Netflix daily change based on News

QBS181 WallStreetBets

11/6/2021

News analysis based on daily rise and fall

Getting the data of Netflix daily returns

We have a file including daily returns for all five stocks and we use Netflix as an example to show the relationship between news keywords and stock price. Now, we select Netflix data as a new data set.

```
portfolio_daily <- read.csv("portfolio_daily_ret.csv",header=TRUE)
```

```
# Select Netflix data
Netflix_daily <- portfolio_daily[which(portfolio_daily$symbol=="NFLX"),]
```

Then we use which() function to get Netflix data which daily rise or fall of more than 8% and save them as a new csv file.

```
# Select Netflix data with daily increase or decrease of more than 8%
Netflix_daily_change <- data.frame(Netflix_daily[which(abs(Netflix_daily$returns)>0.08),])
Netflix_daily_change
```

```
##         X symbol       date     returns
## 3       3   NFLX 2016-01-06  0.09307074
## 8       8   NFLX 2016-01-13 -0.08594960
## 74     74   NFLX 2016-04-19 -0.12970485
## 137   137   NFLX 2016-07-19 -0.13126204
## 201   201   NFLX 2016-10-18  0.19028054
## 388   388   NFLX 2017-07-18  0.13543605
## 518   518   NFLX 2018-01-23  0.09978904
## 576   576   NFLX 2018-04-17  0.09188381
## 699   699   NFLX 2018-10-10 -0.08383227
## 709   709   NFLX 2018-10-24 -0.09403895
## 751   751   NFLX 2018-12-26  0.08461601
## 757   757   NFLX 2019-01-04  0.09723449
## 891   891   NFLX 2019-07-18 -0.10272048
## 1055 1055   NFLX 2020-03-12 -0.09907982
## 1057 1057   NFLX 2020-03-16 -0.11138862
## 1062 1062   NFLX 2020-03-23  0.08244450
## 1138 1138   NFLX 2020-07-10  0.08068767
## 1171 1171   NFLX 2020-08-26  0.11608717
## 1223 1223   NFLX 2020-11-09 -0.08592851
## 1271 1271   NFLX 2021-01-20  0.16854344
```

```
# Save the data as a new csv file
write.csv(Netflix_daily_change,"Netflix_daily_change.csv",row.names = FALSE)
```

Show the frequcency of words in news

We use Excel to add a new column "News" and search for relational news by the date to fill in the new column. Then save it as a new csv file and load into R. Delete empty rows and columns.

```
# Load new data set and delete empty rows/columns
News <- read.csv("Netflix Daily News.csv",header = TRUE)
News <- News[c(1:20),c(1:5)]
```

First, we use the stringr package in R and get the number of characters in the text column "News". Then we use pattern matching to find spaces and count the number of words in the text column.

It is noteworthy that we need to add one at pattern part, since the first word will always be omitted as it is not carried out by spaces.

```
# Show the number of words in the News column
(str_count(News$News))
```

```
##  [1] 135  24 287  24  68 146  24 274  62  24 285  24 194 448 225 118 149 321 140
## [20] 148
```

```
(str_count(News$News,pattern=" ")+1)
```

```
##  [1] 20  4 51  4  8 27  4 44 11  4 48  4 31 74 35 22 22 52 23 24
```

```
#the one is necessary because the first word will always be missed as it's not proceeded by a space
```

Next, we use tidyverse package and create a new cleaning data which just include the News column.

```
# Get the cleaning News column as a new data set
News_information <-News %>%
  dplyr::select(News) %>%
  mutate_all(funs(str_replace_na(.,"")))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
News_information
```

```
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14 U.S. stocks went south on Wednesday, after the World Health Organization (WHO) declared coronavir
## 15
## 16
## 17
## 18
## 19
## 20
```

