Insider Trading Analysis

Nihal Desai ·Ramya Kondrakunta · Santosh Saranyan · Srinath Iyer

Summary

The project aims to deal with collecting and analyzing Insider trading data. This project will collect data and do an exploratory analysis of the insider filing and stock price data, based on the Form 4 filings from the U.S. Securities and Exchange Commission (SEC)[1] and Yahoo Finance Data[2]. In this project, we aim to answer questions such as 'How to collect insider trading data from the US SEC website?', 'How do companies' insider trade?', 'How trading behavior of insiders' changes over time, or by the position of an insider?', so on. Additionally, we try to check if there is a correlation between insider trading and stock prices of the company?

Insider Trading is the trading of public company stocks or securities, based on nonpublic information about the company. An example can be a CEO buying shares right before making a big announcement, or a CFO selling stocks based on the quarterly sales information before they are made public. To avoid this the SEC requires Chief Officers, Directors, so on to file Form 4 whenever they trade their company's share[3][4].

For this project, we are generating the Insider trading dataset by scraping the SEC website. We will be using APIs from Yahoo Finance to get a company's stock prices. We will be using methods such as web scraping to collect data, statistical inferences, and data visualization to analyze and present the findings. Finally, we used tools such as R, Python, and Excel to present the most relevant visualizations and work on the suggested analysis.

Method

Web Scraping

The first method that we used to collect Insider data from the U.S. Securities and Exchange Commission was Web Scraping. We were unable to find any sensible data on insider trading. So, we decided to collect data from the U.S. Securities and Exchange Commission website which is an independent United States Federal Agency, and its primary purpose is to enforce the law against market manipulation.

We constructed a code that scrapes the SEC website and finds Form 4. The code starts at the SEC daily index listing page, then drills down into the year and Quarter of the desired data interval. It downloads a list of all the filings filled by companies, individuals, etc. to the SEC. These are different types of forms such as 8-K (quarter sales report). The code then filters out all the forms other than Form 4. After this, we have a dataset that includes the *date, fid (form id), and link (to the form).*

Table 1

S.No	Attribute Name	Description	Data Type
1	new_relationship	Tidy version of relationship attribute	String
2	isCMO	Indicates if report owner is a CMO or not	Logical
3	isCTO	Indicates if report owner is a CTO or not	Logical
4	isCFO	Indicates if report owner is a CFO or not	Logical
5	isCEO	Indicates if report owner is a CEO or not	Logical
6	isCAO	Indicates if report owner is a CAO or not	Logical
7	isCRO	Indicates if report owner is a CRO or not	Logical
8	isCIO	Indicates if report owner is a CIO or not	Logical
9	isCSO	Indicates if report owner is a CSO or not	Logical
10	isEVP	Indicates if report owner is an EVP or not	Logical
11	isPresident_vp	Indicates if report owner is a president/VP or not	Logical
12	isDirector	Indicates if report owner is a director or not	Logical
13	isSecretary	Indicates if report owner is a secretary or not	Logical
14	isChairman	Indicates if report owner is a chairman or not	Logical
15	Is10percentowner	Indicates if report owner is a 10% owner or not	Logical
16	percentchangeafterweek	Percentage change in share price after a week	Numeric
17	percentchangeafteronemonth	Percentage change in share price after a month	Numeric

Using the collected dataset, we then scrape every single Form 4 filling. After collecting the following data, we needed data about the share prices of the company after a week, month, quarter, six months, and a year. To collect this data, we used Yahoo Finance API. First, we found the *ticker (a symbol used to trade in the open market)* of each company using the *cik*. We then used the *ticker* and got the share prices of the company for the above-mentioned time intervals.

