00127075  0.00033903  0.00033903  0.00143183  0.00113137
[529]  0.00113137  0.00113137  0.00143183  0.00113137  0.00113137  0.00113137
[535]  0.00113137  0.00113137  0.00113137  0.00113137  0.00175248  0.00205294
[541]  0.00113137  0.00113137  0.00175248  0.00113137  0.00192830  0.00113137
[547]  0.00113137  0.00113137  0.00113137  0.00192830  0.00113137  0.00113137
[553]  0.00113137  0.00192830  0.00113137  0.00113137  0.00113137  0.00113137
[559]  0.00192830  0.00113137  0.00113137  0.00113137  0.00113137  0.00113137
[565]  0.00113137  0.00142754  0.00142754  0.00142754  0.00142754  0.00142754
[571]  0.00142754  0.00192830  0.00142754  0.00142754  0.00142754  0.00142754
[577]  0.00192830  0.00142754  0.00142754  0.00142754  0.00142754  0.00118678
[583]  0.00142754  0.00192830  0.00142754  0.00142754  0.00142754  0.00142754
[589]  0.00025933  0.00025933  0.00025933  0.00053703  0.00053703  0.00022812
[595]  0.00025933  0.00025933  0.00053703  0.00053703  0.00025933  0.00025933
[601]  0.00053703  0.00025933  0.00053703  0.00053703  0.00025933  0.00022812
[607]  0.00025933  0.00025933  0.00025613  0.00025613  0.00057741  0.00050583
[613]  0.00053703  0.00053703  0.00025933  0.00053703  0.00025933  0.00014768
[619]  0.00053703  0.00053703  0.00053703  0.00025933  0.00053703  0.00053703
[625]  0.00053703 -0.00059129  0.00053703  0.00019334  0.00050583  0.00053703
[631]  0.00053703  0.00025613  0.00025933  0.00025933  0.00025613  0.00050583
[637]  0.00053703  0.00053703  0.00053703  0.00025933  0.00025613  0.00053703
[643]  0.00022812  0.00025933  0.00025933  0.00025933  0.00053703 -0.00065409
[649]  0.00014768  0.00053703  0.00053703  0.00025933  0.00025933  0.00053703
[655]  0.00025613  0.00022812  0.00025933  0.00025933  0.00025933  0.00053703
```

```
[661]  0.00053703  0.00053703  0.00022812  0.00025933  0.00053703  0.00053703
[667]  0.00025933  0.00025933  0.00010735 -0.00073602  0.00000385  0.00016906
[673]  0.00016906  0.00057076  0.00016906  0.00010735  0.00016906  0.00090893
[679]  0.00016906  0.00057076  0.00016906  0.00016906 -0.00077452 -0.00071280
[685] -0.00071280  0.00287206  0.00057076  0.00016906  0.00057076  0.00010735
[691]  0.00016906  0.00090893  0.00057076  0.00057076  0.00057076  0.00281876
[697]  0.00044824  0.00057076  0.00057076  0.00016906  0.00016906  0.00016906
[703]  0.00057076  0.00044824  0.00057076  0.00057076  0.00057076  0.00016906
[709]  0.00252204  0.00016906  0.00010735  0.00016906  0.00090893  0.00016906
[715]  0.00016906  0.00016906  0.00281876  0.00010735  0.00016906  0.00090893
[721]  0.00057076  0.00016906  0.00057076  0.00057076  0.00050905  0.00057076
[727]  0.00057076 -0.00071280  0.00057076  0.00057076  0.00057076  0.00050905
[733]  0.00057076  0.00057076  0.00057076  0.00016906  0.00016906  0.00057076
[739]  0.00016906  0.00090893  0.00016906  0.00057076  0.00016906  0.00016906
[745]  0.00050905  0.00057076  0.00057076  0.00057076  0.00057076  0.00016906
[751]  0.00016906  0.00010735  0.00016906  0.00090893  0.00016906  0.00057076
[757] -0.00071280  0.00016906 -0.00004682  0.00169083  0.00135086  0.00167757
[763]  0.00062534  0.00062534  0.00135086  0.00062534  0.00169083  0.00135086
[769]  0.00167757  0.00062534  0.00017135  0.00062534  0.00062534  0.00096531
[775]  0.00062534  0.00095205  0.00062534  0.00062534  0.00135086  0.00135086
[781]  0.00169083  0.00135086  0.00167757  0.00062534  0.00135086  0.00062534
[787]  0.00062534  0.00096531  0.00062534  0.00095205  0.00062534  0.00062534
[793]  0.00135086  0.00062534  0.00051132  0.00039626  0.00072297  0.00062534
[799]  0.00062534  0.00167214  0.00174720  0.00169083  0.00135086  0.00167757
[805]  0.00062534  0.00174720  0.00135086  0.00062534  0.00169083  0.00135086
[811]  0.00167757  0.00062534  0.00135086  0.00135086  0.00062534  0.00096531
[817]  0.00062534  0.00095205  0.00039626  0.00062534  0.00135086  0.00135086
[823]  0.00096531  0.00062534  0.00095205  0.00062534  0.00062534  0.00062534
[829]  0.00062534  0.00169083  0.00135086  0.00167757  0.00039626  0.00135086
[835]  0.00017135  0.00135086  0.00169083  0.00135086  0.00167757  0.00062534
[841]  0.00062534  0.00135086 -0.00021817  0.00096531  0.00062534  0.00095205
[847]  0.00062534  0.00039626  0.00039626  0.00062534  0.00093635 -0.00003911
[853]  0.00035024  0.00056090  0.00035024  0.00035024  0.00035024  0.00035024
[859] -0.00003911  0.00056090  0.00035024  0.00035024  0.00035024  0.00035024
[865] -0.00003911  0.00035024  0.00056090  0.00035024  0.00035024  0.00035024
[871]  0.00035024 -0.00003911  0.00035024  0.00056090  0.00035024  0.00035024
[877]  0.00035024  0.00035024  0.00056090  0.00035024  0.00035024  0.00035024
[883]  0.00035024  0.00035024  0.00056090  0.00035024  0.00035024  0.00035024
[889]  0.00035024  0.00056090  0.00035024  0.00035024  0.00035024  0.00035024
[895] -0.00003911  0.00035024  0.00056090  0.00035024  0.00035024  0.00035024
[901]  0.00035024 -0.00003911  0.00035024  0.00056090  0.00035024  0.00035024
[907]  0.00035024  0.00035024 -0.00003911  0.00035024  0.00056090  0.00035024
[913]  0.00035024  0.00035024  0.00035024 -0.00003911  0.00056090  0.00035024
[919]  0.00035024  0.00035024  0.00035024 -0.00003911  0.00056090  0.00035024
[925]  0.00035024  0.00035024  0.00035024  0.00056090  0.00035024  0.00035024
[931]  0.00035024  0.00047383  0.00047383  0.00047383  0.00047383  0.00068449
[937]  0.00047383  0.00047383  0.00047383  0.00047383  0.00047383  0.00047383
[943]  0.00047383  0.00047383  0.00068449  0.00047383  0.00047383  0.00047383
[949]  0.00047383  0.00044263  0.00068449  0.00052637  0.00047383  0.00047383
[955]  0.00047383  0.00068449  0.00047383  0.00047383  0.00047383  0.00047383
[961]  0.00068449  0.00047383  0.00047383  0.00047383  0.00047383  0.00047383
[967]  0.00047383  0.00047383  0.00047383  0.00035131  0.00068449  0.00047383
[973]  0.00047383  0.00047383  0.00047383  0.00052637  0.00047383  0.00047383
[979]  0.00068449  0.00047383  0.00047383  0.00047383  0.00047383  0.00068449
```

```
 [985]  0.00047383  0.00047383  0.00047383  0.00068449  0.00047383  0.00047383
 [991]  0.00047383  0.00041031  0.00019720  0.00019720  0.00019720  0.00041031
 [997]  0.00019720 -0.00008369 -0.00008369  0.00019720  0.00041031  0.00019720
[1003]  0.00014121  0.00019720  0.00019720  0.00041031  0.00019720  0.00019720
[1009]  0.00019720  0.00019720  0.00041031  0.00042211  0.00019720  0.00023886
[1015]  0.00019720  0.00041031  0.00019720  0.00019720  0.00019720 -0.00008369
[1021]  0.00041031  0.00019720  0.00019720  0.00019720  0.00019720  0.00012941
[1027]  0.00019720  0.00019720  0.00039707  0.00039707  0.00041031  0.00019720
[1033]  0.00019720  0.00019720  0.00019720  0.00019720  0.00123211 -0.00008369
[1039]  0.00019720  0.00041031  0.00019720  0.00019720  0.00019720  0.00041031
[1045]  0.00019720  0.00019720  0.00019720  0.00019720  0.00041031  0.00019720
[1051]  0.00019720  0.00019720  0.00019720  0.00019707  0.00044315 -0.00022501
[1057] -0.00022501 -0.00022501 -0.00022501 -0.00022501 -0.00022501  0.00044315
[1063] -0.00022501 -0.00022501 -0.00022501 -0.00022501 -0.00022501 -0.00022501
[1069] -0.00022501 -0.00022501  0.00044315 -0.00022501 -0.00022501  0.00044315
[1075] -0.00022501 -0.00022501 -0.00022501  0.00044315 -0.00022501 -0.00022501
[1081] -0.00022501 -0.00022501  0.00044315 -0.00022501 -0.00022501 -0.00022501
[1087] -0.00022501 -0.00022501  0.00044315 -0.00022501 -0.00022501 -0.00022501
[1093]  0.00044315 -0.00022501 -0.00022501 -0.00022501  0.00044315 -0.00022501
[1099] -0.00022501 -0.00012513  0.00028446  0.00028446  0.00090556  0.00090556
[1105]  0.00042344  0.00028446 -0.00012513  0.00028446  0.00028446  0.00028446
[1111]  0.00028446  0.00028446  0.00028446 -0.00012513  0.00028446  0.00028446
[1117]  0.00028446  0.00028446  0.00028446  0.00028446 -0.00012513  0.00028446
[1123]  0.00028446  0.00006467  0.00028446  0.00028446  0.00028446 -0.00012513
[1129]  0.00028446  0.00028446  0.00028446  0.00028446  0.00028446 -0.00012513
[1135]  0.00028446  0.00028446  0.00028446  0.00028446  0.00028446  0.00028446
[1141] -0.00012513  0.00028446  0.00028446  0.00028446  0.00028446  0.00028446
[1147]  0.00028446 -0.00012513  0.00028446  0.00028446  0.00028446  0.00028446
[1153]  0.00028446 -0.00012513  0.00028446  0.00028446  0.00028446  0.00028446
[1159]  0.00028446  0.00028446  0.00049598  0.00090556  0.00090556  0.00090556
[1165]  0.00028446  0.00028446  0.00028446 -0.00012513  0.00028446  0.00028446
[1171]  0.00028446  0.00028446  0.00028446  0.00028446 -0.00012513  0.00028446
[1177]  0.00028446  0.00028446  0.00028446  0.00028446  0.00028446 -0.00012513
[1183]  0.00028446  0.00028446  0.00028446  0.00028446  0.00028446  0.00028446
[1189] -0.00012513  0.00084442  0.00084442  0.00084442  0.00112532  0.00112532
[1195]  0.00112532  0.00112532  0.00112532  0.00084442  0.00112532  0.00084442
[1201]  0.00112532  0.00112532  0.00112532  0.00132518  0.00073596  0.00112532
[1207]  0.00132518  0.00112532  0.00112532  0.00112532  0.00112532  0.00196342
[1213]  0.00164646  0.00315960  0.00196342  0.00112532  0.00112532  0.00084442
[1219]  0.00112532  0.00112532  0.00112532  0.00112532  0.00112532  0.00315960
[1225]  0.00112532  0.00112532  0.00084442  0.00112532  0.00112532  0.00112532
[1231]  0.00112532  0.00084442  0.00112532  0.00164646  0.00315960  0.00109411
[1237]  0.00112532  0.00112532  0.00112532  0.00315960  0.00112532  0.00132518
[1243]  0.00315960  0.00084442  0.00132518  0.00112532  0.00196342  0.00084442
[1249]  0.00132518  0.00112532  0.00164646  0.00112532  0.00112532  0.00112532
[1255]  0.00112532  0.00084442  0.00084442  0.00074170  0.00041778  0.00041778
[1261]  0.00041778  0.00041778  0.00074170  0.00041778  0.00041778  0.00041778
[1267]  0.00041778  0.00074170  0.00041778  0.00041778  0.00041778  0.00041778
[1273]  0.00074170  0.00041778  0.00041778  0.00041778  0.00041778  0.00002843
[1279]  0.00041778  0.00074170  0.00041778  0.00041778  0.00041778  0.00041778
[1285]  0.00074170  0.00041778  0.00041778  0.00041778  0.00041778  0.00002843
[1291]  0.00041778  0.00074170  0.00041778  0.00041778  0.00041778  0.00041778
[1297]  0.00074170  0.00041778  0.00041778  0.00041778  0.00041778  0.00074170
[1303]  0.00041778  0.00041778  0.00041778  0.00041778  0.00041778  0.00041778
```

[1309]	0.00041778	0.00041778	0.00035607	0.00074170	0.00041778	0.00041778
[1315]	0.00041778	0.00041778	0.00035607	0.00041778	0.00074170	0.00041778
[1321]	0.00041778	0.00041778	0.00041778	0.00041778	0.00074170	0.00041778
[1327]	0.00041778	0.00041778	0.00041778	0.00036402	0.00155361	0.00155361
[1333]	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361
[1339]	0.00155361	0.00090805	0.00155361	0.00155361	0.00155361	0.00155361
[1345]	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361
[1351]	0.00090805	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361
[1357]	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361
[1363]	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361
[1369]	0.00155361	0.00155361	0.00155361	0.00090805	0.00155361	0.00155361
[1375]	0.00155361	0.00155361	0.00155361	0.00155361	0.00090805	0.00155361
[1381]	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361	0.00090805
[1387]	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361
[1393]	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361	0.00155361
[1399]	0.00017246	0.00088574	0.00056182	0.00056182	0.00056182	0.00056182
[1405]	0.00088574	0.00056182	0.00056182	0.00056182	0.00056182	0.00088574
[1411]	0.00056182	0.00056182	0.00056182	0.00056182	0.00150684	0.00056182
[1417]	0.00056182	0.00034203	0.00118292	0.00088574	0.00056182	0.00056182
[1423]	0.00056182	0.00056182	0.00056182	0.00088574	0.00056182	0.00056182
[1429]	0.00056182	0.00056182	0.00088574	0.00056182	0.00056182	0.00056182
[1435]	0.00088574	0.00056182	0.00056182	0.00056182	0.00056182	0.00088574
[1441]	0.00056182	0.00056182	0.00056182	0.00056182	0.00050010	0.00056182
[1447]	0.00088574	0.00056182	0.00056182	-0.00071537	0.00056182	0.00050010
[1453]	0.00088574	0.00056182	0.00056182	0.00056182	0.00056182	0.00050010
[1459]	0.00056182	0.00088574	0.00056182	0.00056182	0.00056182	0.00056182
[1465]	0.00050010	0.00056182	0.00088574	0.00056182	0.00056182	0.00056182
[1471]	0.00056182	0.00040322	0.00059079	0.00059079	-0.00085575	0.00034043
[1477]	0.00040322	0.00040322	0.00040322	0.00066171	0.00040322	0.00059079
[1483]	0.00059079	0.00034043	0.00040322	-0.00085575	0.00069330	0.00066171
[1489]	0.00034043	0.00072450	0.00040322	0.00040322	0.00069330	0.00072450
[1495]	0.00072450	0.00066171	-0.00085575	0.00213321	0.00072450	0.00040322
[1501]	0.00040322	0.00066171	0.00040322	0.00040322	0.00040322	0.00040322
[1507]	0.00040322	0.00072450	0.00072450	0.00040322	0.00040322	0.00037202
[1513]	0.00040322	0.00040322	0.00059079	0.00072450	0.00040322	0.00034043
[1519]	0.00145870	0.00040322	0.00040322	0.00040322	0.00072450	0.00072450
[1525]	0.00040322	0.00059079	0.00034043	0.00213321	0.00040322	0.00134538
[1531]	0.00040322	0.00040322	0.00059079	0.00072450	0.00059079	0.00072450
[1537]	0.00040322	0.00055958	0.00059079	0.00059079	0.00072450	0.00040322
[1543]	-0.00085575	0.00004366	0.00006782	0.00012953	0.00034018	0.00012953
[1549]	0.00012953	0.00012953	0.00012953	-0.00025983	0.00012953	0.00034018
[1555]	0.00012953	0.00012953	0.00012953	0.00012953	0.00006782	0.00012953
[1561]	0.00034018	0.00012953	0.00012953	0.00012953	0.00018207	0.00006782
[1567]	0.00012953	0.00034018	0.00012953	0.00012953	0.00012953	0.00012953
[1573]	0.00006782	0.00012953	0.00034018	0.00012953	0.00012953	0.00012953
[1579]	0.00012953	0.00006782	0.00012953	0.00034018	0.00012953	0.00012953
[1585]	0.00012953	0.00012953	0.00006782	0.00012953	0.00034018	0.00012953
[1591]	0.00018207	0.00012953	0.00012953	0.00006782	0.00012953	0.00034018
[1597]	0.00012953	0.00012953	0.00012953	0.00012953	0.00006782	0.00012953
[1603]	0.00034018	0.00012953	0.00012953	0.00012953	0.00012953	0.00006782
[1609]	0.00012953	0.00034018	0.00012953	0.00012953	0.00012953	0.00012953
[1615]	0.00006782	0.00012953	0.00034018	0.00012953	0.00012953	0.00012953
[1621]	0.00012953	0.00006782	0.00012953	0.00034018	0.00012953	0.00012953
[1627]	0.00012953	0.00012953	-0.00025983	0.00012953	0.00034018	0.00012953