Then we get "NewsSub" by mutate() function which can add new variables response by filling from 1 to the length of "News" and preserve existing News column.

```
(NewsSub <- News_information %>% mutate(response=1:length(News_information$News)))
```

```
##
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
## 9
## 10
## 11
## 12
## 13
## 14 U.S. stocks went south on Wednesday, after the World Health Organization (WHO) declared coronavir
## 15
## 16
## 17
## 18
## 19
## 20
##    response
## 1         1
```

```
## 2         2
## 3         3
## 4         4
## 5         5
## 6         6
## 7         7
## 8         8
## 9         9
## 10       10
## 11       11
## 12       12
## 13       13
## 14       14
## 15       15
## 16       16
## 17       17
## 18       18
## 19       19
## 20       20
```

Next step, we split the column "News" into tokens by unnest_tokens() function.

```
# Split News into tokens
NewsSub %<>%
  unnest_tokens(word, News)
head(NewsSub,10)
```

```
##    response           word
## 1         1        netflix
## 2         1       launched
## 3         1            its
## 4         1        service
## 5         1       globally
## 6         1 simultaneously
## 7         1       bringing
## 8         1            its
## 9         1       internet
## 10        1             tv
```

Now we can see that the data is displayed as one-word-per-row format.

The other thing we need to notice is stop words. Usually, some words appear frequently, but they provide little information and can not help analysis. Like "is", "it", "the", "a","of","to", etc., these are called stop words, and we need to remove them from the analysis by anti_join() function.

```
# Load data of stop words
data(stop_words)
# Remove stop words
NewsSub <- NewsSub %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
head(NewsSub,10)
```

```
##    response           word
## 1         1        netflix
## 2         1       launched
## 3         1        service
## 4         1       globally
## 5         1 simultaneously
## 6         1       bringing
## 7         1       internet
## 8         1             tv
## 9         1        network
## 10        1            130
```

Show the number of occurrences of each word

Now, we can use the count() function from dplyr to find the number of occurrences of each word and most represented words from the "News" column.

```
# Count the number of occurrences of each word
NewsSub %>%
  dplyr::count(word,sort = TRUE)
```

```
##              word  n
## 1          netflix 17
## 2         releases  6
## 3      subscribers  6
## 4         pandemic  5
## 5        streaming  5
## 6            added  4
## 7            close  4
## 8      coronavirus  4
## 9              dow  4
## 10           giant  4
## 11         million  4
## 12         quarter  4
## 13           stock  4
## 14       customers  3
## 15          dollar  3
## 16        earnings  3
## 17           month  3
## 18            plan  3
## 19            2020  2
## 20             500  2
## 21          closed  2
## 22        declared  2
## 23          donald  2
## 24            fell  2
## 25           index  2
## 26           jones  2
## 27          nasdaq  2
## 28         october  2
```

```
## 29       president  2
## 30           price  2
## 31          recent  2
## 32          record  2
## 33        released  2
## 34         service  2
## 35            time  2
## 36           trump  2
## 37             u.s  2
## 38           world  2
## 39             0.7  1
## 40               1  1
## 41              10  1
## 42              13  1
## 43             130  1
## 44            19.8  1
## 45        2,351.10  1
## 46             2.2  1
## 47             2.7  1
## 48             2.9  1
## 49             200  1
## 50            2008  1
## 51            2011  1
## 52            2019  1
## 53       21,792.20  1
## 54           25.86  1
## 55               3  1
## 56              30  1
## 57              34  1
## 58           36.07  1
## 59              40  1
## 60            5.96  1
## 61              50  1
## 62             580  1
## 63         6192.92  1
## 64            7.41  1
## 65               8  1
## 66             800  1
## 67               9  1
## 68          adding  1
## 69        addition  1
## 70      aggressive  1
## 71       alongside  1
## 72         analyst  1
## 73        analysts  1
## 74       announced  1
## 75         average  1
## 76              ba  1
## 77        backlash  1
## 78       beginning  1
## 79      benchmarks  1
## 80     benefitting  1
## 81             bid  1
## 82         biggest  1
```

```
## 83            bill  1
## 84         billion  1
## 85         blowout  1
## 86           boeing  1
## 87        bringing  1
## 88        buybacks  1
## 89          caused  1
## 90            cboe  1
## 91         company  1
## 92       composite  1
## 93       countries  1
## 94        customer  1
## 95             day  1
## 96            days  1
## 97       declining  1
## 98             dji  1
## 99            drop  1
## 100            due  1
## 101            dvd  1
## 102      emergency  1
## 103         europe  1
## 104      expecting  1
## 105           fall  1
## 106          fears  1
## 107        feature  1
## 108       february  1
## 109           film  1
## 110          films  1
## 111         finish  1
## 112       finished  1
## 113        foreign  1
## 114         friday  1
## 115          funds  1
## 116           gain  1
## 117       globally  1
## 118           goal  1
## 119  grandfathered  1
## 120          green  1
## 121         growth  1
## 122             hd  1
## 123         health  1
## 124          heavy  1
## 125           home  1
## 126        imposed  1
## 127      increases  1
## 128        indices  1
## 129     industrial  1
## 130       industry  1
## 131       internet  1
## 132       investor  1
## 133         jumped  1
## 134       launched  1
## 135           lost  1
## 136          major  1
```

```
## 137        managed  1
## 138            mar  1
## 139         market  1
## 140   meaningfully  1
## 141         monday  1
## 142         months  1
## 143          names  1
## 144       national  1
## 145      nationals  1
## 146        network  1
## 147           news  1
## 148          onset  1
## 149   organization  1
## 150           paid  1
## 151         people  1
## 152           play  1
## 153      plummeted  1
## 154        plunges  1
## 155        posting  1
## 156       pressure  1
## 157       previous  1
## 158         prices  1
## 159         pulled  1
## 160             q4  1
## 161      rages.the  1
## 162          raise  1
## 163          rally  1
## 164        release  1
## 165         report  1
## 166       reported  1
## 167       revealed  1
## 168         senate  1
## 169        setting  1
## 170         shares  1
## 171       shipping  1
## 172 simultaneously  1
## 173          sinks  1
## 174          south  1
## 175          split  1
## 176         spread  1
## 177         stalls  1
## 178       standard  1
## 179          start  1
## 180        staying  1
## 181          steep  1
## 182          stick  1
## 183         stocks  1
## 184         stoked  1
## 185      surpassed  1
## 186         survey  1
## 187     suspension  1
## 188           tech  1
## 189      temporary  1
## 190           term  1
```