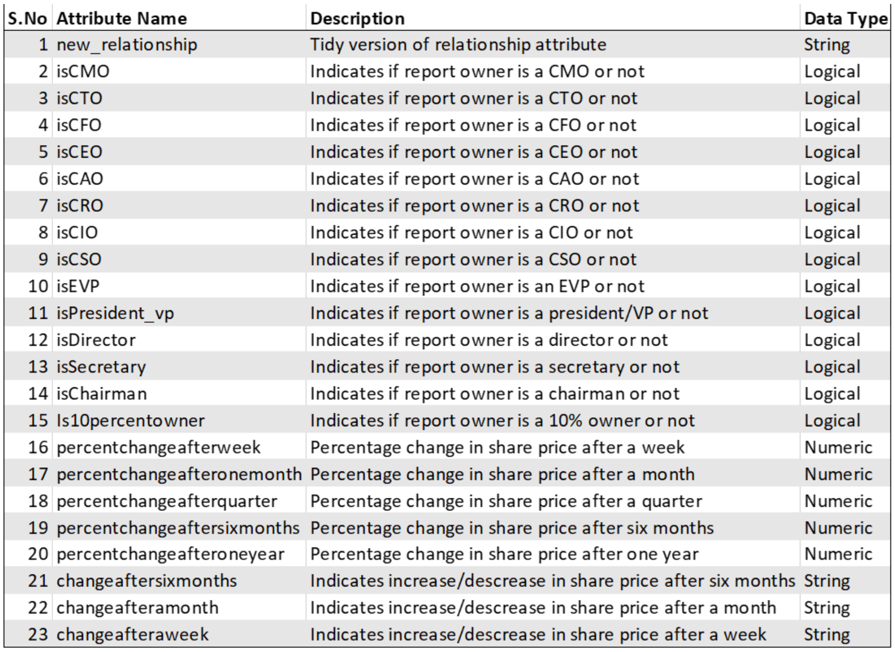
The final data set had the data about the insider trading, and the share prices of the company after a week, month, quarter, six months, and a year after which we started analyzing the dataset.

Data Transformation and Tidying

Since we collected the data in a tidied format because of web-scraping, there wasn't much tidying required other than handling NA values majorly. After importing the dataset in a .csv format, we got rid of all NA values in the observations using 'na.omit' function since imputing the missing values would have likely resulted in a tampered analysis for our purpose. We then filtered the dataset further to not contain '-1' value in any of the observations. We chose to include purchase transactions indicated by 'P' under the 'transactioncode' variable.

To calculate the percentage change of a share price within a certain period, we created individual variables namely percentchangeafterweek, percentchangeaftermonth, percentchangeafterquarter, percentchangeaftersixmonths, and percentchangeafteryear which shows either a positive value indicating a positive price change or a negative value indicating a negative price change. Based on whether a share price increases or decreases over a certain period, we generated four more columns namely changeafteraweek, changeafteronemonth, changeafteraquarter and changeaftersixmonths which are categorical and contain two categories where 'Decrease' indicates a decrease in the share price while 'Increase' indicates an increase in the share price. These columns are helpful to get an overview regarding the price change of certain stocks over weeks or months.

Table 2



S.No	Attribute Name	Description	Data Type
1	new_relationship	Tidy version of relationship attribute	String
2	isCMO	Indicates if report owner is a CMO or not	Logical
3	isCTO	Indicates if report owner is a CTO or not	Logical
4	isCFO	Indicates if report owner is a CFO or not	Logical
5	isCEO	Indicates if report owner is a CEO or not	Logical
6	isCAO	Indicates if report owner is a CAO or not	Logical
7	isCRO	Indicates if report owner is a CRO or not	Logical
8	isCIO	Indicates if report owner is a CIO or not	Logical
9	isCSO	Indicates if report owner is a CSO or not	Logical
10	isEVP	Indicates if report owner is an EVP or not	Logical
11	isPresident_vp	Indicates if report owner is a president/VP or not	Logical
12	isDirector	Indicates if report owner is a director or not	Logical
13	isSecretary	Indicates if report owner is a secretary or not	Logical
14	isChairman	Indicates if report owner is a chairman or not	Logical
15	Is10percentowner	Indicates if report owner is a 10% owner or not	Logical
16	percentchangeafterweek	Percentage change in share price after a week	Numeric
17	percentchangeafteronemonth	Percentage change in share price after a month	Numeric
18	percentchangeafterquarter	Percentage change in share price after a quarter	Numeric
19	percentchangeaftersixmonths	Percentage change in share price after six months	Numeric
20	percentchangeafteroneyear	Percentage change in share price after one year	Numeric
21	changeaftersixmonths	Indicates increase/descrease in share price after six months	String
22	changeafteramonth	Indicates increase/descrease in share price after a month	String
23	changeafteraweek	Indicates increase/descrease in share price after a week	String