```
[1633]  0.00012953  0.00012953  0.00012953  0.00006782  0.00038179 -0.00001991
[1639] -0.00001991 -0.00001991  0.00071996  0.00038179  0.00028310 -0.00001991
[1645]  0.00028310  0.00038179  0.00038179 -0.00001991  0.00038179  0.00038179
[1651]  0.00071996  0.00038179 -0.00001991  0.00028310 -0.00127888  0.00268309
[1657]  0.00038179 -0.00001991 -0.00001991  0.00038179  0.00038179 -0.00001991
[1663] -0.00001991 -0.00001991 -0.00001991 -0.00001991 -0.00001991  0.00028310
[1669]  0.00038179 -0.00001991  0.00038179  0.00038179  0.00004233  0.00038179
[1675]  0.00038179 -0.00001991  0.00038179  0.00038179  0.00038179 -0.00001991
[1681] -0.00001991  0.00038179  0.00071996  0.00038179  0.00038179  0.00038179
[1687]  0.00038179  0.00025927  0.00038179  0.00038179  0.00038179  0.00038179
[1693]  0.00038179 -0.00026067  0.00071996 -0.00001991  0.00038179  0.00038179
[1699]  0.00038179  0.00046471  0.00049591  0.00070657  0.00049591  0.00049591
[1705]  0.00049591  0.00010656  0.00070657  0.00049591  0.00049591  0.00049591
[1711]  0.00049591  0.00046471  0.00049591  0.00049591  0.00049591  0.00049591
[1717]  0.00046471  0.00049591  0.00070657  0.00049591  0.00049591  0.00049591
[1723]  0.00049591  0.00010656  0.00070657  0.00049591  0.00049591  0.00046471
[1729]  0.00049591  0.00049591  0.00049591  0.00049591  0.00070657  0.00049591
[1735]  0.00049591  0.00049591  0.00046471  0.00049591  0.00070657  0.00049591
[1741]  0.00049591  0.00049591  0.00010656  0.00070657  0.00049591  0.00049591
[1747]  0.00049591  0.00046471  0.00049591  0.00070657  0.00049591  0.00049591
[1753]  0.00049591  0.00049591  0.00046471  0.00070657  0.00021501  0.00049591
[1759]  0.00049591  0.00049591  0.00046471  0.00049591  0.00070657  0.00049591
[1765]  0.00049591  0.00049591  0.00049591  0.00049591  0.00070657  0.00049591
[1771]  0.00049591  0.00049591  0.00049591  0.00050720  0.00046689  0.00074170
[1777]  0.00052860  0.00052860  0.00052860  0.00052860  0.00046689  0.00074170
[1783]  0.00052860  0.00052860  0.00052860  0.00052860  0.00040608  0.00074170
[1789]  0.00052860  0.00052860  0.00052860  0.00052860  0.00046689  0.00074170
[1795]  0.00052860  0.00052860  0.00052860  0.00052860  0.00046689  0.00074170
[1801]  0.00052860  0.00052860  0.00052860  0.00052860  0.00046689  0.00074170
[1807]  0.00052860  0.00052860  0.00052860  0.00046581  0.00046689  0.00074170
[1813]  0.00052860  0.00052860  0.00052860  0.00052860  0.00046689  0.00052860
[1819]  0.00074170  0.00052860  0.00052860  0.00052860  0.00052860  0.00040608
[1825]  0.00074170  0.00052860  0.00052860  0.00052860  0.00052860  0.00046689
[1831]  0.00052860  0.00074170  0.00052860  0.00052860  0.00052860  0.00052860
[1837]  0.00074170  0.00052860  0.00052860  0.00052860  0.00052860  0.00074170
[1843]  0.00052860  0.00052860  0.00052860  0.00052860  0.00046689  0.00074170
[1849]  0.00052860  0.00052860  0.00052860  0.00052860  0.00445127  0.00022916
[1855]  0.00040299  0.00046992  0.00046992  0.00046992  0.00046992  0.00022916
[1861]  0.00040299  0.00046992  0.00060891  0.00046992  0.00046992  0.00040299
[1867]  0.00046992  0.00046992  0.00046992  0.00046992  0.00040299  0.00046992
[1873]  0.00046992  0.00046992  0.00046992  0.00156062  0.00046992  0.00060891
[1879]  0.00046992  0.00046992  0.00022916  0.00040299  0.00046992  0.00046992
[1885]  0.00046992  0.00046992  0.00040299  0.00046992  0.00046992  0.00046992
[1891]  0.00124339  0.00040299  0.00046992  0.00046992  0.00046992  0.00046992
[1897]  0.00022916  0.00040299  0.00046992  0.00046992  0.00046992  0.00046992
[1903]  0.00040299  0.00046992  0.00046992  0.00046992  0.00046992  0.00008057
[1909]  0.00040299  0.00046992  0.00046992  0.00046992  0.00046992  0.00040299
[1915]  0.00046992  0.00046992  0.00046992  0.00046992  0.00040299  0.00046992
[1921]  0.00046992  0.00046992  0.00046992  0.00082588  0.00048591  0.00069901
[1927] -0.00082564 -0.00082564  0.00048591  0.00048591  0.00082588  0.00048591
[1933]  0.00069901  0.00048591  0.00048591  0.00045793 -0.00082564  0.00013014
[1939]  0.00045793  0.00067103  0.00045793  0.00275923  0.00045793  0.00048591
[1945]  0.00082588  0.00048591  0.00069901  0.00045793  0.00029834  0.00048591
[1951]  0.00045793  0.00082588  0.00048591  0.00076058  0.00054748  0.00054748
```

```
[1957]  0.00054748  0.00051950  0.00079790  0.00051950  0.00073260 -0.00076406
[1963]  0.00054748  0.00082588  0.00054748  0.00076058 -0.00076406  0.00054748
[1969]  0.00054748  0.00309920  0.00282080  0.00303390  0.00051950 -0.00076406
[1975]  0.00051950  0.00013014  0.00051950  0.00073260  0.00054748  0.00035992
[1981]  0.00051950  0.00054748 -0.00048566 -0.00055096  0.00054748 -0.00076406
[1987]  0.00051950  0.00054748  0.00076058  0.00035992  0.00051950  0.00054748
[1993]  0.00282080 -0.00048566 -0.00076406 -0.00055096  0.00054748  0.00054748
[1999]  0.00054748  0.00051950  0.00013014  0.00051950  0.00073260  0.00054748
[2005]  0.00054748  0.00054748 -0.00076406  0.00024600  0.00036852  0.00039389
[2011]  0.00036852  0.00036852  0.00036852  0.00036852  0.00024600  0.00039389
[2017]  0.00036852  0.00036852  0.00036852  0.00036852  0.00024600  0.00036852
[2023]  0.00036852  0.00036852  0.00036852  0.00036852  0.00030681  0.00036852
[2029]  0.00039389  0.00036852  0.00036852  0.00036852  0.00036852  0.00036852
[2035]  0.00036852  0.00036852  0.00036852  0.00024600  0.00039389  0.00036852
[2041]  0.00036852  0.00036852  0.00024600  0.00039389  0.00036852  0.00036852
[2047]  0.00036852  0.00039389  0.00036852  0.00036852  0.00036852  0.00036852
[2053]  0.00039389  0.00036852  0.00036852  0.00036852  0.00024600  0.00036852
[2059]  0.00039389  0.00036852  0.00036852  0.00036852  0.00036852  0.00036852
[2065]  0.00039389  0.00036852  0.00036852  0.00036852  0.00036852  0.00024600
[2071]  0.00036852  0.00039389  0.00036852  0.00036852  0.00036852  0.00036852
[2077]  0.00030681  0.00036852  0.00039389  0.00036852  0.00036852  0.00036852
[2083]  0.00036852  0.00024600  0.00052265  0.00052265  0.00052265  0.00195795
[2089] -0.00038243  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265
[2095]  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265
[2101]  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265
[2107]  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265  0.00195795
[2113]  0.00052265  0.00049144  0.00052265  0.00052265  0.00052265  0.00049144
[2119]  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265  0.00049144
[2125]  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265
[2131]  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265
[2137]  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265  0.00052265
[2143]  0.00052265  0.00052265  0.00074255  0.00074269  0.00058603  0.00085680
[2149]  0.00085680  0.00074269  0.00074269  0.00074255  0.00096760  0.00081094
[2155]  0.00074269  0.00096760  0.00074269  0.00074269  0.00074255  0.00074269
[2161]  0.00058603  0.00074269  0.00096760  0.00096760  0.00096760  0.00074255
[2167]  0.00074269  0.00097019  0.00074269  0.00074269  0.00074269  0.00074269
[2173]  0.00074255  0.00074269  0.00011708  0.00027374  0.00027374  0.00110878
[2179]  0.00027374  0.00011708  0.00049865  0.00027374  0.00049865  0.00011553
[2185]  0.00011708  0.00027374  0.00027374  0.00027374  0.00110878  0.00027360
[2191]  0.00011708  0.00027374  0.00049865  0.00049865  0.00027374  0.00011708
[2197]  0.00027374  0.00049865  0.00207905  0.00207905  0.00185401  0.00268919
[2203]  0.00185415  0.00185415  0.00185415  0.00185401  0.00169748  0.00185415
[2209]  0.00169594  0.00185415  0.00185415  0.00185401  0.00169748  0.00185415
[2215]  0.00268919  0.00185415  0.00185415  0.00185401  0.00169748  0.00207905
[2221]  0.00185415  0.00207905  0.00207905  0.00265622  0.00271793  0.00001493
[2227]  0.00001493  0.00271793  0.00031794 -0.00004678  0.00078017  0.00041663
[2233]  0.00041663  0.00001493  0.00041663  0.00041663  0.00044200  0.00001493
[2239]  0.00041663  0.00041663  0.00001493 -0.00004678  0.00001493  0.00078017
[2245]  0.00041663  0.00031794  0.00031794  0.00001493 -0.00004678  0.00078017
[2251]  0.00001493  0.00001493  0.00001493  0.00001493  0.00035492  0.00044200
[2257]  0.00001493  0.00001493  0.00001493  0.00001493  0.00031794  0.00034331
[2263]  0.00001493  0.00041663  0.00025622  0.00031794  0.00034331  0.00001493
[2269]  0.00001493  0.00001493  0.00041663 -0.00004678  0.00078017  0.00001493
[2275]  0.00001493  0.00001493  0.00041663 -0.00004678  0.00001493  0.00078017
```