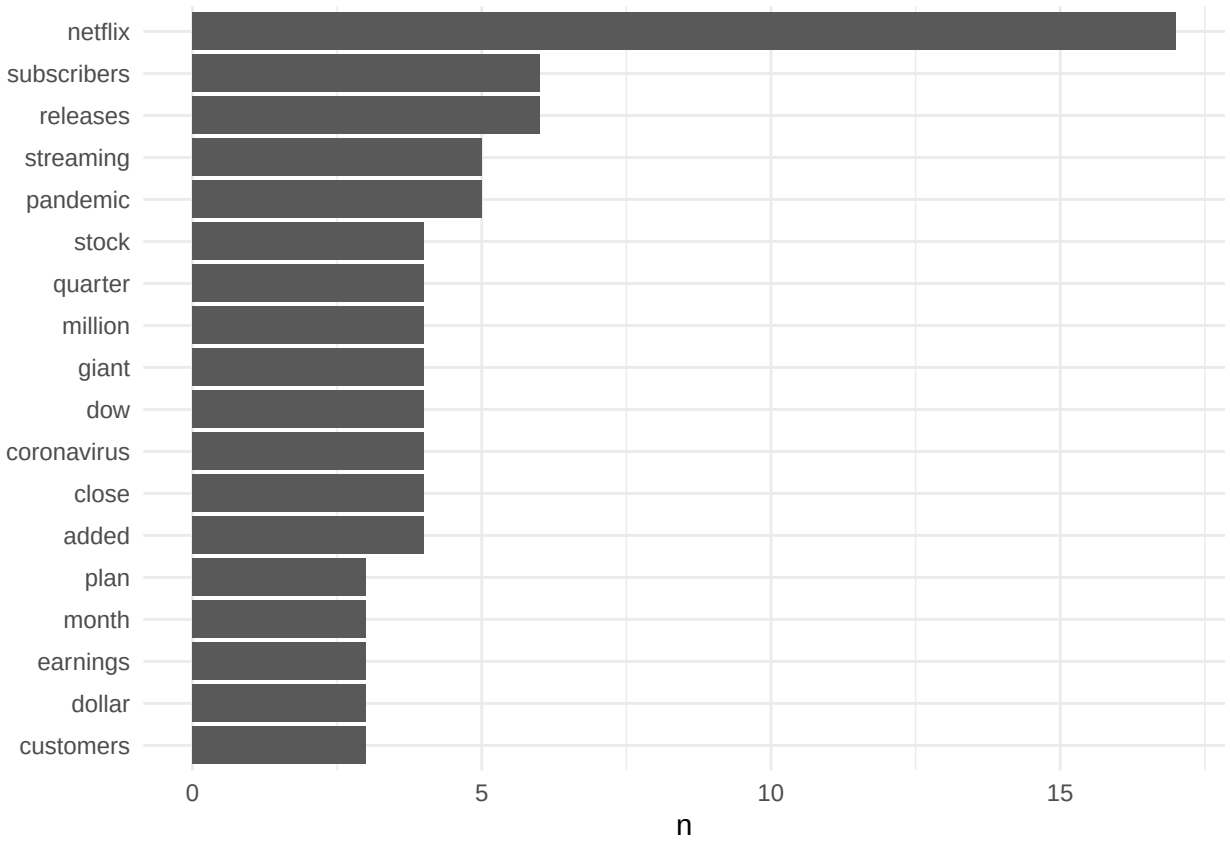
```
## 191          total  1
## 192     travelling  1
## 193             tv  1
## 194         united  1
## 195  unprecedented  1
## 196        vaccine  1
## 197         viewer  1
## 198          virus  1
## 199            vix  1
## 200     volatility  1
## 201            web  1
## 202      wednesday  1
## 203          worst  1
```