The initial dataset contained a 'relationship' variable indicating whether a certain transaction in observation was performed by a CEO, CFO, Director, President, etc. of that company. As this variable contained several observations with multiple relationship positions, we found a need to segregate all the major relationship positions into separate variables of their own which would contain binary values defined as 'True' or 'False'. The relationship positions that we considered for this analysis were Director, President/VP, CEO, CFO, ten percent owner, Secretary, CAO, CTO, CIO, CMO, CRO, CSO, and Chairman which amounted to 13 relationship positions. Each of these positions has a dedicated variable namely isDirector, isCEO, isCFO, etc. containing categorical data to indicate whether a certain individual helms a single position or multiple positions.

Analysis

Data Visualization

For our visualizations, we first used bar plots to see how different types of transactions were distributed and how they were split amongst different positions. Furthermore, we used line plots with the stat value as a count, to visualize purchase and sales transactions over time as well as general trends of share prices over a certain period. We looked at the trends for a week, a month, six months, and ultimately a year. Based on the plots, we decided that looking at share prices a month after purchasing them was the best choice. We also created boxplots to get an idea of the relationship between sales and purchase transactions and the change in percentage of the price of shares. Finally, we created a scatterplot to determine if there was an increase in the number of shares that increased in price after a purchase transaction was made, and the number of shares bought in that transaction. From what we observed, we chose an optimum value as a threshold of the number of shares involved in a transaction based on both the number of shares that increased and the number of observations we had to work with (there were fewer rows of data for transactions with a greater number of shares).

Statistical Inference

We used p-values and Chi-square values to do hypothesis testing. We calculated the p-values and Chi-square values to check which relationship (role) influenced the change in share prices of the company. We also looked at proportion increase as p-value penalized relationship (role) with fewer data points. Similarly, we used proportion to increase to select a threshold for the number of shares purchased in that transaction.

Results

Transaction Segregation

The number of transactions for each of the transaction codes is visualized to see how they are distributed.

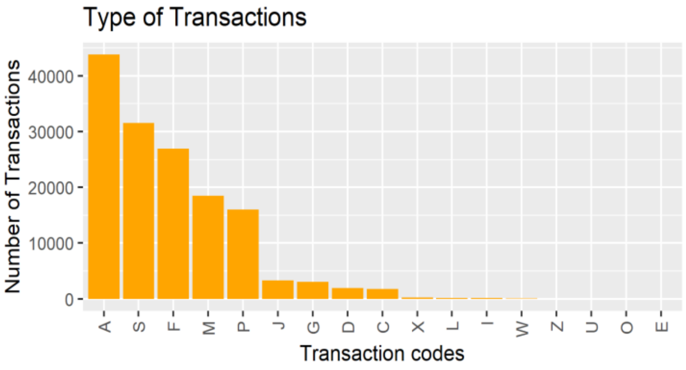


Figure 1: Transaction Types found in Form 4

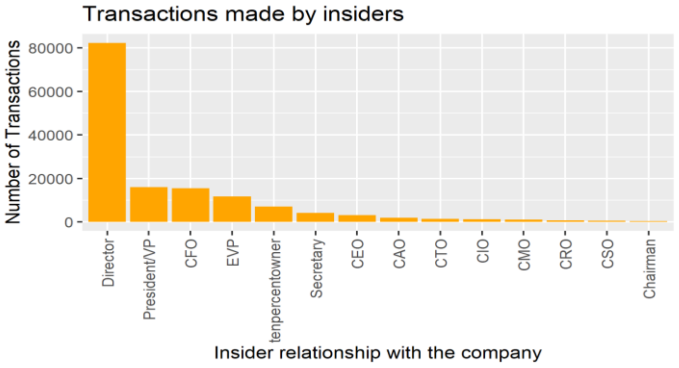


Figure 2: Transactions by Insiders

From Fig. 1, we can see that acquisitions have the greatest number of transactions, which is because the employees get paid stocks or get grants or get awarded stocks. It is closely followed by sales. There are a lot of sales transactions as the employees usually sell their stocks for their expenditure. For this project, we were mostly concerned with the purchase and sales transactions.