```
[2281]  0.00001493  0.00001493  0.00001493  0.00001493 -0.00004678  0.00031794
[2287]  0.00001493  0.00001493  0.00041663  0.00001493  0.00078017  0.00041663
[2293]  0.00001493  0.00001493  0.00001493  0.00078017  0.00001493  0.00001493
[2299]  0.00001493  0.00001493  0.00071320  0.00071320  0.00071320  0.00049341
[2305]  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320
[2311]  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320
[2317]  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320
[2323]  0.00071320  0.00071320  0.00006764  0.00071320  0.00071320  0.00071320
[2329] -0.00056399  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320
[2335]  0.00006764  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320
[2341]  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320
[2347]  0.00071320  0.00071320  0.00071320  0.00071320  0.00071320  0.00036852
[2353]  0.00036852  0.00036852  0.00036852  0.00036852  0.00036852  0.00036852
[2359]  0.00024600  0.00069244  0.00036852  0.00036852  0.00036852  0.00036852
[2365]  0.00069244  0.00036852  0.00036852  0.00036852  0.00069244  0.00036852
[2371]  0.00036852  0.00036852  0.00036852  0.00069244  0.00036852  0.00036852
[2377]  0.00036852  0.00036852  0.00024600  0.00036852  0.00036852  0.00036852
[2383]  0.00064692  0.00069244  0.00036852  0.00069244  0.00036852  0.00313355
[2389]  0.00036852  0.00036852  0.00069244  0.00036852  0.00036852  0.00036852
[2395]  0.00069244  0.00036852  0.00036852  0.00036852  0.00036852  0.00069244
[2401]  0.00036852  0.00036852  0.00036852  0.00033731  0.00124203  0.00124203
[2407]  0.00124203  0.00124203  0.00124203  0.00183898  0.00124203  0.00124203
[2413]  0.00124203  0.00183898  0.00124203  0.00124203  0.00124203  0.00124203
[2419]  0.00100127  0.00129457  0.00189152  0.00124203  0.00124203  0.00124203
[2425]  0.00183898  0.00124203  0.00124203  0.00124203  0.00124203  0.00183898
[2431]  0.00124203  0.00124203  0.00124203  0.00183898  0.00124203  0.00124203
[2437]  0.00124203  0.00124203  0.00124203  0.00124203  0.00124203  0.00124203
[2443]  0.00124203  0.00124203  0.00124203  0.00124203  0.00183898  0.00124203
[2449]  0.00124203  0.00124203  0.00129457  0.00100127  0.00124203  0.00183898
[2455]  0.00124203  0.00004111  0.00007231  0.00007746  0.00007231  0.00007231
[2461]  0.00007231  0.00007231  0.00004111  0.00007231  0.00007746  0.00007231
[2467]  0.00007231  0.00007231  0.00007231  0.00004111  0.00007231  0.00007746
[2473]  0.00007231  0.00007231  0.00007231  0.00007231  0.00004111  0.00007231
[2479]  0.00007746  0.00007231  0.00007231  0.00007231  0.00007231  0.00004111
[2485]  0.00007231  0.00007746  0.00007231  0.00007231  0.00007231  0.00007231
[2491] -0.00005021  0.00007231  0.00007746  0.00007231  0.00007231  0.00007231
[2497]  0.00007231  0.00004111  0.00007231  0.00007746  0.00007231  0.00007231
[2503]  0.00007231  0.00007231  0.00004111  0.00007231  0.00007746  0.00007231
[2509]  0.00007231  0.00007231  0.00007231  0.00004111  0.00007231  0.00007746
[2515]  0.00007231  0.00007231  0.00052363  0.00052363  0.00049243  0.00052363
[2521]  0.00073429  0.00052363  0.00052363  0.00052363  0.00052363  0.00049243
[2527]  0.00052363  0.00073429  0.00052363  0.00052363  0.00052363  0.00052363
[2533]  0.00049243  0.00052363  0.00073429  0.00052363  0.00046084  0.00046084
[2539]  0.00052363  0.00049243  0.00052363  0.00073429  0.00052363  0.00052363
[2545]  0.00052363  0.00052363  0.00049243 -0.00026793  0.00067791  0.00067791
[2551]  0.00067791  0.00067791  0.00067791  0.00067791 -0.00026793  0.00067791
[2557]  0.00067791  0.00067791  0.00067791  0.00067791  0.00067791 -0.00026793
[2563]  0.00067791  0.00067791  0.00067791  0.00067791  0.00067791  0.00067791
[2569] -0.00026793  0.00067791  0.00067791  0.00067791  0.00067791  0.00067791
[2575]  0.00067791 -0.00026793  0.00067791  0.00067791  0.00067791  0.00067791
[2581]  0.00067791  0.00067791 -0.00026793  0.00067791  0.00067791  0.00067791
[2587]  0.00067791  0.00067791  0.00067791 -0.00026793  0.00067791  0.00067791
[2593]  0.00067791  0.00067791  0.00067791  0.00067791 -0.00026793  0.00067791
[2599]  0.00067791  0.00067791  0.00067791  0.00067791  0.00067791 -0.00026793
```

[2605]	0.00067791	0.00067791	0.00067791	0.00067791	0.00067791	0.00067791
[2611]	-0.00026793	0.00067791	0.00067791	0.00067791	0.00067791	0.00067791
[2617]	0.00067791	-0.00026793	0.00067791	0.00067791	0.00067791	0.00067791
[2623]	0.00067791	0.00067791	0.00067791	0.00067791	0.00067791	0.00067791
[2629]	0.00067791	-0.00026793	0.00067791	0.00067791	0.00067791	0.00067791
[2635]	0.00067791	0.00067791	0.00020489	0.00016544	0.00013507	0.00014022
[2641]	0.00035998	0.00013507	0.00013507	0.00016544	0.00013507	0.00014022
[2647]	0.00024918	0.00013507	0.00013507	0.00013507	0.00016544	0.00013507
[2653]	0.00014022	0.00013507	0.00013507	0.00013507	0.00013507	0.00016544
[2659]	0.00013507	0.00036012	0.00013507	0.00013507	0.00013507	0.00035998
[2665]	0.00016544	0.00024918	0.00025433	0.00035998	0.00013507	0.00035998
[2671]	0.00035998	-0.00004412	0.00013507	0.00014022	0.00013507	0.00013507
[2677]	0.00013507	0.00013507	-0.00004412	0.00035998	0.00036513	0.00013507
[2683]	0.00013507	0.00035998	0.00013507	-0.00004412	0.00024918	0.00047423
[2689]	0.00024918	0.00035998	0.00035998	0.00024918	-0.00004412	0.00035998
[2695]	0.00036513	0.00013507	0.00013507	0.00013507	0.00013507	-0.00004412
[2701]	0.00035998	0.00036513	0.00035998	0.00013507	0.00013507	0.00013507
[2707]	-0.00004412	0.00013507	0.00014022	0.00024918	0.00035998	0.00013507
[2713]	0.00013507	-0.00004412	0.00013507	0.00014022	0.00024918	0.00024918
[2719]	0.00013507	0.00024918	-0.00004412	0.00013507	0.00014022	0.00035998
[2725]	0.00013507	0.00013507	0.00013507	-0.00069104	0.00025715	0.00016144
[2731]	0.00031810	0.00031810	0.00031810	0.00031810	0.00025715	0.00016144
[2737]	0.00031810	0.00031810	0.00031810	0.00031810	0.00025715	0.00031810
[2743]	0.00016144	0.00031810	0.00031810	0.00031810	0.00031810	0.00025715
[2749]	0.00016144	0.00031810	0.00031810	0.00031810	0.00031810	0.00025715
[2755]	0.00031810	0.00016144	0.00031810	0.00031810	0.00031810	0.00031810
[2761]	0.00025715	0.00031810	0.00016144	0.00031810	0.00031810	0.00031810
[2767]	0.00031810	0.00025715	0.00031810	0.00016144	0.00031810	0.00031810
[2773]	0.00031810	0.00031810	0.00025715	0.00031810	0.00016144	0.00031810
[2779]	0.00031810	0.00031810	0.00031810	0.00025715	0.00031810	0.00016144
[2785]	0.00031810	0.00031810	0.00031810	0.00031810	0.00025715	0.00031810
[2791]	0.00016144	0.00031810	0.00031810	0.00031810	0.00031810	0.00025715
[2797]	0.00031810	0.00016144	0.00031810	0.00031810	0.00031810	0.00031810
[2803]	0.00031810	0.00016144	0.00031810	0.00031810	0.00031810	0.00031810
[2809]	0.00025715	0.00031810	0.00016144	0.00031810	0.00031810	0.00031810
[2815]	0.00031810	0.00025715	0.00031037	0.00011496	0.00046703	0.00046703
[2821]	0.00046703	0.00031037	0.00011496	0.00046703	0.00046703	0.00046703
[2827]	0.00046703	0.00031037	0.00046703	0.00046703	0.00046703	0.00046703
[2833]	0.00046703	0.00031037	0.00046703	0.00046703	0.00046703	0.00046703
[2839]	-0.00004170	0.00046703	0.00011496	0.00046703	0.00031037	0.00046703
[2845]	0.00046703	0.00046703	0.00046703	0.00046703	0.00031037	0.00046703
[2851]	0.00046703	0.00011496	0.00058114	0.00069194	0.00053527	0.00046703
[2857]	0.00046703	0.00046703	0.00066759	0.00031037	0.00046703	0.00046703
[2863]	0.00046703	0.00046703	0.00046703	0.00031037	0.00058114	0.00046703
[2869]	0.00046703	0.00046703	0.00031037	0.00046703	0.00046703	0.00046703
[2875]	0.00046703	0.00046703	0.00031037	0.00046703	0.00046703	0.00046703
[2881]	0.00046703	0.00031037	0.00046703	0.00046703	0.00046703	0.00046703
[2887]	0.00036402	0.00042574	0.00045110	0.00042574	0.00042574	0.00042574
[2893]	0.00042574	0.00036402	0.00042574	0.00045110	0.00042574	0.00042574
[2899]	0.00042574	0.00042574	0.00030322	0.00042574	0.00045110	0.00042574
[2905]	0.00042574	0.00042574	0.00042574	0.00036402	0.00042574	0.00045110
[2911]	0.00042574	0.00042574	0.00042574	0.00042574	0.00030322	0.00042574
[2917]	0.00045110	0.00042574	0.00042574	0.00042574	0.00042574	0.00036402
[2923]	0.00042574	0.00045110	0.00042574	0.00042574	0.00042574	0.00047827

```
[2929]  0.00036402  0.00042574  0.00045110  0.00042574  0.00042574  0.00042574
[2935]  0.00042574  0.00036402  0.00042574  0.00045110  0.00042574  0.00042574
[2941]  0.00042574  0.00042574  0.00030322  0.00042574  0.00045110  0.00042574
[2947]  0.00042574  0.00042574  0.00042574  0.00036402  0.00042574  0.00045110
[2953]  0.00042574  0.00042574  0.00042574  0.00042574  0.00036402  0.00042574
[2959]  0.00045110  0.00042574  0.00042574  0.00042574  0.00319076  0.00042574
[2965]  0.00045110  0.00042574  0.00042574  0.00042574  0.00042574  0.00042574
[2971]  0.00045110  0.00042574  0.00042574  0.00042574  0.00042574  0.00035607
[2977]  0.00029109  0.00035280  0.00037265  0.00037265  0.00037265  0.00035280
[2983]  0.00029109  0.00035280  0.00035280  0.00037265  0.00035280 -0.00053243
[2989]  0.00037265  0.00037265  0.00037265  0.00037265  0.00037265  0.00035280
[2995]  0.00037265  0.00031094  0.00037265  0.00037265  0.00035280  0.00037265
[3001]  0.00035280  0.00063221  0.00063113  0.00037265  0.00037265  0.00037265
[3007]  0.00035280  0.00029109  0.00035280 -0.00008134  0.00037265  0.00037265
[3013]  0.00037265 -0.00014305 -0.00008134 -0.00008134  0.00035280  0.00037265
[3019]  0.00037265  0.00037265 -0.00014305 -0.00008134 -0.00008134  0.00037265
[3025]  0.00035280  0.00037265  0.00037265  0.00029109  0.00035280  0.00035280
[3031]  0.00030986  0.00037265  0.00035280  0.00037265  0.00035280  0.00035280
[3037]  0.00037265  0.00037265  0.00037265  0.00035280  0.00035280 -0.00008134
[3043] -0.00008134  0.00035280  0.00037265  0.00035280  0.00035280  0.00035280
[3049]  0.00037265  0.00037265  0.00037265  0.00031094  0.00037265  0.00037265
[3055] -0.00008134  0.00046934  0.00046948  0.00038158  0.00046948  0.00046948
[3061]  0.00046948  0.00046948  0.00038158  0.00046948  0.00046948  0.00046948
[3067]  0.00046948  0.00046934  0.00069438  0.00060649  0.00046948  0.00069438
[3073]  0.00046948  0.00046948  0.00046934  0.00038158  0.00046948  0.00046948
[3079]  0.00046948  0.00046948  0.00071065  0.00005867  0.00005867  0.00005867
[3085]  0.00032294  0.00005853  0.00071065  0.00005867  0.00005867  0.00005867
[3091]  0.00017278  0.00005853  0.00005867  0.00071065  0.00017278  0.00017278
[3097]  0.00005867  0.00017278  0.00005853  0.00028358  0.00093555  0.00005867
[3103]  0.00005867  0.00005867 -0.00082319  0.00005853  0.00017278  0.00005867
[3109]  0.00005867  0.00005867  0.00005867  0.00005853  0.00005867  0.00071065
[3115]  0.00005867  0.00005867  0.00005867  0.00005867  0.00005853  0.00005867
[3121]  0.00071065  0.00005867  0.00005867  0.00028358  0.00005867  0.00005853
[3127]  0.00005867  0.00071065  0.00005867  0.00005867  0.00005867  0.00046037
[3133] -0.00082333 -0.00082319 -0.00091109  0.00005867  0.00028358  0.00005867
[3139]  0.00005867  0.00005853  0.00012146  0.00051093  0.00046948  0.00046948
[3145]  0.00031127  0.00012146  0.00031037  0.00046948  0.00046948  0.00046948
[3151]  0.00046948  0.00031037  0.00046948  0.00046948  0.00046948  0.00046948
[3157]  0.00069438  0.00069438  0.00058358  0.00046948  0.00031037  0.00069438
[3163]  0.00031127  0.00046948  0.00046948  0.00031037  0.00046948  0.00046948
[3169]  0.00046948  0.00046948  0.00031037  0.00046948  0.00046948  0.00046948
[3175]  0.00012146  0.00046948  0.00031037  0.00046948  0.00046948  0.00046948
[3181]  0.00046948  0.00012146  0.00031037  0.00046948  0.00046948  0.00046948
[3187]  0.00031037  0.00046948  0.00058358  0.00046948  0.00058358  0.00091943
[3193]  0.00046948  0.00046948  0.00046948  0.00046948  0.00031037  0.00046948
[3199]  0.00046948  0.00046948  0.00046948  0.00012146  0.00046948  0.00031037
[3205]  0.00046948  0.00046948  0.00046948  0.00036378  0.00036378  0.00036378
[3211]  0.00023711  0.00033580  0.00033580  0.00036378  0.00036378  0.00036378
[3217]  0.00033580  0.00033580  0.00023711  0.00036378  0.00033580  0.00036378
[3223]  0.00036378  0.00023697  0.00023711  0.00023711  0.00033580  0.00033580
[3229]  0.00036378  0.00033580  0.00036378  0.00088630  0.00036378  0.00065576
[3235]  0.00040113  0.00036378  0.00036378  0.00036378  0.00036378  0.00036378
[3241]  0.00033580  0.00036378  0.00036378  0.00036364  0.00036378  0.00036378
[3247]  0.00036378  0.00036378  0.00036378  0.00036378  0.00023697  0.00023711
```

[3253]	0.00036378	0.00033580	0.00017500	0.00036364	0.00036378	0.00036378
[3259]	0.00033580	0.00036378	0.00036378	0.00033566	0.00033580	0.00033580
[3265]	0.00065698	0.00036378	0.00036378	0.00065684	0.00088189	0.00033580
[3271]	0.00033580	0.00036378	0.00036378	0.00036364	0.00016544	0.00053056
[3277]	0.00053056	0.00053056	0.00047423	0.00053056	0.00053056	0.00053056
[3283]	0.00053571	0.00053056	0.00053056	0.00053056	0.00053056	0.00053571
[3289]	0.00024918	0.00053056	0.00053571	0.00053056	0.00053056	0.00075561
[3295]	0.00053056	0.00053056	0.00053056	0.00053056	0.00024918	0.00053571
[3301]	0.00053056	0.00053056	0.00053056	0.00053571	0.00053056	0.00053056
[3307]	0.00053056	0.00053056	0.00053056	0.00053056	0.00067380	0.00153116
[3313]	0.00098099	0.00070010	0.00181205	0.00098099	0.00098099	0.00098099
[3319]	0.00070010	0.00098099	0.00181205	0.00098099	0.00098099	0.00098099
[3325]	0.00098099	0.00181205	0.00098099	0.00168803	0.00168803	0.00105885
[3331]	0.00048267	0.00048267	0.00048267	0.00105885	0.00048267	0.00048267
[3337]	0.00048267	0.00105885	0.00048267	0.00020177	0.00105885	0.00048267
[3343]	0.00048267	0.00048267	0.00105885	0.00020177	0.00020177	0.00020177
[3349]	0.00020177	0.00020177	0.00020177	0.00020177	0.00020177	0.00105885
[3355]	0.00020177	0.00048267	0.00048267	0.00048267	0.00048267	0.00105885
[3361]	0.00048267	0.00048267	0.00048267	0.00048267	0.00077795	0.00048267
[3367]	0.00048267	0.00020177	0.00048267	0.00094204	0.00062332	0.00062332
[3373]	0.00062332	0.00094204	0.00062332	0.00062332	0.00062332	0.00062332
[3379]	0.00065369	0.00062332	0.00094204	0.00062332	0.00062332	0.00062332
[3385]	0.00117747	0.00065369	0.00062332	0.00062332	0.00062332	0.00034242
[3391]	0.00062332	0.00094204	0.00062577	0.00062577	0.00062577	0.00062577
[3397]	0.00094204	0.00062577	0.00034487	0.00062577	0.00062577	0.00062577
[3403]	0.00062577	0.00062577	0.00034487	0.00065614	0.00094204	0.00062577
[3409]	0.00062577	0.00062577	0.00065614	0.00062577	0.00094204	0.00062577
[3415]	0.00062577	0.00062577	0.00062577	0.00065614	0.00062577	0.00062577
[3421]	0.00117991	0.00062577	0.00062577	0.00066114	0.00062577	0.00062577
[3427]	0.00062577	0.00062577	0.00065614	0.00062577	0.00094204	0.00062577
[3433]	0.00062577	0.00062577	0.00094204	0.00062577	0.00062577	0.00062577
[3439]	0.00062577	0.00039111	0.00240002	0.00240002	0.00240002	0.00240002
[3445]	0.00240002	0.00240002	0.00211912	0.00240002	0.00240002	0.00240002
[3451]	0.00240002	0.00240002	0.00240002	0.00240002	0.00240002	0.00240002
[3457]	0.00240002	0.00240002	0.00240002	0.00240002	0.00240002	0.00240002
[3463]	0.00211912	0.00057305	0.00078615	0.00053571	0.00053571	0.00033245
[3469]	0.00078615	0.00057305	0.00053571	0.00054870	0.00053571	0.00053571
[3475]	0.00074881	0.00053571	0.00053571	0.00053571	0.00052001	0.00074881
[3481]	0.00040903	0.00053571	0.00034814	0.00053571	0.00053571	0.00053571
[3487]	0.00053571	0.00053571	0.00053571	0.00049203	0.00050772	0.00040903
[3493]	0.00053571	0.00053571	0.00074881	0.00053571	0.00280902	0.00049203
[3499]	0.00072082	0.00053571	0.00050772	0.00053571	0.00053571	0.00076061
[3505]	0.00050772	0.00072082	0.00050772	0.00034814	0.00053571	0.00040903
[3511]	0.00052001	0.00053571	0.00074881	0.00034814	0.00053571	0.00052001
[3517]	0.00053571	0.00074881	0.00053571	0.00053571	0.00053571	0.00039334
[3523]	0.00040903	0.00062213	0.00053571	0.00053571	0.00057305	0.00052001
[3529]	0.00009732	0.00015827	0.00015827	0.00015827	0.00015827	0.00015827
[3535]	0.00015827	0.00009732	0.00015827	0.00015827	0.00015827	0.00015827
[3541]	0.00015827	0.00015827	0.00009732	0.00015827	0.00015827	0.00015827
[3547]	0.00015827	0.00015827	0.00015827	0.00009732	0.00015827	0.00015827
[3553]	0.00015827	0.00015827	0.00015827	0.00015827	0.00009732	0.00015827
[3559]	0.00015827	0.00015827	0.00015827	0.00015827	0.00015827	0.00009732
[3565]	0.00015827	0.00015827	0.00015827	0.00015827	0.00015827	0.00015827
[3571]	0.00009732	0.00015827	0.00015827	0.00015827	0.00015827	0.00015827