From the above output, we find that "releases" and "subscribers" these two words appear most frequently except "netflix" (Since Netflix is the company name of our data, we can just ignore it).

Display the plot to show high frequency words

For more intuitive observation, we display a plot to show words that appear more than twice.

```
# Select rows with more than 2 occurrences of words and get a plot
NewsSub %>%
  dplyr::count(word, sort = TRUE) %>%
  filter(n > 2) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL) + theme_minimal()
```



Based on the above plot, we find that "netflix" as the company name has highest frequency. Next, "subscribers" and "releases" appear more times. These two words, the former represents the number of subscriptions of the video website, which is closely related to profits and stock prices, and the latter represents the launch of new movies or dramas, which is a representative word of the video website products. So, they are reasonable to have a high frequency.

Besides, "pandemic", "stock", "quarter", "dow", "earning", etc., they all appear more than twice and show some general rule. Netflix stock price will change with the social environment and the stock market environment, so pandemic and stock indexes (Dow Jones index) will become news keywords. In addition, stock price reflects the operating conditions of the company, so the financial report (quarter earning report) is highly related to the stock price.

Overall, in the above analysis, we use Netflix as the example to prove the relationship between the daily rise and fall of stock price and news. To a certain extent, we can predict the feasibility of stock changes by comparing some key words of the news.

To better illustrate the impact of news keywords on stock price, we could do the same analysis on daily rise data and daily fall data respectively.