Next, the number of transactions made by each role in the company is visualized as shown in Fig. 2. In Fig.2, we can see that the directors have made the greatest number of transactions, and this is because there are many directors compared to the other positions as each company may have two or more directors. The presidents and CFOs have the next highest transactions made.

General Trends

Fig. 3 visualizes the number of sales and purchase transactions on a day-to-day basis. There are days where there are fewer sales transactions and they almost equal the number of purchase transactions, however, we see that generally, there are almost always more sales transactions than purchase transactions.

It is observed from Fig. 3, that there is a high spike in purchase transactions and a big drop in sales during a certain period which corresponds to March 2020. This was the time of the COVID-19 outbreak and at this time the stock prices most likely dropped leading to more insiders buying stocks rather than selling.

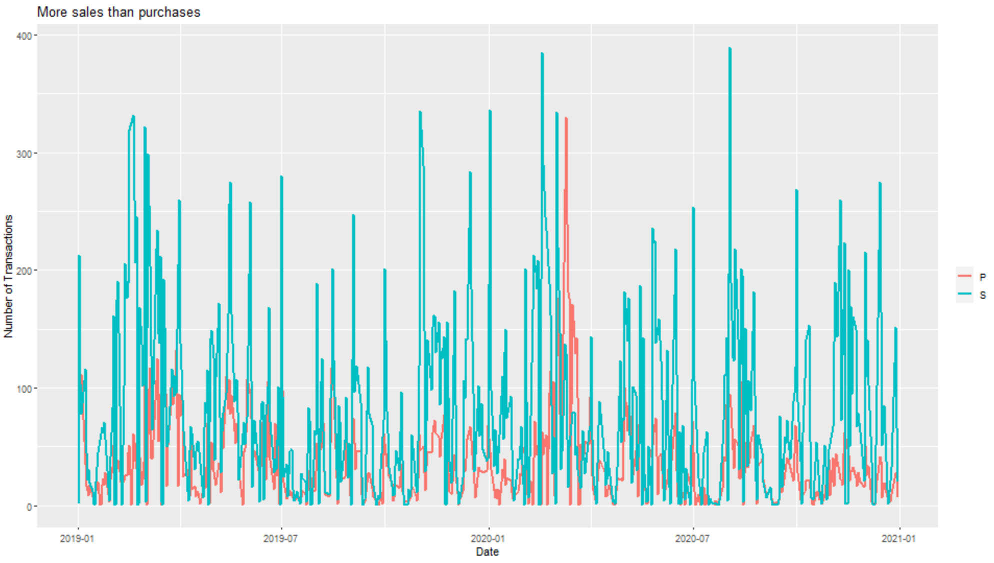


Figure 3: Distribution of sales and purchase transactions

Fig. 4 shows the number of purchase and sales transactions carried out by each role. Overall, the general trend says that the sales transactions supersede purchase transactions for every role there is. The Directors carry out most of the purchase and sales transactions followed by Ten-Percent Owners.