```
[3577]  0.00015827  0.00009732  0.00015827  0.00015827  0.00015827  0.00015827
[3583]  0.00015827  0.00015827  0.00009732  0.00015827  0.00015827  0.00015827
[3589]  0.00015827  0.00014916  0.00014916  0.00008821  0.00014916  0.00014916
[3595]  0.00014916  0.00014916  0.00014916  0.00014916  0.00008821  0.00014916
[3601]  0.00014916  0.00014916  0.00014916  0.00014916  0.00014916  0.00008821
[3607]  0.00014916  0.00014916  0.00014916  0.00014916  0.00014916  0.00014916
[3613]  0.00008821  0.00014916  0.00014916  0.00014916  0.00014916  0.00014916
[3619]  0.00014916  0.00009732  0.00002414  0.00002414  0.00002414  0.00034198
[3625]  0.00002414  0.00002414  0.00002414  0.00011708  0.00002414  0.00002414
[3631]  0.00002414  0.00002414  0.00011708  0.00002414  0.00002414  0.00002414
[3637]  0.00002414  0.00011708  0.00002414  0.00013825  0.00011708  0.00002414
[3643]  0.00013825  0.00005451  0.00011708  0.00002414  0.00002414  0.00013825
[3649]  0.00011708  0.00002414  0.00002414  0.00024905  0.00024905  0.00002414
[3655]  0.00011708  0.00002414  0.00002414  0.00024905  0.00002414  0.00005451
[3661]  0.00011708  0.00002414  0.00002414  0.00002414  0.00002414  0.00013825
[3667]  0.00023119  0.00002414  0.00024905  0.00002414  0.00002414  0.00011708
[3673]  0.00002414  0.00002414  0.00002414  0.00002414  0.00024905  0.00034198
[3679]  0.00024905  0.00002414  0.00024905  0.00024905  0.00029999  0.00032283
[3685]  0.00032283  0.00032283  0.00027915  0.00029485  0.00029485  0.00032283
[3691]  0.00029485  0.00019615  0.00029485  0.00029485  0.00019615  0.00029485
[3697]  0.00029485  0.00032283  0.00019615  0.00032283  0.00036017  0.00014041
[3703]  0.00054773  0.00032283  0.00036017  0.00032283  0.00020130  0.00029485
[3709]  0.00032283  0.00029485  0.00029485  0.00020130  0.00019615  0.00036017
[3715]  0.00029485  0.00027915  0.00029485  0.00029485  0.00032283  0.00029999
[3721]  0.00029485  0.00019615  0.00032283  0.00029485  0.00027915  0.00029999
[3727]  0.00029485  0.00032283  0.00013527  0.00029485  0.00027915  0.00029999
[3733]  0.00019615  0.00024937  0.00032283  0.00029485  0.00011957  0.00013527
[3739]  0.00036031  0.00032283  0.00032283  0.00032283  0.00036017  0.00025315
[3745]  0.00041226  0.00041226  0.00041226  0.00041226  0.00025315  0.00041226
[3751]  0.00041226  0.00041226  0.00041226  0.00025315  0.00041226  0.00041226
[3757]  0.00041226  0.00041226  0.00025315  0.00041226  0.00041226  0.00052637
[3763]  0.00041226  0.00025315  0.00041226  0.00041226  0.00041226  0.00041226
[3769]  0.00025315  0.00041226  0.00041226  0.00041226  0.00041226  0.00025315
[3775]  0.00041226  0.00041226  0.00041226  0.00041226  0.00006425  0.00025315
[3781]  0.00041226  0.00041226  0.00041226  0.00041226  0.00006425  0.00041226
[3787]  0.00025315  0.00041226  0.00041226  0.00041226  0.00041226  0.00025315
[3793]  0.00041226  0.00041226  0.00041226  0.00061282  0.00063716  0.00086221
[3799]  0.00063716  0.00041226  0.00041226  0.00041226  0.00041226  0.00025315
[3805]  0.00041226  0.00041226  0.00041226  0.00041226  0.00006425  0.00041226
[3811]  0.00025315  0.00041226  0.00041226  0.00041226  0.00041226  0.00069987
[3817] -0.00175216  0.00180904  0.00158414  0.00171754  0.00158414  0.00158414
[3823]  0.00171754 -0.00175216  0.00180904  0.00171754  0.00158414  0.00157885
[3829]  0.00171754  0.00158414  0.00102637  0.00171754  0.00339789  0.00339789
[3835]  0.00339789  0.00157885  0.00251603  0.00339789  0.00339789  0.00339789
[3841]  0.00339789  0.00080146  0.00339789  0.00171754  0.00171754  0.00339789
[3847]  0.00157885  0.00171754  0.00171754  0.00171754  0.00171754  0.00339789
[3853]  0.00383759  0.00215724  0.00215724  0.00383759  0.00406250  0.00383759
[3859]  0.00383759  0.00215724  0.00383759  0.00215724  0.00383759  0.00383759
[3865]  0.00215724  0.00383759  0.00383759  0.00383759  0.00383759  0.00072984
[3871]  0.00232621  0.00067906  0.00062237  0.00071027  0.00071027  0.00071027
[3877]  0.00067906  0.00062237  0.00071027  0.00067906  0.00071027  0.00062237
[3883]  0.00071027  0.00071027  0.00067906  0.00062237  0.00071027  0.00071027
[3889]  0.00071027  0.00039817  0.00034148  0.00071027  0.00071027  0.00071027
[3895]  0.00067906  0.00062237  0.00071027  0.00067906  0.00062237  0.00071027
```

```
[3901]  0.00071027  0.00071027  0.00067906  0.00062237  0.00071027  0.00071027
[3907]  0.00032091  0.00062237  0.00071027  0.00151267  0.00145598  0.00154387
[3913]  0.00126298  0.00123177  0.00117508  0.00154387  0.00154387  0.00154387
[3919]  0.00154387  0.00115452  0.00154387  0.00145598  0.00154387  0.00154387
[3925]  0.00154387  0.00154387  0.00151267  0.00154387  0.00145598  0.00154387
[3931]  0.00154387  0.00154387  0.00154387  0.00151267  0.00029618  0.00032739
[3937]  0.00033253  0.00171157  0.00171157  0.00032739  0.00125911  0.00032739
[3943]  0.00055243  0.00125911  0.00032739  0.00125911  0.00032739  0.00029618
[3949]  0.00032739  0.00033253  0.00032739  0.00032739  0.00032739  0.00032739
[3955]  0.00148415  0.00032739  0.00032739  0.00125911  0.00032739  0.00029618
[3961]  0.00072326  0.00071811  0.00077065  0.00077065  0.00071811 -0.00022773
[3967]  0.00071811  0.00072326  0.00077065  0.00071811  0.00071811  0.00077065
[3973] -0.00022773  0.00071811  0.00094316  0.00071811  0.00071811  0.00071811
[3979]  0.00071811 -0.00022773  0.00071811  0.00094316  0.00071811  0.00071811
[3985]  0.00071811  0.00071811 -0.00022773  0.00071811  0.00072326  0.00071811
[3991]  0.00071811  0.00077065  0.00071811  0.00077065  0.00077065  0.00071811
[3997]  0.00071811  0.00071811  0.00071811  0.00094316  0.00071811  0.00071811
[4003]  0.00071811  0.00071811 -0.00022773  0.00071811  0.00072326  0.00071811
[4009]  0.00071811  0.00071811  0.00071811 -0.00022773  0.00071811  0.00094316
[4015]  0.00071811  0.00071811  0.00071811  0.00071811  0.00075014  0.00222309
[4021]  0.00200040  0.00075014  0.00075014  0.00340260  0.00340260  0.00224748
[4027]  0.00224748  0.00099721  0.00130022  0.00099721  0.00224748  0.00099721
[4033]  0.00130022  0.00099721  0.00130022  0.00099721  0.00139891  0.00099721
[4039]  0.00099721  0.00099721  0.00139891  0.00404282  0.00099721  0.00099721
[4045]  0.00193561  0.00120239  0.00120239  0.00142730  0.00120239  0.00216051
[4051]  0.00120239  0.00120239  0.00120239  0.00120239  0.00120239  0.00120239
[4057]  0.00120239  0.00120117  0.00193561  0.00120239  0.00120239  0.00120239
[4063]  0.00120117  0.00193561  0.00120239  0.00142730  0.00120117  0.00120117
[4069]  0.00193561  0.00142730  0.00142730  0.00120239  0.00120117  0.00193561
[4075]  0.00142730  0.00120239  0.00142730  0.00120117  0.00193561  0.00120239
[4081]  0.00142730  0.00120239  0.00120239  0.00193561  0.00120239  0.00120239
[4087]  0.00120239  0.00120239  0.00216051  0.00142730  0.00120239  0.00120239
[4093]  0.00120239  0.00193561  0.00120239  0.00120239  0.00120239  0.00120239
[4099]  0.00193561  0.00142730  0.00120239  0.00120239  0.00120239  0.00193561
[4105]  0.00120117  0.00120239  0.00142730  0.00120239  0.00420170  0.00470491
[4111]  0.00420170  0.00420170 -0.00049399 -0.00022052 -0.00022052  0.00362493
[4117]  0.00362493  0.00441137  0.00420170 -0.00049399 -0.00022052 -0.00022052
[4123]  0.00441137  0.00240524  0.00441137  0.00441137 -0.00128081  0.00441137
[4129]  0.00420170  0.00335146 -0.00128081 -0.00128081  0.00420170  0.00335108
[4135]  0.00335146  0.00420170  0.00420170  0.00420170  0.00335108  0.00335108
[4141] -0.00143984 -0.00049399 -0.00049399  0.00420170  0.00526199 -0.00128081
[4147] -0.00143984 -0.00049399 -0.00049399 -0.00128081 -0.00049399  0.00362493
[4153]  0.00240562  0.00036157  0.00035642  0.00035642  0.00035642  0.00035642
[4159]  0.00036157  0.00035642  0.00035642  0.00035642  0.00027915  0.00036157
[4165]  0.00035642  0.00035642  0.00035642  0.00035642  0.00036157  0.00035642
[4171]  0.00035642  0.00035642  0.00035642  0.00035642  0.00036157  0.00035642
[4177]  0.00035642  0.00035642  0.00035642  0.00036157  0.00035642  0.00035642
[4183]  0.00035642  0.00036157  0.00035642  0.00035642  0.00035642  0.00035642
[4189]  0.00036157  0.00035642  0.00035642  0.00035642  0.00035642  0.00027915
[4195]  0.00035642  0.00036157  0.00035642  0.00035642  0.00035642  0.00035642
[4201]  0.00027915  0.00036157  0.00035642  0.00035642  0.00035642  0.00035642
[4207]  0.00036157  0.00035642  0.00035642  0.00035642  0.00035642  0.00027915
[4213]  0.00036157  0.00035642  0.00035642  0.00035642  0.00035642  0.00036157
[4219]  0.00035642  0.00035642  0.00035642  0.00035642  0.00042080  0.00054748
```