Analysis on daily rise data

Like we do for the whole data, we repeat the same steps to get the number of occurrences of each word and find words with high correlation with the rise of stock price.

```
# Select the daily rise data form the News data set
News_increase <- News[which(News$returns>0),]
```

```
News_increase <-News_increase %>%
  dplyr::select(News) %>%
  mutate_all(funs(str_replace_na(.,"")))
News_increase
```

```
##
## 1
## 5
## 6
## 7
## 8                                                    Netflix released their first-quarter earnings and
## 11                                      The Dow Jones Industrial Average (DJI) fell 2.9% to close at
## 12
## 16
## 17
## 18 Netflix has seen record-setting viewer growth since the onset of the coronavirus pandemic. The st
## 20
```

```
(IncreaseSub <- News_increase %>% mutate(response=1:length(News_increase$News)))
```

```
##
## 1
## 5
## 6
## 7
## 8                                                    Netflix released their first-quarter earnings and
## 11                                      The Dow Jones Industrial Average (DJI) fell 2.9% to close at
## 12
## 16
## 17
## 18 Netflix has seen record-setting viewer growth since the onset of the coronavirus pandemic. The st
## 20
##    response
## 1         1
## 5         2
## 6         3
## 7         4
## 8         5
## 11        6
## 12        7
## 16        8
## 17        9
## 18       10
## 20       11
```

```
# Split column to unnest tokens
IncreaseSub %<>%
  unnest_tokens(word, News)
head(IncreaseSub,10)
```

```
##    response           word
## 1         1        netflix
```

```
## 2         1       launched
## 3         1            its
## 4         1        service
## 5         1       globally
## 6         1 simultaneously
## 7         1       bringing
## 8         1            its
## 9         1       internet
## 10        1             tv
```

```
# Remove stop words
IncreaseSub <- IncreaseSub %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
head(IncreaseSub,10)
```

```
##    response           word
## 1         1        netflix
## 2         1       launched
## 3         1        service
## 4         1       globally
## 5         1 simultaneously
## 6         1       bringing
## 7         1       internet
## 8         1             tv
## 9         1        network
## 10        1            130
```