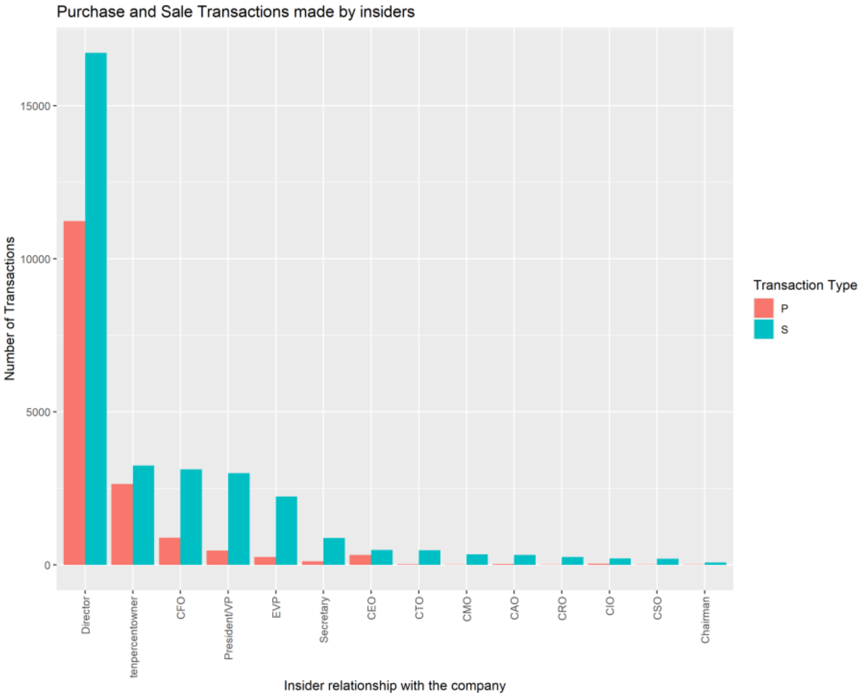


Figure 4: Distribution of sales and purchase transactions per position

Stock Price change after a purchase transaction over time

Fig. 5 shows if the price of a stock that was purchased by an insider a month before, increased or decreased in the years of 2019 and 2020. It is observed that the prices of stocks purchased by insiders usually increase a month after they purchase them. Only a few stocks that the insiders bought seem to decrease in price except during March-April 2020 (COVID-19), which confirms that a lot of stocks decreased in price during that period.

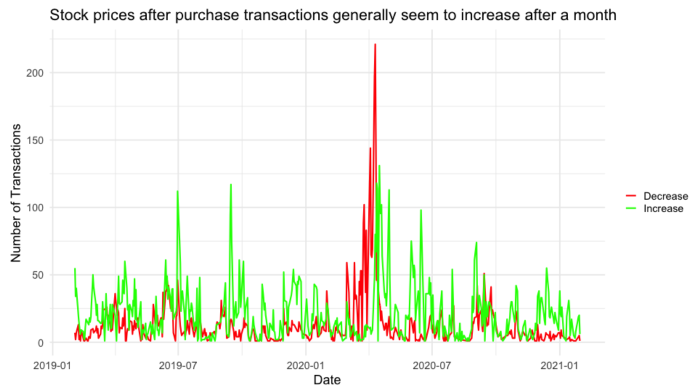


Figure 5: Change in stock prices after one month

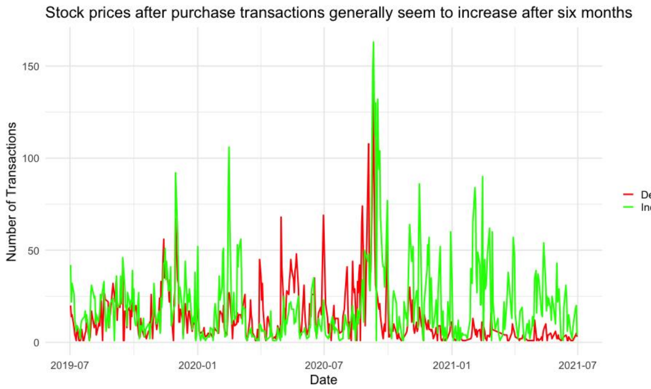


Figure 6: Change in stock prices after six months

Fig. 6 shows if the price of a stock that was purchased by an insider six months before, increased or decreased in the years of 2019 and 2020. Unlike the previous graph, before March 2020, the number of shares that increase and decrease seems to go almost hand in hand. This could be due to the stock prices changing due to external factors and not because an insider bought them. After the COVID-19 period, many stocks increase compared to ones that decrease which are probably because a lot of insiders bought stocks when their prices were low during the dip, and these stocks increased in price after the market stabilized.

Fig. 7 shows how the price of a stock that was sold by an insider a month before changed in price. Much, like the graph for the six-month purchases, there is not a clear distinction between the number of stocks that increase or decrease. This is mostly because the employees are paid in or acquire a lot of their company's stocks and they simply sell for their expenditure and not because they know their company's value is going to go down.

For our analysis, we took the period of one month compared to six months as that seem to have more stability.