```
[4225]  0.00046203  0.00054748  0.00051950  0.00042080  0.00005646  0.00051289
[4231]  0.00051804  0.00051289  0.00051289  0.00005646  0.00051289  0.00051804
[4237]  0.00051289  0.00051289  0.00051289  0.00051289  0.00051804  0.00051289
[4243]  0.00051289  0.00051289  0.00051289  0.00005646  0.00051289  0.00051804
[4249]  0.00051289  0.00051289  0.00051289  0.00051289  0.00005646  0.00051289
[4255]  0.00017898  0.00017898  0.00017898  0.00017898  0.00005646  0.00017898
[4261]  0.00018412  0.00017898  0.00017898  0.00017898  0.00017898  0.00005646
[4267]  0.00017898  0.00018412  0.00017898  0.00017898  0.00017898  0.00017898
[4273]  0.00005646  0.00017898  0.00018412  0.00017898  0.00017898  0.00017898
[4279]  0.00017898  0.00005646  0.00017898  0.00018412  0.00017898  0.00017898
[4285]  0.00017898  0.00017898  0.00005646  0.00017898  0.00018412  0.00017898
[4291]  0.00017898  0.00017898  0.00005646  0.00017898  0.00018412  0.00017898
[4297]  0.00017898  0.00017898  0.00017898  0.00005646  0.00017898  0.00018412
[4303]  0.00017898  0.00017898  0.00017898  0.00017898  0.00005646  0.00017898
[4309]  0.00018412  0.00017898  0.00017898  0.00017898  0.00017898  0.00005646
[4315]  0.00057230  0.00045389  0.00045389  0.00021312  0.00057230  0.00045389
[4321]  0.00045389  0.00045389  0.00057230  0.00045389  0.00045389  0.00045389
[4327]  0.00045389  0.00057230  0.00045389  0.00045389  0.00045389  0.00045389
[4333]  0.00057230  0.00045389  0.00045389  0.00045389  0.00045389  0.00050643
[4339]  0.00045389  0.00021312  0.00045389  0.00057230  0.00045389  0.00045389
[4345]  0.00045389  0.00021312  0.00057230  0.00045389  0.00045389  0.00045389
[4351]  0.00045389  0.00057230  0.00045389  0.00045389  0.00045389  0.00045389
[4357]  0.00045389  0.00045389  0.00045389  0.00057230  0.00050643  0.00050643
[4363]  0.00057230  0.00050643  0.00045389  0.00045389  0.00021312  0.00045389
[4369]  0.00057230  0.00045389  0.00045389  0.00045389  0.00021312  0.00135410
[4375]  0.00372872  0.00372872  0.00132873  0.00132873  0.00102572  0.00102572
[4381]  0.00145279  0.00102572  0.00132873  0.00142742  0.00102572  0.00179096
[4387]  0.00142742  0.00179919  0.00102572  0.00102572  0.00102572  0.00102572
[4393]  0.00142742  0.00145279  0.00372872  0.00102572  0.00102572  0.00179096
[4399]  0.00132873  0.00102572  0.00102572  0.00102572  0.00375409  0.00102572
[4405]  0.00102572  0.00102572  0.00132873  0.00142742  0.00142742  0.00102572
[4411]  0.00102572  0.00179096  0.00102572  0.00132873  0.00102572  0.00145279
[4417]  0.00102572  0.00102572  0.00132873  0.00179096  0.00102572  0.00142742
[4423]  0.00142742  0.00145279  0.00102572  0.00132873  0.00084535  0.00013385
[4429]  0.00053056  0.00013385  0.00044864  0.00053056  0.00053056  0.00053056
[4435]  0.00053056  0.00044864  0.00053056  0.00053056  0.00053056  0.00013385
[4441]  0.00053056  0.00084535  0.00053056  0.00053056  0.00053056  0.00053056
[4447] -0.00004412  0.00084535  0.00053056  0.00053056  0.00013385  0.00053056
[4453]  0.00053056  0.00084535  0.00053056  0.00013385  0.00053056  0.00053056
[4459]  0.00084535  0.00053056  0.00013385  0.00053056  0.00013385  0.00084535
[4465]  0.00053056  0.00053056  0.00053056  0.00053056  0.00084535  0.00053056
[4471]  0.00053056  0.00053056  0.00053056  0.00084535  0.00053056  0.00053056
[4477]  0.00053056  0.00053056  0.00053056  0.00084535  0.00013385  0.00053056
[4483]  0.00053056  0.00013385  0.00053056  0.00084535  0.00053056  0.00053056
[4489]  0.00053056  0.00053056  0.00084535  0.00053056  0.00053056  0.00053056
[4495]  0.00053056  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813
[4501]  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813
[4507]  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813
[4513]  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813
[4519]  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813
[4525]  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813  0.00042813
[4531]  0.00042813  0.00047413  0.00044376  0.00044376  0.00047174  0.00047174
[4537]  0.00047174  0.00047174  0.00050211  0.00069665  0.00069665  0.00047174
[4543]  0.00047174  0.00047174  0.00044376  0.00047413  0.00044376  0.00044376
```

[4549]	0.00047174	0.00047174	0.00047174	0.00050211	0.00047174	0.00047174
[4555]	0.00044376	0.00050908	0.00060545	0.00047174	0.00038281	0.00044376
[4561]	0.00045286	0.00045286	0.00045286	0.00045286	0.00045286	0.00048323
[4567]	0.00045286	0.00045286	0.00045286	0.00045286	0.00045286	0.00045164
[4573]	0.00048323	0.00045286	0.00045286	0.00045286	0.00045286	0.00045286
[4579]	0.00045286	0.00048323	0.00045286	0.00045286	0.00045286	0.00045286
[4585]	0.00045286	0.00045286	0.00048323	0.00045286	0.00045286	0.00045286
[4591]	0.00045286	0.00045286	0.00039192	0.00045286	0.00045286	0.00045286
[4597]	0.00045286	0.00045286	0.00045286	0.00048323	0.00045286	0.00045286
[4603]	0.00067777	0.00045286	0.00045286	0.00048323	0.00045286	0.00045286
[4609]	0.00045286	0.00045286	0.00045286	0.00045286	0.00048323	0.00045286
[4615]	0.00045286	0.00045286	0.00045286	0.00045286	0.00048323	0.00051020
[4621]	0.00057191	0.00057191	0.00057191	0.00057191	0.00057191	0.00057191
[4627]	0.00051020	0.00057191	0.00057191	0.00057191	0.00057191	0.00057191
[4633]	0.00057191	0.00051020	0.00057191	0.00057191	0.00057191	0.00057191
[4639]	0.00057191	0.00057191	0.00051020	0.00057191	0.00057191	0.00057191
[4645]	0.00057191	0.00057191	0.00083039	0.00051020	0.00057191	0.00057191
[4651]	0.00057191	0.00057191	0.00057191	0.00057191	0.00051020	0.00057191
[4657]	0.00057191	0.00057191	0.00057191	0.00057191	0.00057191	0.00051020
[4663]	0.00057191	0.00057191	0.00057191	0.00057191	0.00057191	0.00057191
[4669]	0.00051020	0.00057191	0.00057191	0.00057191	0.00057191	0.00057191
[4675]	0.00057191	0.00051020	0.00057191	0.00057191	0.00057191	0.00057191
[4681]	0.00059079	0.00059079	0.00044029	0.00056281	0.00056281	0.00056281
[4687]	0.00059079	0.00059079	0.00286411	0.00032204	0.00056281	0.00056281
[4693]	0.00056281	0.00059079	0.00056281	0.00056281	0.00032204	0.00056281
[4699]	0.00056281	0.00056281	0.00059079	0.00046411	0.00059079	0.00035002
[4705]	0.00059079	0.00059079	0.00040322	0.00063254	0.00059079	0.00056281
[4711]	0.00033115	0.00027177	0.00042843	0.00042843	0.00270175	0.00027177
[4717]	0.00030176	0.00270175	0.00042843	0.00042843	0.00027177	0.00040045
[4723]	0.00040045	0.00042843	0.00042829	0.00027177	0.00040045	0.00042843
[4729]	0.00042843	0.00030176	0.00040031	0.00040045	0.00042843	0.00040045
[4735]	0.00042843	0.00040045	0.00042829	0.00042843	0.00042843	0.00042843
[4741]	0.00042829	0.00042843	0.00040045	0.00042843	0.00024087	0.00062899
[4747]	0.00042829	0.00034298	0.00042843	0.00065334	0.00042843	0.00042843
[4753]	0.00040045	0.00040045	0.00040045	0.00040045	0.00042829	0.00042843
[4759]	0.00270175	0.00270175	0.00042843	0.00040045	0.00042843	0.00040045
[4765]	0.00040045	0.00040045	0.00040045	0.00031500	0.00042843	0.00040045
[4771]	0.00186925	0.00186925	0.00052860	0.00052860	0.00052860	0.00044315
[4777]	0.00052860	0.00052860	0.00052860	0.00052860	0.00044315	0.00052860
[4783]	0.00052860	0.00052860	0.00044315	0.00052860	0.00052860	0.00052860
[4789]	0.00052860	0.00052860	0.00052860	0.00052860	0.00052860	0.00044315
[4795]	0.00052860	0.00052860	0.00052860	0.00052860	0.00044315	0.00052860
[4801]	0.00052860	0.00052860	0.00052860	0.00052860	0.00052860	0.00052860
[4807]	0.00052860	0.00044315	0.00052860	0.00052860	0.00052860	0.00052860
[4813]	0.00052860	0.00052860	0.00052860	0.00052860	0.00052860	0.00052860
[4819]	0.00052860	0.00044315	0.00052860	0.00052860	0.00052860	0.00052860
[4825]	0.00044315	0.00052860	0.00052860	0.00052860	0.00052860	0.00064517
[4831]	0.00064531	0.00065290	0.00064531	0.00064531	0.00045775	0.00064531
[4837]	0.00064517	0.00064531	0.00065290	0.00064531	0.00064531	0.00291863
[4843]	0.00064531	0.00064517	0.00087781	0.00064531	0.00061733	0.00064531
[4849]	0.00064531	0.00064517	0.00064531	0.00065290	0.00087021	0.00064409
[4855]	0.00064531	0.00045652	0.00059069	0.00081819	0.00045652	0.00045775
[4861]	0.00045775	0.00087021	0.00064517	0.00064531	0.00065290	0.00068265
[4867]	0.00064531	0.00064531	0.00064409	0.00064517	0.00064531	0.00065290

```
[4873]  0.00061733  0.00068265  0.00064531  0.00045652  0.00046534  0.00045775
[4879]  0.00068265  0.00064531  0.00064531  0.00045761  0.00045775  0.00046534
[4885]  0.00064531  0.00064531  0.00051863  0.00064531  0.00064517  0.00064531
[4891]  0.00065290  0.00064531  0.00087021  0.00065830  0.00068265  0.00064517
[4897]  0.00064531  0.00065290  0.00064531  0.00064531  0.00064531  0.00064531
[4903]  0.00064531  0.00065290  0.00064531  0.00064531  0.00064531  0.00045775
[4909]  0.00291849  0.00291863  0.00292622  0.00064531  0.00064531  0.00064531
[4915]  0.00045775  0.00009264  0.00012385  0.00012899  0.00012385  0.00012385
[4921]  0.00012385  0.00012385  0.00009264  0.00012385  0.00012899  0.00012385
[4927]  0.00012385  0.00012385  0.00012385  0.00009264  0.00012385  0.00012899
[4933]  0.00012385  0.00012385  0.00012385  0.00012385  0.00009264  0.00012385
[4939]  0.00012899  0.00012385  0.00012385  0.00012385  0.00012385  0.00009264
[4945]  0.00012385  0.00052771  0.00052256  0.00052256  0.00052256  0.00052256
[4951] -0.00042328  0.00052256  0.00052771  0.00052256  0.00052256  0.00052256
[4957]  0.00057510 -0.00042328  0.00052256  0.00052771  0.00052256  0.00057510
[4963]  0.00052256  0.00052256 -0.00042328  0.00052256  0.00052771  0.00052256
[4969]  0.00052256  0.00052256  0.00052256 -0.00042328  0.00057510  0.00080014
[4975]  0.00052256  0.00052256  0.00052256  0.00052256 -0.00042328  0.00052256
[4981]  0.00052771  0.00052256  0.00052256  0.00052256  0.00052256 -0.00042328
[4987]  0.00052256  0.00052771  0.00052256  0.00052256  0.00052256  0.00052256
[4993] -0.00042328  0.00052256  0.00052771  0.00052256  0.00052256  0.00052256
[4999]  0.00052256 -0.00042328  0.00052256  0.00052771  0.00052256  0.00052256
[5005]  0.00052256  0.00052256 -0.00042328  0.00130361  0.00131808  0.00131808
[5011]  0.00026260  0.00026260  0.00026260  0.00026260  0.00018534  0.00026260
[5017]  0.00026260  0.00026260  0.00026260  0.00026260  0.00018534  0.00026260
[5023]  0.00026260  0.00058388  0.00026260  0.00026260  0.00026260  0.00018534
[5029]  0.00026260  0.00026260  0.00026260  0.00026260  0.00019981  0.00032722
[5035]  0.00018534  0.00026260  0.00026260  0.00026260  0.00032722  0.00026260
[5041]  0.00058388  0.00018534  0.00026260  0.00026260 -0.00064762 -0.00064762
[5047]  0.00058388  0.00018534  0.00026260  0.00026260  0.00019981  0.00026260
[5053]  0.00026260  0.00026260  0.00018534  0.00026260  0.00026260  0.00026260
[5059]  0.00026260  0.00026260  0.00026260  0.00018534  0.00026260  0.00026260
[5065]  0.00026260  0.00026260  0.00026260  0.00026260  0.00018534  0.00026260
[5071]  0.00026260  0.00026260  0.00026260  0.00026260  0.00026260  0.00050661
[5077]  0.00058388  0.00058388  0.00058388  0.00058388  0.00026260  0.00050661
[5083]  0.00058388  0.00058388  0.00052108  0.00026260  0.00026260  0.00026260
[5089]  0.00018534  0.00026260  0.00026260  0.00026260  0.00058388  0.00058388
[5095]  0.00026260 -0.00048148 -0.00045028 -0.00044513 -0.00045028 -0.00045028
[5101] -0.00045028 -0.00045028 -0.00048148 -0.00045028 -0.00044513 -0.00039774
[5107] -0.00045028 -0.00039774 -0.00045028 -0.00048148 -0.00039774 -0.00039260
[5113] -0.00045028 -0.00039774 -0.00045028 -0.00045028 -0.00048148 -0.00045028
[5119] -0.00044513 -0.00045028 -0.00045028 -0.00045028 -0.00045028 -0.00048148
[5125] -0.00045028  0.00012899  0.00012385  0.00012385  0.00012385  0.00012385
[5131]  0.00009264  0.00012385  0.00012899  0.00012385  0.00026283  0.00012385
[5137]  0.00017638  0.00009264  0.00012385  0.00012899  0.00012385  0.00017638
[5143]  0.00012385  0.00012385  0.00009264  0.00012385  0.00012899  0.00017638
[5149]  0.00012385  0.00012385  0.00017638  0.00009264  0.00012385  0.00012899
[5155]  0.00012385  0.00012385  0.00012385  0.00012385  0.00009264  0.00012385
[5161]  0.00012899  0.00012385  0.00012385  0.00012385  0.00012385  0.00009264
[5167]  0.00012385  0.00012899  0.00012385  0.00012385  0.00012385  0.00017638
[5173]  0.00009264  0.00012385  0.00012899  0.00012385  0.00026283  0.00012385
[5179]  0.00026283  0.00009264  0.00012385  0.00012899  0.00012385  0.00012385
[5185]  0.00012385  0.00012385  0.00009264  0.00020921 -0.00013076 -0.00013076
[5191] -0.00013076  0.00009415  0.00009415  0.00013376 -0.00013076 -0.00013076
```

```
[5197] -0.00013076  0.00020921 -0.00013076 -0.00013076  0.00020921 -0.00013076
[5203]  0.00029300  0.00009415 -0.00013076 -0.00013076  0.00009415  0.00020921
[5209] -0.00013076 -0.00009115 -0.00013076 -0.00013076  0.00009415 -0.00001665
[5215]  0.00040711 -0.00013076 -0.00001665 -0.00028897 -0.00013076  0.00020921
[5221]  0.00009415  0.00051791 -0.00013076 -0.00001665 -0.00013076  0.00009415
[5227]  0.00020921 -0.00013076 -0.00009115 -0.00013076  0.00009415 -0.00013076
[5233]  0.00009415  0.00009415  0.00051791 -0.00013076  0.00009415 -0.00013076
[5239]  0.00009415  0.00020921  0.00009415  0.00051791  0.00009415 -0.00013076
[5245]  0.00009415 -0.00001665  0.00020921  0.00009415  0.00051791 -0.00013076
[5251] -0.00013076 -0.00013076 -0.00013076  0.00020921 -0.00013076 -0.00009115
[5257] -0.00013076 -0.00013076  0.00009415 -0.00013076  0.00020921 -0.00013076
[5263] -0.00009115  0.00009415 -0.00013076 -0.00013076 -0.00013076  0.00020921
[5269]  0.00038014  0.00044186  0.00044186  0.00044186  0.00044186  0.00044186
[5275]  0.00025429  0.00038014  0.00044186  0.00044186  0.00044186  0.00044186
[5281]  0.00044186  0.00044186  0.00038014  0.00044186  0.00044186  0.00044186
[5287]  0.00044186  0.00044186  0.00044186  0.00035216  0.00041387  0.00041387
[5293]  0.00044186 -0.00074789  0.00031891  0.00019150  0.00038014  0.00044186
[5299]  0.00044186  0.00044186  0.00044186  0.00025429  0.00025429  0.00019258
[5305]  0.00025429  0.00025429  0.00044186  0.00044186  0.00025429  0.00044186
[5311]  0.00019258  0.00025429  0.00025429  0.00044186  0.00037906  0.00044186
[5317]  0.00044186  0.00265346  0.00271517  0.00271517  0.00044186  0.00044186
[5323]  0.00271517  0.00019150  0.00019258  0.00025429  0.00025429  0.00044186
[5329]  0.00044186  0.00044186  0.00025429  0.00038014  0.00044186  0.00044186
[5335]  0.00044186  0.00044186  0.00044186  0.00044186  0.00019258  0.00025429
[5341]  0.00025429  0.00025429  0.00025429  0.00044186  0.00044186  0.00019258
[5347]  0.00025429  0.00025429  0.00044186  0.00041387  0.00044186  0.00041387
[5353]  0.00038014  0.00044186  0.00044186 -0.00065079  0.00044186  0.00044186
[5359]  0.00044186  0.00035216  0.00006537  0.00006429  0.00007188  0.00006551
[5365]  0.00006551  0.00006551  0.00006551  0.00006537  0.00029042  0.00051791
[5371]  0.00006551  0.00006551  0.00006551  0.00006551  0.00006537  0.00006551
[5377]  0.00007310  0.00006551  0.00006551  0.00029042  0.00006429  0.00006537
[5383]  0.00029042  0.00051791  0.00006551  0.00006429  0.00029042  0.00006551
[5389]  0.00006537  0.00029042  0.00051791  0.00026607  0.00006551  0.00006429
[5395]  0.00006551  0.00006537  0.00006551  0.00007310  0.00006551  0.00006551
[5401]  0.00006551  0.00006429  0.00006537  0.00006551  0.00007310  0.00006551
[5407]  0.00006429  0.00006551  0.00006429  0.00006537  0.00006551  0.00007310
[5413]  0.00006429  0.00006551  0.00006551  0.00006429  0.00006551  0.00007310
[5419]  0.00006551  0.00029042  0.00006551  0.00006551  0.00006537  0.00006551
[5425]  0.00007310  0.00006551  0.00006551  0.00006551  0.00029042  0.00007310
[5431]  0.00006551  0.00029042  0.00006429  0.00006429  0.00006537  0.00006429
[5437]  0.00029178  0.00006551  0.00006551  0.00006429  0.00006551  0.00006537
[5443]  0.00006429  0.00029178  0.00006551  0.00029042  0.00029042  0.00006429
[5449]  0.00006537 -0.00004170  0.00011740  0.00011740  0.00011740 -0.00004170
[5455]  0.00011740  0.00011740  0.00011740  0.00011740 -0.00004170  0.00011740
[5461]  0.00011740 -0.00004170  0.00011740  0.00011740 -0.00004170  0.00011740
[5467]  0.00011740 -0.00004170  0.00011740  0.00011740 -0.00004170  0.00011740
[5473]  0.00011740 -0.00004170  0.00011740  0.00011740 -0.00004170  0.00011740
[5479]  0.00011740 -0.00004170  0.00011740  0.00011740  0.00011740  0.00011740
[5485] -0.00004170  0.00011740  0.00011740  0.00011740  0.00011740 -0.00004170
[5491]  0.00011740  0.00011740  0.00011740  0.00011740 -0.00004170  0.00011740
[5497]  0.00011740  0.00011740  0.00011740  0.00080259  0.00086430  0.00086430
[5503]  0.00048214  0.00048214  0.00086430  0.00048214  0.00080259  0.00086430
[5509]  0.00086430  0.00086430  0.00086430  0.00086430  0.00086430  0.00042043
[5515]  0.00048214  0.00048214  0.00086430  0.00086430  0.00086430  0.00048214
```