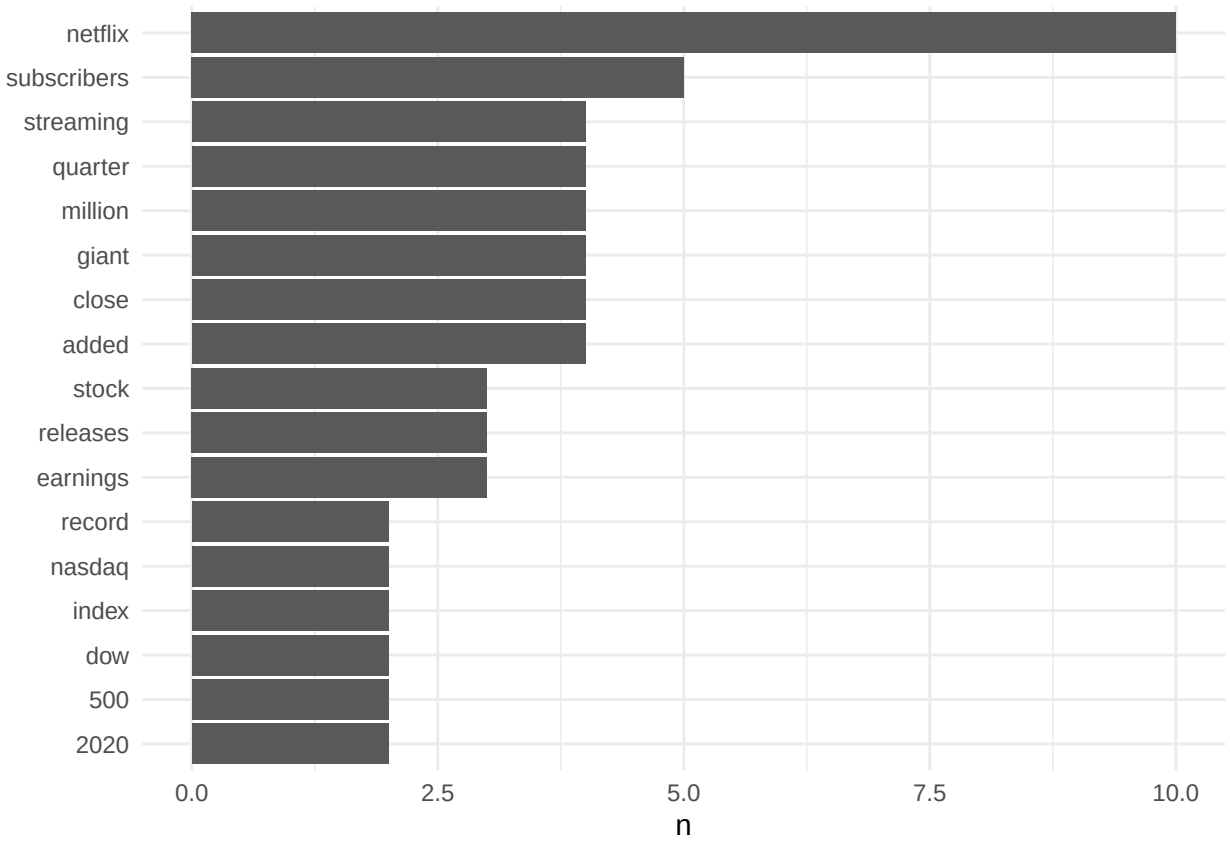
```
# Count the number of occurences of each word
IncreaseSub %>%
  dplyr::count(word, sort = TRUE)
```

```
##              word  n
## 1         netflix 10
## 2     subscribers  5
## 3           added  4
## 4           close  4
## 5           giant  4
## 6         million  4
## 7         quarter  4
## 8       streaming  4
## 9        earnings  3
## 10       releases  3
## 11          stock  3
## 12           2020  2
## 13            500  2
## 14            dow  2
## 15          index  2
## 16         nasdaq  2
## 17         record  2
```

```
## 18            0.7  1
## 19              1  1
## 20            130  1
## 21           19.8  1
## 22       2,351.10  1
## 23            2.2  1
## 24            2.7  1
## 25            2.9  1
## 26            200  1
## 27           2019  1
## 28      21,792.20  1
## 29          25.86  1
## 30              3  1
## 31             34  1
## 32          36.07  1
## 33             40  1
## 34           5.96  1
## 35            580  1
## 36        6192.92  1
## 37           7.41  1
## 38         adding  1
## 39      aggressive  1
## 40         analyst  1
## 41        analysts  1
## 42         average  1
## 43            bill  1
## 44         blowout  1
## 45        bringing  1
## 46        buybacks  1
## 47            cboe  1
## 48          closed  1
## 49       composite  1
## 50     coronavirus  1
## 51       countries  1
## 52       declining  1
## 53             dji  1
## 54       expecting  1
## 55         feature  1
## 56            fell  1
## 57            film  1
## 58           films  1
## 59          finish  1
## 60        finished  1
## 61        globally  1
## 62            goal  1
## 63          growth  1
## 64           heavy  1
## 65         indices  1
## 66      industrial  1
## 67        industry  1
## 68        internet  1
## 69           jones  1
## 70          jumped  1
## 71        launched  1
```

```
## 72           major  1
## 73         managed  1
## 74          market  1
## 75    meaningfully  1
## 76          monday  1
## 77          months  1
## 78         network  1
## 79           onset  1
## 80        pandemic  1
## 81            plan  1
## 82            play  1
## 83       plummeted  1
## 84        previous  1
## 85              q4  1
## 86       rages.the  1
## 87           rally  1
## 88          recent  1
## 89         release  1
## 90        released  1
## 91          report  1
## 92        reported  1
## 93        revealed  1
## 94          senate  1
## 95         service  1
## 96         setting  1
## 97  simultaneously  1
## 98           sinks  1
## 99          stalls  1
## 100          stick  1
## 101      surpassed  1
## 102         survey  1
## 103           tech  1
## 104           time  1
## 105          total  1
## 106             tv  1
## 107  unprecedented  1
## 108         viewer  1
## 109          virus  1
## 110            vix  1
## 111     volatility  1
## 112            web  1
## 113          world  1
```

```
# Show rows with more than one occurrence of words and get a plot
IncreaseSub %>%
  dplyr::count(word, sort = TRUE) %>%
  filter(n > 1) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL) + theme_minimal()
```



The above outputs show that "subscribers", "quarter", "stock", "earnings", etc. are more likely to give a positive impact on stock price. Besides, "nasdaq", "dow" and "index" are also positively correlated with the daily rise of stock price.