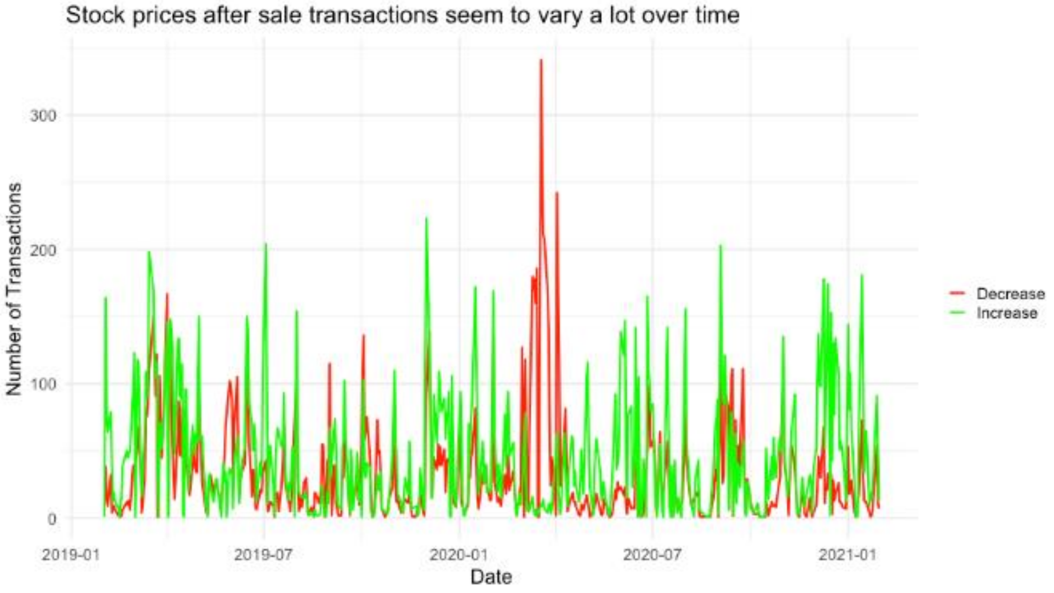


Figure 7: Change in stock prices after a month for sale transactions.

Fig. 8 shows the distribution of change in share prices after one month of purchasing for 6 major insiders namely Director, CEO, CFO, EVP, President/VP, and Ten-Percent Owner. It can be observed that there is a huge spike in the number of transactions for Directors, CFOs, Presidents, and EVPs in the period of early 2020. This huge spike can be attributed to the COVID-19 pandemic wherein insiders purchased the greatest number of shares due to a massive price drop. Ten-Percent Owners often show a normalized trend in purchase transactions throughout the two years.