```
[5521]  0.00080259  0.00086430  0.00086430  0.00086430  0.00048214  0.00086430
[5527]  0.00048214  0.00080259  0.00086430  0.00086430  0.00048214  0.00086430
[5533]  0.00048214  0.00086430  0.00074178  0.00086430  0.00086430  0.00086430
[5539]  0.00086430  0.00048214  0.00154562  0.00042043  0.00154243  0.00154243
[5545]  0.00086430  0.00086430  0.00086430  0.00048214  0.00042043  0.00048214
[5551]  0.00048214  0.00086430  0.00086430  0.00086430  0.00048214  0.00042043
[5557]  0.00048214  0.00048214  0.00048214  0.00086430  0.00086430  0.00086430
[5563]  0.00074178  0.00086430  0.00086430  0.00086430  0.00086430  0.00048214
[5569]  0.00086430  0.00080259  0.00086430  0.00086430  0.00086430  0.00086430
[5575]  0.00048214  0.00086430  0.00080259  0.00086430  0.00086430  0.00086430
[5581]  0.00086430  0.00048214  0.00086430  0.00080259  0.00086430  0.00086430
[5587]  0.00048214  0.00048214  0.00086430  0.00086430  0.00080259  0.00071320
[5593]  0.00071320  0.00071320  0.00071320  0.00006764  0.00071320  0.00071320
[5599]  0.00071320  0.00006764  0.00071320  0.00071320  0.00071320  0.00071320
[5605]  0.00071320  0.00071320  0.00071320 -0.00004170  0.00011496  0.00011496
[5611]  0.00011496  0.00011496 -0.00004170  0.00011496  0.00011496  0.00011496
[5617]  0.00011496 -0.00004170  0.00011496  0.00011496  0.00011496  0.00011496
[5623] -0.00004170  0.00011496  0.00011496  0.00011496  0.00011496 -0.00009892
[5629]  0.00005774  0.00005774  0.00005774  0.00005774 -0.00009892  0.00005774
[5635]  0.00005774  0.00005774  0.00005774 -0.00009892  0.00005774  0.00005774
[5641]  0.00005774  0.00005774 -0.00009892  0.00005774  0.00005774  0.00005774
[5647]  0.00005774 -0.00009892  0.00005774  0.00005774  0.00005774  0.00005774
[5653]  0.00005774  0.00005774  0.00005774  0.00005774  0.00005774 -0.00009892
[5659]  0.00005774  0.00005774  0.00005774  0.00005774 -0.00009892  0.00005774
[5665]  0.00005774  0.00005774  0.00005774  0.00005774 -0.00009892  0.00005774
[5671]  0.00005774  0.00005774  0.00005774 -0.00017251 -0.00011156 -0.00011156
[5677]  0.00024051 -0.00011156  0.00024051  0.00024051 -0.00017251 -0.00011156
[5683] -0.00011156  0.00024051  0.00024051  0.00024051  0.00024051 -0.00017251
[5689] -0.00011156  0.00024051 -0.00011156 -0.00011156  0.00024051  0.00024051
[5695] -0.00017251 -0.00011156 -0.00011156 -0.00011156  0.00024051 -0.00011156
[5701] -0.00011156  0.00027088  0.00024051  0.00024051  0.00024051  0.00024051
[5707] -0.00011156 -0.00011156  0.00017957  0.00024051 -0.00011156 -0.00011156
[5713] -0.00004038 -0.00011156  0.00024051  0.00027088  0.00024051  0.00024051
[5719]  0.00024051  0.00024051  0.00024051  0.00043916  0.00027088  0.00024051
[5725]  0.00024051  0.00024051  0.00024051 -0.00011156 -0.00011156  0.00027088
[5731]  0.00024051  0.00024051  0.00024051 -0.00011156 -0.00011156  0.00024051
[5737] -0.00017251 -0.00011156 -0.00011156 -0.00004038 -0.00011156 -0.00011156
[5743]  0.00024051 -0.00017251 -0.00011156 -0.00011156 -0.00011156 -0.00011156
[5749]  0.00024051  0.00024051  0.00017957  0.00024051  0.00024051 -0.00011156
[5755] -0.00011156  0.00024051 -0.00011156 -0.00017251 -0.00011156 -0.00011156
[5761] -0.00011156  0.00046542  0.00024051  0.00024051 -0.00017251  0.00015827
[5767]  0.00015827  0.00015827  0.00009732  0.00015827  0.00015827  0.00015827
[5773]  0.00015827  0.00015827  0.00015827  0.00009732  0.00015827  0.00015827
[5779]  0.00015827  0.00015827  0.00015827  0.00015827  0.00009732  0.00015827
[5785]  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827  0.00009732
[5791]  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827
[5797]  0.00009732  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827
[5803]  0.00015827  0.00009732  0.00015827  0.00015827  0.00015827  0.00015827
[5809]  0.00015827  0.00015827  0.00009732  0.00015827  0.00015827  0.00015827
[5815]  0.00015827  0.00015827  0.00015827  0.00009732  0.00015827  0.00015827
[5821]  0.00015827  0.00015827  0.00015827  0.00015827  0.00009732  0.00015827
[5827]  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827  0.00009732
[5833]  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827
[5839]  0.00009732  0.00015827  0.00015827  0.00015827  0.00015827  0.00015827
```

```
[5845]  0.00015827  0.00009732  0.00015827  0.00015827  0.00015827  0.00015827
[5851]  0.00015827  0.00015827  0.00009732 -0.00025001 -0.00043757 -0.00025001
[5857] -0.00025001 -0.00043757 -0.00043757 -0.00025001 -0.00025001  0.00094141
[5863]  0.00046072  0.00036203 -0.00025001  0.00139388 -0.00025001 -0.00025001
[5869]  0.00048870 -0.00025001  0.00126269  0.00106882 -0.00043757 -0.00025001
[5875]  0.00036203 -0.00011630  0.00036203 -0.00025001  0.00106882 -0.00025001
[5881] -0.00043757 -0.00025001 -0.00025001 -0.00025001 -0.00025001  0.00048870
[5887]  0.00094141 -0.00025001 -0.00025001  0.00048870 -0.00025001 -0.00025001
[5893] -0.00025001 -0.00025001 -0.00025001 -0.00025001 -0.00025001 -0.00025001
[5899] -0.00025001 -0.00025001 -0.00025001 -0.00025001 -0.00043757 -0.00043757
[5905] -0.00025001 -0.00025001  0.00006782  0.00012953  0.00034018  0.00012953
[5911]  0.00012953  0.00018207  0.00018207  0.00006782  0.00012953  0.00034018
[5917]  0.00012953  0.00018207  0.00012953  0.00012953  0.00006782  0.00012953
[5923]  0.00034018  0.00012953  0.00012953  0.00012953  0.00012953  0.00006782
[5929]  0.00018207  0.00039272  0.00012953  0.00012953  0.00012953  0.00012953
[5935]  0.00006782  0.00012953  0.00034018  0.00012953  0.00012953  0.00012953
[5941]  0.00012953  0.00006782  0.00012953  0.00034018  0.00012953  0.00018207
[5947]  0.00012953  0.00012953  0.00006782  0.00012953  0.00034018  0.00012953
[5953]  0.00018207  0.00012953  0.00018207  0.00006782  0.00012953  0.00034018
[5959]  0.00012953  0.00012953  0.00012953  0.00012953  0.00006782  0.00012953
[5965]  0.00034018  0.00012953  0.00012953  0.00012953  0.00012953  0.00006782
[5971]  0.00012953  0.00034018  0.00012953  0.00012953  0.00012953  0.00018207
[5977]  0.00006782  0.00012953  0.00034018  0.00012953  0.00012953  0.00012953
[5983]  0.00012953  0.00006782  0.00012953  0.00034018  0.00012953  0.00012953
[5989]  0.00012953  0.00012953  0.00006782  0.00012953  0.00034018  0.00012953
[5995]  0.00012953  0.00012953  0.00012953  0.00006782  0.00525170  0.00525170
[6001]  0.00525170  0.00525170  0.00517626  0.00525170  0.00525170  0.00525170
[6007]  0.00525170  0.00048675  0.00014677  0.00026520  0.00014677  0.00014677
[6013]  0.00017475  0.00017475  0.00051473  0.00017475  0.00029318  0.00014677
[6019]  0.00004808  0.00244807  0.00017475  0.00008583  0.00014677  0.00026520
[6025]  0.00014677  0.00017475  0.00004808  0.00014677  0.00278805  0.00244807
[6031]  0.00256650  0.00014677  0.00004808  0.00017475  0.00017475  0.00038805
[6037]  0.00004808  0.00016651  0.00017475  0.00244807  0.00017475  0.00014677
[6043] -0.00001403  0.00048855 -0.00001281  0.00017475  0.00017475  0.00014677
[6049]  0.00051473  0.00029318  0.00017475  0.00017475  0.00004808  0.00014677
[6055]  0.00016651  0.00017475  0.00039966  0.00014677  0.00014677  0.00017475
[6061]  0.00029318  0.00017475  0.00014677  0.00014677  0.00014677  0.00014677
[6067]  0.00014677  0.00039966  0.00017475  0.00004808  0.00029318  0.00017475
[6073]  0.00004808  0.00017475  0.00017475  0.00051473  0.00017475  0.00029318
[6079]  0.00004808  0.00014677  0.00017475  0.00017475  0.00048675  0.00014677
[6085]  0.00026520  0.00017475  0.00017475  0.00017475  0.00004808  0.00386426
```

```
suppressWarnings({
while (any(is.na(Test26$followers))) {
  Test26 <- Test26 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+pred.xgb.tuned26), follower
}
while (any(is.na(Test27$followers))) {
  Test27 <- Test27 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+pred.xgb.tuned27), follower
}
while (any(is.na(Test28$followers))) {
  Test28 <- Test28 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+pred.xgb.tuned28), follower
```

```
}
while (any(is.na(Test29$followers))) {
  Test29 <- Test29 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+pred.xgb.tuned29), follower
}
while (any(is.na(Test30$followers))) {
  Test30 <- Test30 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+pred.xgb.tuned30), follower
}
})
```

```
pacman::p_load(tidyverse)
# merge the six companies with other companies
Test_submission26 <- Test26 %>% select(gvkey, date, followers, ID)
Test_submission26 <- rbind(Test_submission26, Test_6)
Test_submission26 <- Test_submission26 %>% filter(date>=20170701)
Test_submission26$followers <- round(Test_submission26$followers)
Followers_submisson_M26 <- Test_submission26 %>% select(ID, followers)

Test_submission27 <- Test27 %>% select(gvkey, date, followers, ID)
Test_submission27 <- rbind(Test_submission27, Test_6)
Test_submission27 <- Test_submission27 %>% filter(date>=20170701)
Test_submission27$followers <- round(Test_submission27$followers)
Followers_submisson_M27 <- Test_submission27 %>% select(ID, followers)

Test_submission28 <- Test28 %>% select(gvkey, date, followers, ID)
Test_submission28 <- rbind(Test_submission28, Test_6)
Test_submission28 <- Test_submission28 %>% filter(date>=20170701)
Test_submission28$followers <- round(Test_submission28$followers)
Followers_submisson_M28 <- Test_submission28 %>% select(ID, followers)

Test_submission29 <- Test29 %>% select(gvkey, date, followers, ID)
Test_submission29 <- rbind(Test_submission29, Test_6)
Test_submission29 <- Test_submission29 %>% filter(date>=20170701)
Test_submission29$followers <- round(Test_submission29$followers)
Followers_submisson_M29 <- Test_submission29 %>% select(ID, followers)

Test_submission30 <- Test30 %>% select(gvkey, date, followers, ID)
Test_submission30 <- rbind(Test_submission30, Test_6)
Test_submission30 <- Test_submission30 %>% filter(date>=20170701)
Test_submission30$followers <- round(Test_submission30$followers)
Followers_submisson_M30 <- Test_submission30 %>% select(ID, followers)

# save the sample of submission
write.csv(Followers_submisson_M26, "submission_XGM26.csv", row.names = FALSE)
write.csv(Followers_submisson_M27, "submission_XGM27.csv", row.names = FALSE)
write.csv(Followers_submisson_M28, "submission_XGM28.csv", row.names = FALSE)
write.csv(Followers_submisson_M29, "submission_XGM29.csv", row.names = FALSE)
write.csv(Followers_submisson_M30, "submission_XGM30.csv", row.names = FALSE)
```

9.6 Out of Sample Evaluation

It is worthy to note that XGBoost models outperform our domain knowledge models in terms of out of sample prediction.

✓	**submission_XGM30.csv** Complete · now	**3192.68171**
✓	**submission_XGM29.csv** Complete · 16s ago	**2463.9722**
✓	**submission_XGM28.csv** Complete · 27s ago	**1909.79259**
✓	**submission_XGM27.csv** Complete · 39s ago	**3021.97579**
✓	**submission_XGM26.csv** Complete · 1m ago	**2688.35717**

10. Random Forest

10.1 Hyperparameter Tuning

Random Forest is a powerful ensemble learning technique widely used for regression and classification tasks. The technique involves the building of decision trees on bootstrap dataset and running predictions on each tree before aggregating (or bagging) the results. This mitigate the overfitting issues faced by decision trees, thus reducing variance.

Hyperparameter tuning is crucial for optimizing the performance of Random Forest models. various combinations of hyperparameters are tested to find the optimal configuration that yields the best model performance.

We performed hyperparameter tuning for Random Forest using the tuneRF function from the randomForest package in R. Below is the brief explanation of the steps performed:

1. set.seed(2021) for reproducibility of results.
2. randomForest::tuneRF : Executes grid search with cross-validation using the tuneRF function, which
   automatically selects the optimal values for hyperparameters based on out-of-bag (OOB) error estimates.
3. Extract Optimal mtry Value: Identifies the optimal value for the mtry hyperparameter, which represents the number of variables randomly sampled as candidates at each split.
4. Run Random Forest Models with Optimal mtry: Train Random Forest models (rf_model31, rf_model32, ..., rf_model35) using the optimal mtry value obtained from the tuning process for different formulas (ML_Mod21_formula, ML_Mod22_formula, ..., ML_Mod25_formula). kindly refer to section 8.1 for the 5 models selected for running on advanced ML algorthims

```
#---------Random Forest Hyperparameter Tuning and Ensembling Method ---------------------#
set.seed(2021)
#define hyperparameter grid for tuning
```