Analysis on daily fall data

Do the same steps as the daily rise data and get words with high correlation with the fall of stock price.

```
# Select the daily rise data form the News data set
News_decrease <- News[which(News$returns<0),]
```

```
News_decrease <-News_decrease %>%
  dplyr::select(News) %>%
  mutate_all(funs(str_replace_na(.,"")))
News_decrease
```

```
##
## 2
## 3
## 4
## 9
## 10
## 13
## 14 U.S. stocks went south on Wednesday, after the World Health Organization (WHO) declared coronavir
## 15
## 19
```

```
(DecreaseSub <- News_decrease %>% mutate(response=1:length(News_decrease$News)))
```

```
##
## 2
## 3
## 4
## 9
## 10
## 13
## 14 U.S. stocks went south on Wednesday, after the World Health Organization (WHO) declared coronavir
## 15
## 19
##    response
## 2         1
## 3         2
## 4         3
## 9         4
## 10        5
## 13        6
## 14        7
## 15        8
## 19        9
```

```
# Split column to unnest tokens
DecreaseSub %<>%
  unnest_tokens(word, News)
head(DecreaseSub,10)
```

```
##    response     word
## 1         1      new
## 2         1 releases
## 3         1       on
## 4         1  netflix
## 5         2  netflix
## 6         2   prices
## 7         2     will
## 8         2    start
## 9         2    going
## 10        2       up
```

```
# Remove stop words
DecreaseSub <- DecreaseSub %>%
  anti_join(stop_words)
```

```
## Joining, by = "word"
```

```
head(DecreaseSub,10)
```

```
##    response      word
## 1         1  releases
## 2         1   netflix
```

```
## 3         2   netflix
## 4         2    prices
## 5         2     start
## 6         2 customers
## 7         2   netflix
## 8         2 announced
## 9         2   october
## 10        2     raise
```