```r
hyperparameters <- list(
  n_estimators = c(50, 100, 150),       #no of trees in the forest
  max_depth = c(5, 10, 15),             #max depth of trees
  min_samples_split = c(2, 5, 10),      #min samples required to split a node
  min_samples_leaf = c(1, 2, 4),        #min samples required at each leaf node
  max_features = c("sqrt", "log2"),     #no of features to consider at each split
  bootstrap = c(TRUE, FALSE)            #whether to use bootstrap samples
)

#identify categorical variables
categorical_vars <- sapply(Train, is.factor)
print (categorical_vars)
```

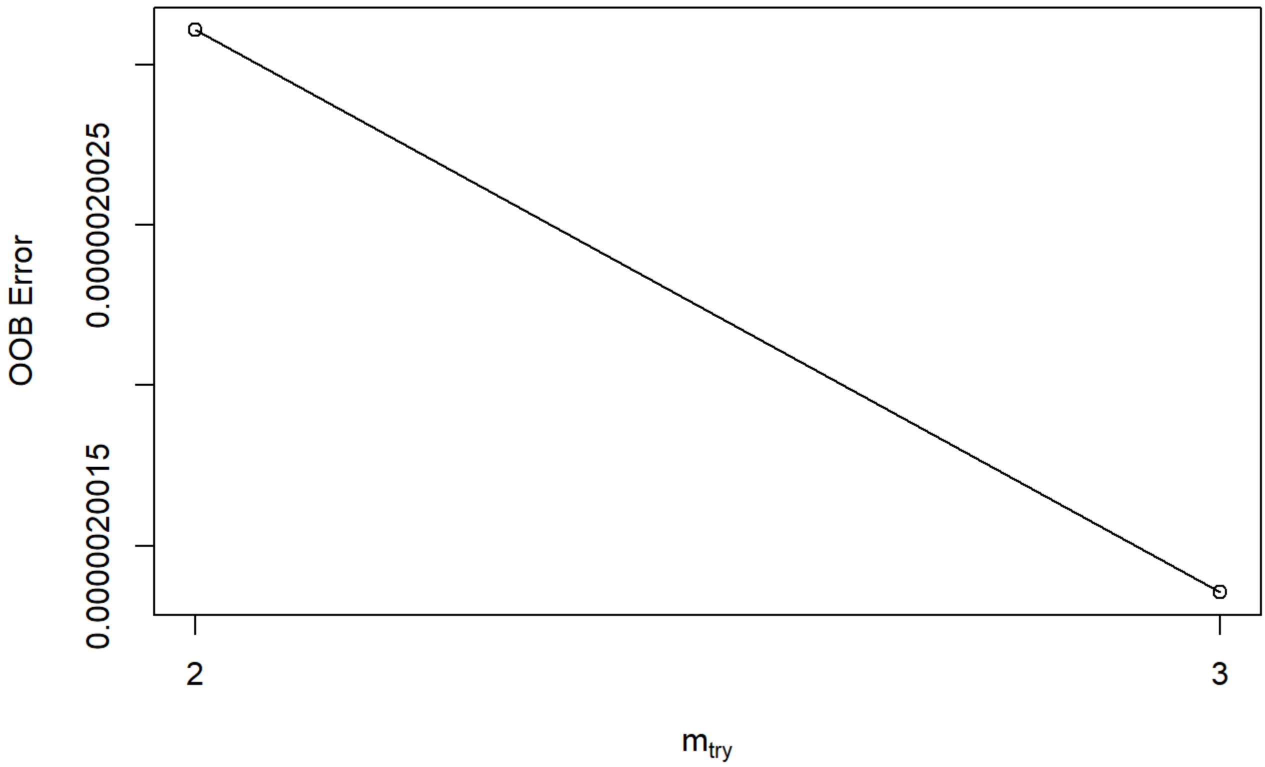
```
                 date                 gvkey                   atq 
                FALSE                 FALSE                 FALSE 
       p_daily_change              volatile    p_daily_change_lag 
                FALSE                 FALSE                 FALSE 
         volatile_lag                    bm                pe_exi 
                FALSE                 FALSE                 FALSE 
                   ps                   pcf                   dpr 
                FALSE                 FALSE                 FALSE 
                  npm                opmbd                 opmad 
                FALSE                 FALSE                 FALSE 
                  gpm                   roa                   roe 
                FALSE                 FALSE                 FALSE 
                 roce               debt_at              de_ratio 
                FALSE                 FALSE                 FALSE 
              intcov            cash_ratio           quick_ratio 
                FALSE                 FALSE                 FALSE 
           curr_ratio              inv_turn               at_turn 
                FALSE                 FALSE                 FALSE 
          debt_assets                   ptb             rect_turn 
                FALSE                 FALSE                 FALSE 
             sale_nwc              divyield                bm_lag 
                FALSE                 FALSE                 FALSE 
           pe_exi_lag                ps_lag               pcf_lag 
                FALSE                 FALSE                 FALSE 
              dpr_lag               npm_lag             opmbd_lag 
                FALSE                 FALSE                 FALSE 
            opmad_lag               gpm_lag               roa_lag 
                FALSE                 FALSE                 FALSE 
              roe_lag              roce_lag           debt_at_lag 
                FALSE                 FALSE                 FALSE 
          de_ratio_lag            intcov_lag        cash_ratio_lag 
                FALSE                 FALSE                 FALSE 
       quick_ratio_lag        curr_ratio_lag          inv_turn_lag 
                FALSE                 FALSE                 FALSE 
           at_turn_lag       debt_assets_lag               ptb_lag 
                FALSE                 FALSE                 FALSE 
         rect_turn_lag          sale_nwc_lag          divyield_lag 
                FALSE                 FALSE                 FALSE 
            is_annouce           red_annouce       no_annoucement 
                FALSE                 FALSE                 FALSE 
                  TFF          coleman_liau               flesch 
```

FALSE	FALSE	FALSE
fog_info	num_words	bio_complete
FALSE	FALSE	FALSE
has_url	hashtags_count	has_hashtags
FALSE	FALSE	FALSE
contains_company_name	contains_words	custom_content
FALSE	FALSE	FALSE
days_joined	years_joined	months_joined
FALSE	FALSE	FALSE
pop_ind	is_mon	is_tue
FALSE	FALSE	FALSE
is_wed	is_thu	is_fri
FALSE	FALSE	FALSE
is_sat	followers	ID
FALSE	FALSE	FALSE
sic	followers_lag	growth_rate
FALSE	FALSE	FALSE
Pre_rate_1	Pre_rate_2	Pre_rate_3
FALSE	FALSE	FALSE
Pre_rate_4	Pre_rate_5	Pre_rate_6
FALSE	FALSE	FALSE
Pre_rate_7	Pre_rate_8	Pre_rate_9
FALSE	FALSE	FALSE
Pre_rate_10	Pre_rate_11	Pre_rate_12
FALSE	FALSE	FALSE
Followers_M1	Followers_M2	Followers_M3
FALSE	FALSE	FALSE
Followers_M4	Followers_M5	Followers_M6
FALSE	FALSE	FALSE
Followers_M7	Followers_M8	Followers_M9
FALSE	FALSE	FALSE
Followers_M10	Followers_M11	Followers_M12
FALSE	FALSE	FALSE
Pre_rate_13	Pre_rate_14	Pre_rate_15
FALSE	FALSE	FALSE
Followers_M13	Followers_M14	Followers_M15
FALSE	FALSE	FALSE
Pre_rate_16	Pre_rate_17	Pre_rate_18
FALSE	FALSE	FALSE
Pre_rate_19	Pre_rate_20	Followers_M16
FALSE	FALSE	FALSE
Followers_M17	Followers_M18	Followers_M19
FALSE	FALSE	FALSE
Followers_M20	log_atq	factor_gvkey
FALSE	FALSE	TRUE
factor_sic	LASSO_rate_M21	LASSO_rate_M22
TRUE	FALSE	FALSE
LASSO_rate_M23	LASSO_rate_M24	LASSO_rate_M25
FALSE	FALSE	FALSE
LASSO_rate_SE21	LASSO_rate_SE22	LASSO_rate_SE23
FALSE	FALSE	FALSE
LASSO_rate_SE24	LASSO_rate_SE25	Follower_LA_M21
FALSE	FALSE	FALSE
Follower_LA_M22	Follower_LA_M23	Follower_LA_M24

```
               FALSE                FALSE                FALSE
        Follower_LA_M25     Follower_LA_SE21     Follower_LA_SE22
               FALSE                FALSE                FALSE
       Follower_LA_SE23     Follower_LA_SE24     Follower_LA_SE25
               FALSE                FALSE                FALSE
           XG_rate_26           XG_rate_27           XG_rate_28
               FALSE                FALSE                FALSE
           XG_rate_29           XG_rate_30      Followers_XG_26
               FALSE                FALSE                FALSE
        Followers_XG_27      Followers_XG_28      Followers_XG_29
               FALSE                FALSE                FALSE
        Followers_XG_30
               FALSE
```

```
#exclude categorical variables from the dataset
#randomforest cannot handle categorical predictors with >53 categories
Train_numeric <- Train[, !categorical_vars]

#perform grid search with cross-validation
rf_grid <- randomForest::tuneRF(x = Train_numeric[, -which(names(Train_numeric) == "growth_rat
                                y = Train_numeric$growth_rate,
                                mtryStart = 2,
                                ntreeTry = 100,
                                stepFactor = 1.5,
                                improve = 0.05,
                                trace = TRUE,
                                plot = TRUE)
```

```
mtry = 2  OOB error = 0.000002003
Searching left ...
Searching right ...
mtry = 3    OOB error = 0.000002001
0.0008748 0.05
```




```
#extract the optimal mtry value
optimal_mtry <- rf_grid[which.min(rf_grid)]

#finally, rerun the randomForest() function with the updated mtry parameter
rf_model31 <- randomForest(ML_Mod21_formula,
                           data = Train,
                           mtry = optimal_mtry)
```

```
Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
range
```

```
rf_model32 <- randomForest(ML_Mod22_formula,
                           data = Train,
                           mtry = optimal_mtry)
```

```
Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
range
```

```
rf_model33 <- randomForest(ML_Mod23_formula,
                           data = Train,
                           mtry = optimal_mtry)
```

```
Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
range
```

```r
rf_model34 <- randomForest(ML_Mod24_formula,
                          data = Train,
                          mtry = optimal_mtry)
```

```
Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
range
```

```r
rf_model35 <- randomForest(ML_Mod25_formula,
                          data = Train,
                          mtry = optimal_mtry)
```

```
Warning in randomForest.default(m, y, ...): invalid mtry: reset to within valid
range
```

10.2 In Sample Evaluation

Next, we use the trained Random forest model to make prediction on Twitter follower growth rate on Train data. The Twitter follower growth rate is then use to derive the predicted daily Twitter followers count.

This allows us to perform evaluation on the prediction capabilities of these Random forest model in sample (i.e. on Train data). RF Model 31 (Financial Model) makes the best in sample prediction with the lowest RMSE of 748.0 while RF Model 33 (Non Financial Model) performs the worst in terms of in sample prediction with the highest RMSE of 927.0.

```r
#make predictions on growth_rate on Train data
Train$RTpred_rate31 <- predict(rf_model31, Train)
Train$RTpred_rate32 <- predict(rf_model32, Train)
Train$RTpred_rate33 <- predict(rf_model33, Train)
Train$RTpred_rate34 <- predict(rf_model34, Train)
Train$RTpred_rate35 <- predict(rf_model35, Train)

#create rmse function to measure in sample accuracy
rmse <- function(v1, v2) {
  sqrt(mean((v1 - v2)^2, na.rm = T))
}

#compute predicted followers using predicted growth rate obtained from random Tree forest
Train$RTfollowers31 <- (1+Train$RTpred_rate31)*Train$followers_lag
Train$RTfollowers32 <- (1+Train$RTpred_rate32)*Train$followers_lag
Train$RTfollowers33 <- (1+Train$RTpred_rate33)*Train$followers_lag
Train$RTfollowers34 <- (1+Train$RTpred_rate34)*Train$followers_lag
Train$RTfollowers35 <- (1+Train$RTpred_rate35)*Train$followers_lag

#evaluate the in sample performance (rmse) of the model
RMSE <- c(rmse(Train$followers,Train$RTfollowers31),
          rmse(Train$followers,Train$RTfollowers32),
          rmse(Train$followers,Train$RTfollowers33),
          rmse(Train$followers,Train$RTfollowers34),
          rmse(Train$followers,Train$RTfollowers35))

## To get the RMSE for all the models
```

```r
names(RMSE) <- c("RF31", "RF32", "RF33", "RF34", "RF35")
RMSE
```

```
 RF31  RF32  RF33  RF34  RF35
749.1 786.6 921.8 779.4 814.7
```

10.3 Prediction of Test Followers

To ensure that our Random Forest Model is accurate in prediction, we went on to run our Random Forest model on Test data to make out of sample prediction. Based on the out of sample prediction, RF Model 31 (Financial Model) performs the best followed by RF Model 34 (Combined Model).

Random Forest succeeded in enhancing our domain knowledge model prediction capabilities. Random Forest ensembling techniques on random sampling and creation of random subsets of features and aggregation of each parallel trees is effective in increasing the prediction capabilities of model on smaller dataset. However, the boostraping and bagging techniques in Random Forest is not as powerful as the boosting techniques used in XGBoost on enhancing prediction accuracy of our model.

```r
#----------------- out of sample prediction---------------------------------------#
library(dplyr)

#make predictions on growth_rate on Test data
Test31 <- Test
Test31$RTpred_rate31 <- predict(rf_model31, Test31)
Test32 <- Test
Test32$RTpred_rate32 <- predict(rf_model32, Test32)
Test33 <- Test
Test33$RTpred_rate33 <- predict(rf_model33, Test33)
Test34 <- Test
Test34$RTpred_rate34 <- predict(rf_model34, Test34)
Test35 <- Test
Test35$RTpred_rate35 <- predict(rf_model35, Test35)


#merge the last day of followers in Train data to Test data
Test31 <- full_join(last_day_followers, Test31, by = join_by(gvkey, date))
Test31 <- Test31 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test32 <- full_join(last_day_followers, Test32, by = join_by(gvkey, date))
Test32 <- Test32 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test33 <- full_join(last_day_followers, Test33, by = join_by(gvkey, date))
```

```r
Test33 <- Test33 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test34 <- full_join(last_day_followers, Test34, by = join_by(gvkey, date))
Test34 <- Test34 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test35 <- full_join(last_day_followers, Test35, by = join_by(gvkey, date))
Test35 <- Test35 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

while (any(is.na(Test31$followers))) {
  Test31 <- Test31 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test31$RTpred_rate31 ), fo
}
while (any(is.na(Test32$followers))) {
  Test32 <- Test32 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test32$RTpred_rate32 ), fo
}
while (any(is.na(Test33$followers))) {
  Test33 <- Test33 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test33$RTpred_rate33 ), fo
}
while (any(is.na(Test34$followers))) {
  Test34 <- Test34 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test34$RTpred_rate34 ), fo
}
while (any(is.na(Test35$followers))) {
  Test35 <- Test35 %>%
    mutate(followers = ifelse(is.na(followers), lag(followers) * (1+Test35$RTpred_rate35 ), fo
}

#-----------------------generate Kaggle submission file-----------------------------#

# merge the six companies with other companies 6492
Test31 <- Test31 %>% select(gvkey, date, followers, ID)
Test31 <- rbind(Test31, Test_6)
Test31 <- Test31 %>% filter(date>=20170701)
Test31$followers <- round(Test31$followers)
Followers_submisson_RF31 <- Test31 %>% select(ID, followers)

Test32 <- Test32 %>% select(gvkey, date, followers, ID)
Test32 <- rbind(Test32, Test_6)
Test32<- Test32 %>% filter(date>=20170701)
Test32$followers <- round(Test32$followers)
```

```
Followers_submisson_RF32 <- Test32 %>% select(ID, followers)

Test33 <- Test33 %>% select(gvkey, date, followers, ID)
Test33 <- rbind(Test33, Test_6)
Test33 <- Test33 %>% filter(date>=20170701)
Test33$followers <- round(Test33$followers)
Followers_submisson_RF33 <- Test33 %>% select(ID, followers)

Test34 <- Test34 %>% select(gvkey, date, followers, ID)
Test34 <- rbind(Test34, Test_6)
Test34 <- Test34 %>% filter(date>=20170701)
Test34$followers <- round(Test34$followers)
Followers_submisson_RF34 <- Test34 %>% select(ID, followers)

Test35 <- Test35 %>% select(gvkey, date, followers, ID)
Test35 <- rbind(Test35, Test_6)
Test35 <- Test35 %>% filter(date>=20170701)
Test35$followers <- round(Test35$followers)
Followers_submisson_RF35 <- Test35 %>% select(ID, followers)

# save the sample of submission
write.csv(Followers_submisson_RF31, "submission_RF31.csv", row.names = FALSE)
write.csv(Followers_submisson_RF32, "submission_RF32.csv", row.names = FALSE)
write.csv(Followers_submisson_RF33, "submission_RF33.csv", row.names = FALSE)
write.csv(Followers_submisson_RF34, "submission_RF34.csv", row.names = FALSE)
write.csv(Followers_submisson_RF35, "submission_RF35.csv", row.names = FALSE)
```

10.4 Out of Sample Evaluation

submission_RF35.csv Complete · now	**3395.81836**
submission_RF34.csv Complete · 13s ago	**2546.87537**
submission_RF33.csv Complete · 29s ago	**3663.6778**
submission_RF32.csv Complete · 44s ago	**3435.91113**
submission_RF31.csv Complete · 1m ago	**2137.71997**

11. Ensemble

After using LASSO, XGboost and Random Forest ensembling techniques, we proceed to use simple Ensemble methods (Ensemble by average and Ensemble by weighted average) to further improve our predictionn.. We choose financial model 3 and non-financial model 1 to try to obtain a better prediction result.

We average the financial model 3 and non-financial model 1 test prediction as the predicted growth rate. The result shows that the RMSE of average predictions in train data is 918.2, which means the bias is not lowered too much. However, the Kaggle result shows that the Ensemble by average significantly improves the model and its rank is higher than non-financial model 3.

The following is to give two models different weights to check how different weights will impact the predictions. Because the best non-financial model which is non-financial model 3 performs better than financial model 3, we make a hypothesis that giving more weights to non-financial model will get better result. We take 0.25 as a step to test the result, it turns out that a higher weights on non-financial model than on financial indeed will get a higher score and 0.25 weights on financial model and 0.75 non-financial model ranks the highest among all the weights.

Next, we allocate more elaborate weights to each model from 0.1 to 0.9 by 0.1 to verify hypothesis. The best performance shows when financial model is given 0.4 weights and non-financial model is given 0.6 weights, and it exceeds 0.25 weight result.

```
#-----------create Financial Model 3 and Non Financial Model1 formula--------------#

Fin_Mod <- lm(growth_rate ~ bm + pe_exi + ps + pcf + dpr + npm + opmbd + opmad + gpm + roa + r

Non_Fin_Mod <- lm(growth_rate ~ is_annouce + red_annouce + no_annoucement + TFF + coleman_liau

## Predict in Train data
Train$pre_finaicl_mod3 <- predict(Fin_Mod, Train)
Train$pre_non_fin_mod1 <- predict(Non_Fin_Mod,Train)

## Get Average predicted growth rate
Train$avr_pre <- (Train$pre_finaicl_mod3+Train$pre_non_fin_mod1)/2
Train$wt_pre_0.25 <- (Train$pre_finaicl_mod3*0.25+Train$pre_non_fin_mod1*0.75)
Train$wt_pre_0.4 <- (Train$pre_finaicl_mod3*0.4+Train$pre_non_fin_mod1*0.6)

## Get predicted followers in Train
Train$avr_followers <- (1+Train$avr_pre)*Train$followers_lag
Train$wt_followers_0.25 <- (1+Train$wt_pre_0.25)*Train$followers_lag
Train$wt_followers_0.4 <- (1+Train$wt_pre_0.4)*Train$followers_lag

rmse <- function(v1, v2) {
  sqrt(mean((v1 - v2)^2, na.rm = T))
}

RMSE <- c(rmse(Train$followers,Train$avr_followers),
          rmse(Train$followers,Train$wt_followers_0.25),
          rmse(Train$followers,Train$wt_followers_0.4))

# retrieve RMSE for all the five models
names(RMSE) <- c("Model36", "Model37", "Model38")
RMSE
```

```
Model36 Model37 Model38
  918.2   946.5   928.3
```

```r
#---------------------------Average & Weighted Ensemble-------------------------------------

#merge the last day of followers in Train data to Test data
Test36 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test36 <- Test36 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test37 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test37 <- Test37 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

Test38 <- full_join(last_day_followers, Test, by = join_by(gvkey, date))
Test38 <- Test38 %>% arrange(gvkey, date) %>%
  select(-sic.x) %>%
  rename(`sic` = sic.y) %>%
  select(-ID.x) %>%
  rename(`ID` = ID.y)

## Get predictions in Test
Test36$Pre_finaicl_mod3 <- predict(Fin_Mod,Test36)
Test36$Pre_non_fin_mod1 <- predict(Non_Fin_Mod, Test36)

# use weighted average to make followers prediction
Test37 <- Test36 %>% mutate(wt_pre=(Pre_finaicl_mod3 *0.25 +Pre_non_fin_mod1*0.75))
Test38 <- Test36 %>% mutate(wt_pre=(Pre_finaicl_mod3 *0.4 +Pre_non_fin_mod1*0.6))

## Use average method to make followers prediction
Test36 <- Test36 %>% mutate(avr_pre=(Pre_finaicl_mod3+Pre_non_fin_mod1)/2)

while (any(is.na(Test36$followers))) {
  Test36 <- Test36 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+Test36$avr_pre), followers)
}
while (any(is.na(Test37$followers))) {
  Test37 <- Test37 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+Test37$wt_pre), followers))
}
while (any(is.na(Test38$followers))) {
  Test38 <- Test38 %>%
    mutate(followers= ifelse(is.na(followers), lag(followers) * (1+Test38$wt_pre), followers))
}
```

```r
#merge the six companies with other companies 6492
Test36 <- Test36 %>% select(gvkey, date, followers, ID)
Test36 <- rbind(Test36, Test_6)
Test36 <- Test36 %>% filter(date>=20170701)
```

```
Test36$followers <- round(Test36$followers)
Followers_submisson_Ensembles36 <- Test36 %>% select(ID, followers)

Test37 <- Test37 %>% select(gvkey, date, followers, ID)
Test37 <- rbind(Test37, Test_6)
Test37 <- Test37 %>% filter(date>=20170701)
Test37$followers <- round(Test37$followers)
Followers_submisson_Ensembles37 <- Test37 %>% select(ID, followers)

Test38 <- Test38 %>% select(gvkey, date, followers, ID)
Test38 <- rbind(Test38, Test_6)
Test38 <- Test38 %>% filter(date>=20170701)
Test38$followers <- round(Test38$followers)
Followers_submisson_Ensembles38 <- Test38 %>% select(ID, followers)


#save the sample of submission
write.csv(Followers_submisson_Ensembles36, "submission_Ensembles36.csv", row.names = FALSE)
write.csv(Followers_submisson_Ensembles37, "submission_Ensembles37.csv", row.names = FALSE)
write.csv(Followers_submisson_Ensembles38, "submission_Ensembles38.csv", row.names = FALSE)
```

out of sample result for ensemble models

submission_Ensembles38 1.csv **2975.9825**
Complete · now

submission_Ensembles37 1.csv **2986.46712**
Complete · 17s ago

submission_Ensembles36 1.csv **3060.3061**
Complete · 35s ago

10. Score Summary

There is bias and variance trade-off in model selections. From our analysis results, the best performing model in train which means low bias is not corresponding to the best out of sample which will give a high variance. We have to choose models we want depending on different goals.

```
Model_name <- c("Model1", "Model2", "Model3", "Model4", "Model5", "Model6", "Model7", "Model8"
                "Model9", "Model10", "Model11", "Model12", "Model13", "Model14", "Model15", "M
                "Model17", "Model18", "Model19", "Model20", "Model21", "Model22", "Model23", "
                "Model25", "Model26", "Model27", "Model28", "Model29", "Model30", "Model31", "
                "Model33", "Model34", "Model35", "Model36", "Model37", "Model38")
rmse_value <- c(rmse(Train$followers,Train$Followers_M1),
                rmse(Train$followers,Train$Followers_M2),
                rmse(Train$followers,Train$Followers_M3),
                rmse(Train$followers,Train$Followers_M4),
                rmse(Train$followers,Train$Followers_M5),
                rmse(Train$followers,Train$Followers_M6),
```

```
                rmse(Train$followers,Train$Followers_M7),
                rmse(Train$followers,Train$Followers_M8),
                rmse(Train$followers,Train$Followers_M9),
                rmse(Train$followers,Train$Followers_M10),
                rmse(Train$followers,Train$Followers_M11),
                rmse(Train$followers,Train$Followers_M12),
                rmse(Train$followers,Train$Followers_M13),
                rmse(Train$followers,Train$Followers_M14),
                rmse(Train$followers,Train$Followers_M15),
                rmse(Train$followers,Train$Followers_M16),
                rmse(Train$followers,Train$Followers_M17),
                rmse(Train$followers,Train$Followers_M18),
                rmse(Train$followers,Train$Followers_M19),
                rmse(Train$followers,Train$Followers_M20),
                rmse(Train$followers,Train$Follower_LA_M21),
                rmse(Train$followers,Train$Follower_LA_M22),
                rmse(Train$followers,Train$Follower_LA_M23),
                rmse(Train$followers,Train$Follower_LA_M24),
                rmse(Train$followers,Train$Follower_LA_M25),
                rmse(Train$followers,Train$Followers_XG_26),
                rmse(Train$followers,Train$Followers_XG_27),
                rmse(Train$followers,Train$Followers_XG_28),
                rmse(Train$followers,Train$Followers_XG_29),
                rmse(Train$followers,Train$Followers_XG_30),
                rmse(Train$followers,Train$RTfollowers31),
                rmse(Train$followers,Train$RTfollowers32),
                rmse(Train$followers,Train$RTfollowers33),
                rmse(Train$followers,Train$RTfollowers34),
                rmse(Train$followers,Train$RTfollowers35),
                rmse(Train$followers,Train$avr_followers),
                rmse(Train$followers,Train$wt_followers_0.25),
                rmse(Train$followers,Train$wt_followers_0.4))

kaggle_score <- c("3381.59815", "3522.23707", "4717.58394", "4828.89567", "5222.18741", "5130.
                 "3436.44338", "3392.02717", "3362.3681", "3563.6714", "5103.70607", "5489.166
                 "3088.92331", "5038.20615", "5544.19647", "4464.32719", "4380.31703", "4483.7
                 "5125.92519", "5003.6044", "3312.65099", "3357.82945", "3108.71325",
                 "4449.15789", "4356.33031", "2688.35717", "3021.97579", "1909.79259",
                 "2463.9722", "3192.68171", "2137.71997", "3435.91113", "3663.6778",
                 "2546.87537", "3395.81836", "3060.3061", "2986.46712", "3603.92503")

score_summary <- as.data.frame(cbind(Model_name, rmse_value, kaggle_score))
score_summary$rmse_value <- sprintf(as.numeric(score_summary$rmse_value), fmt = '%.5f')


# install.packages("formattable")
library(formattable)
```

```
Warning: package 'formattable' was built under R version 4.3.3
```

```
customBlue = "#8fcef2"
```

```r
# define row indices
rmse_min <- which(score_summary$rmse_value %in% sort(score_summary$rmse_value)[1:5])
score_min <- which(score_summary$kaggle_score %in% sort(score_summary$kaggle_score)[1:5])

# define coloring
red_green_formatter <- formatter("span",
                                 style = x ~ style(
                                  display = "block",
                                `background-color` = customBlue))

formattable(score_summary,
            align =c("c","c","c"),
            list(area(row = rmse_min,
                      col = 2) ~ red_green_formatter,
            area(row = score_min,
                      col = 3) ~ red_green_formatter))
```

Model_name	rmse_value	kaggle_score
Model1	934.61733	3381.59815
Model2	936.77688	3522.23707
Model3	893.27749	4717.58394
Model4	892.20942	4828.89567
Model5	903.69925	5222.18741
Model6	895.66332	5130.81508
Model7	938.05861	3436.44338
Model8	941.56043	3392.02717
Model9	896.97653	3362.3681
Model10	898.51322	3563.6714
Model11	897.69261	5103.70607
Model12	904.40451	5489.16648
Model13	984.13136	3088.92331
Model14	901.36393	5038.20615
Model15	904.54643	5544.19647
Model16	917.22583	4464.32719
Model17	918.74586	4380.31703

Model_name	rmse_value	kaggle_score
Model18	918.89335	4483.74433
Model19	900.41807	5125.92519
Model20	896.04379	5003.6044
Model21	937.40416	3312.65099
Model22	942.27461	3357.82945
Model23	984.17883	3108.71325
Model24	917.41457	4449.15789
Model25	919.18209	4356.33031
Model26	694.72936	2688.35717
Model27	707.85466	3021.97579
Model28	640.48177	1909.79259
Model29	677.00035	2463.9722
Model30	706.42559	3192.68171
Model31	749.09895	2137.71997
Model32	786.62173	3435.91113
Model33	921.79921	3663.6778
Model34	779.40179	2546.87537
Model35	814.70118	3395.81836
Model36	918.23445	3060.3061
Model37	946.47544	2986.46712
Model38	928.33965	3603.92503

11. Key Takeaways of the Project

In our case machine learning indeed elevates data analysis model prediction. From the overall ranking in leaderboard, Random Forest ranks the highest which means it can predict the expected number of Twitter followers as accurately as possible and Ensemble by weighted average ranks below it. This means machine learning model outperforms domain knowledge model. However,LASSO does not provide better prediction results than the non-financial model

3. Therefore, machine learning does not always performs better compared to domain knowledge model.

Through this project, we went through the hardship of scrapping Twitter followers data and painstakingly write the Rselenium codes to scrap profile information of companies Twitter account given the restriction imposed by Twitter on scrapping data. We understand the need to have proper understanding of our extracted data through Explanatory Data Analysis before model construction. In addition, the data cleaning process and the thought process on how to fill in NA values for our financial and non financial variables plays a key role in ensuring quality data is used for building our model. Quality, clean,and structured data forms the basis of good model building and accurate prediction results.

The use of Advanced Machine Learning algorithms has the capability to boost our model prediction's abilities both in sample and out of sample. However, we cannot use Advanced Machine Learning Alogrithm blindly without proper domain knowledge. For instance, LASSO performs feature selection using coefficient shrinkage without domain knowledge applied. Like in our case LASSO lambda.1SE removes all features of our model rendering it ineffective to use for enhancing our model prediction accuracy.

Understanding our dataset leads to better choice of Advanced Machine Learning algorithms used for improving our model predictions. For example, the use of XGBoost ensembling method seems to perform better than the use of Random Forest and simple ensembling methods. Hence, through this project, we witness the capabilities of advanced machine learning algorithm better suited for different types of model.

12. References

Hutto, C. J., Yardi, S., & Gilbert, E. (2013). A longitudinal study of follow predictors on twitter. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 821–830. https://doi.org/10.1145/2470654.2470771

Mueller, J., & Stumme, G. (2017). Predicting Rising Follower Counts on Twitter Using Profile Information. Proceedings of the 2017 ACM on Web Science Conference, 121–130. https://doi.org/10.1145/3091478.3091490

Sayed, D. E. (n.d.). The effect of financial ratios on Twitter sentiment and volume.

Tsileponis, N., Stathopoulos, K., & Walker, M. (2020). Do corporate press releases drive media coverage? The British Accounting Review, 52(2), 100881. https://doi.org/10.1016/j.bar.2020.100881