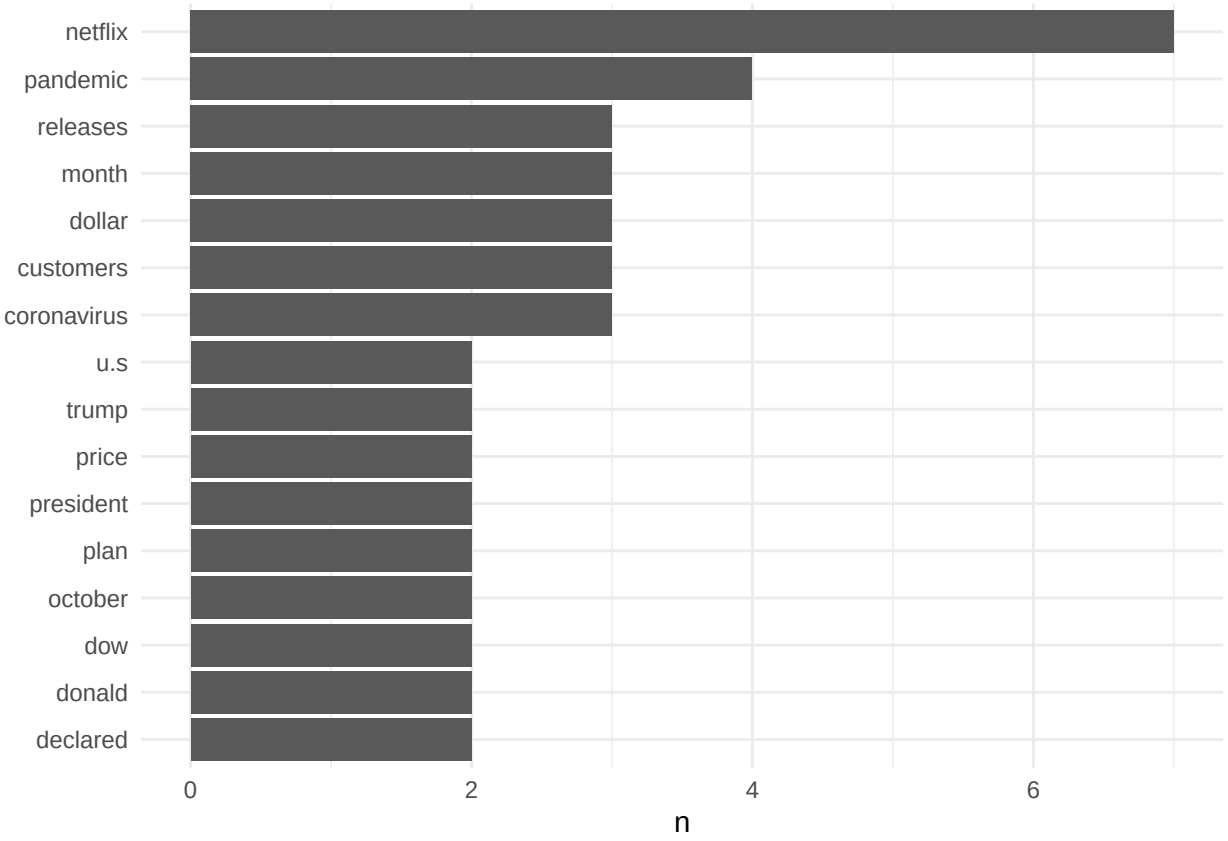
```
# Count the number of occurences of each word
DecreaseSub %>%
  dplyr::count(word, sort = TRUE)
```

```
##             word n
## 1        netflix 7
## 2       pandemic 4
## 3     coronavirus 3
## 4      customers 3
## 5         dollar 3
## 6          month 3
## 7       releases 3
## 8       declared 2
## 9         donald 2
## 10           dow 2
## 11       october 2
## 12          plan 2
## 13     president 2
## 14         price 2
## 15         trump 2
## 16           u.s 2
## 17            10 1
## 18            13 1
## 19          2008 1
## 20          2011 1
## 21            30 1
## 22            50 1
## 23             8 1
## 24           800 1
## 25             9 1
## 26      addition 1
## 27     alongside 1
## 28     announced 1
## 29            ba 1
## 30      backlash 1
## 31     beginning 1
## 32    benchmarks 1
## 33   benefitting 1
## 34           bid 1
## 35       biggest 1
## 36       billion 1
## 37        boeing 1
## 38        caused 1
## 39        closed 1
## 40       company 1
```

```
## 41       customer 1
## 42            day 1
## 43           days 1
## 44           drop 1
## 45            due 1
## 46            dvd 1
## 47      emergency 1
## 48         europe 1
## 49           fall 1
## 50          fears 1
## 51       february 1
## 52           fell 1
## 53        foreign 1
## 54         friday 1
## 55          funds 1
## 56           gain 1
## 57  grandfathered 1
## 58          green 1
## 59             hd 1
## 60         health 1
## 61           home 1
## 62        imposed 1
## 63      increases 1
## 64       investor 1
## 65          jones 1
## 66           lost 1
## 67            mar 1
## 68          names 1
## 69       national 1
## 70      nationals 1
## 71           news 1
## 72   organization 1
## 73           paid 1
## 74         people 1
## 75        plunges 1
## 76        posting 1
## 77       pressure 1
## 78         prices 1
## 79         pulled 1
## 80          raise 1
## 81         recent 1
## 82       released 1
## 83        service 1
## 84         shares 1
## 85       shipping 1
## 86          south 1
## 87          split 1
## 88         spread 1
## 89       standard 1
## 90          start 1
## 91        staying 1
## 92          steep 1
## 93          stock 1
## 94         stocks 1
```

```
## 95          stoked 1
## 96       streaming 1
## 97     subscribers 1
## 98      suspension 1
## 99       temporary 1
## 100           term 1
## 101           time 1
## 102     travelling 1
## 103         united 1
## 104        vaccine 1
## 105      wednesday 1
## 106          world 1
## 107          worst 1
```

```
# Show rows with more than one occurrence of words and get a plot
DecreaseSub %>%
  dplyr::count(word, sort = TRUE) %>%
  filter(n > 1) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(n, word)) +
  geom_col() +
  labs(y = NULL) + theme_minimal()
```



The above outputs show that “pandemic”, “releases”, “coronavirus”, “price”, etc. are more likely to give a negative impact on stock price.

If we want to get more accurate news keywords and better predict the rise and fall, we can collect more news data for the above analysis. For different stocks, the above keywords have certain generality, but they do

not necessarily represent accuracy.