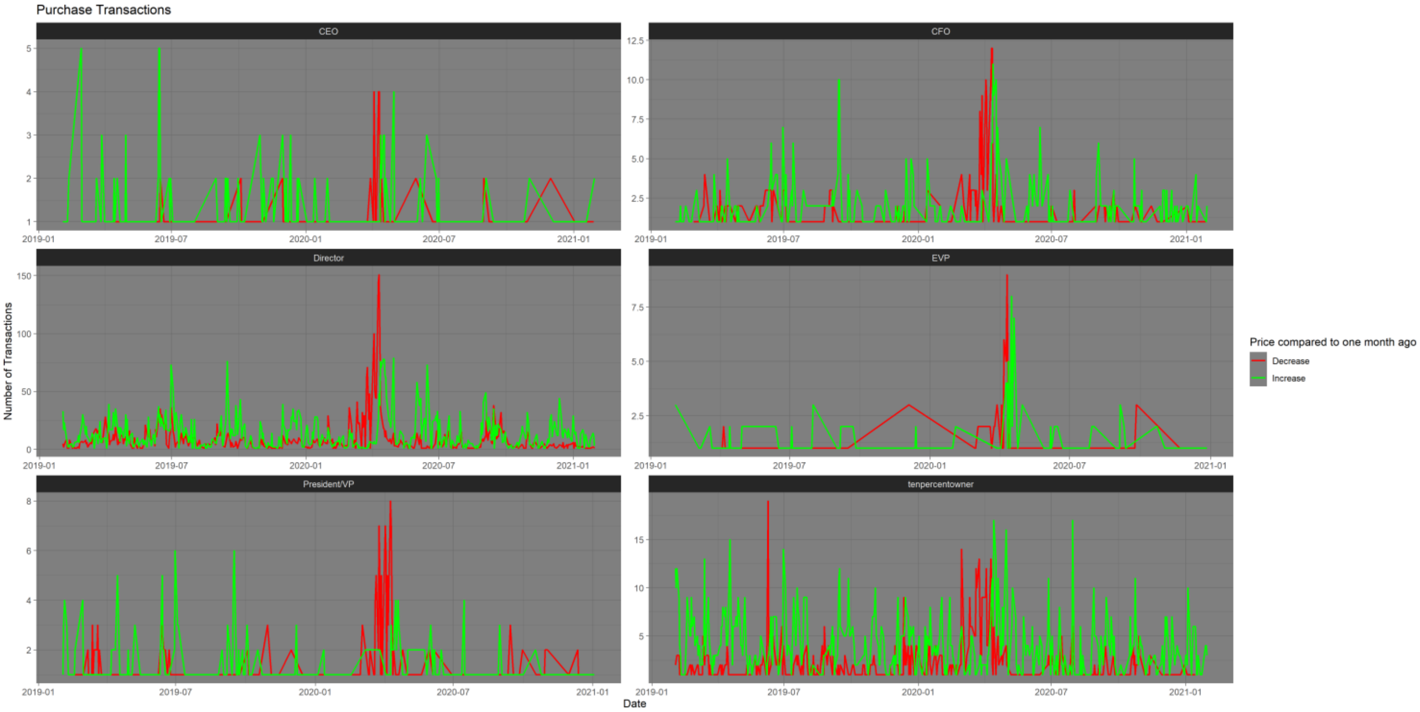


Figure 8: Stock prices after a month of the purchase transactions of CEO, CFO, Directors, EVP, President/VP and 10% owners

Number of shares traded

In Fig. 9, the percent of shares that increase in price is plotted against the number of shares that were involved in a transaction. As we limit the number of transactions to include only those with a certain high amount of shares per transaction, it is found that the percentage of shares that increase greatly goes up. There is a jump from 62% to 68% at 10,000 shares and more and it even goes above 70% at 20,000 shares and more. We take the optimal value of 10,000 shares for a balance between a high increase in the percent of shares that go up and the number of observations for those values.

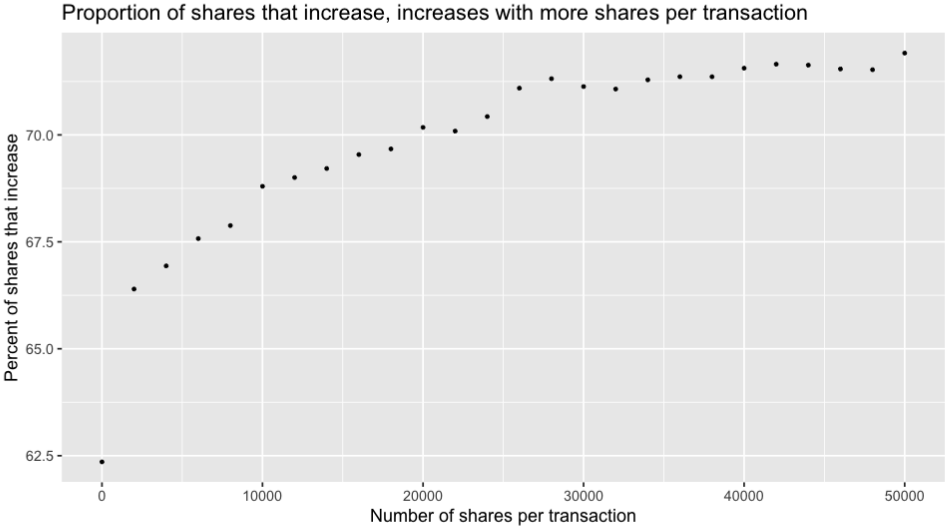


Figure 9: Percentage increase in shares vs number of shares per transaction

Relationship between the positions and the number of shares that increase or decrease

Table 3

Position	#Decrease	#Increase	%Increase	Chi-Square Value	p-value
Overall	1779	3756	67.85	-	-
10PercentOwner	387	1111	74.16	37.057	1.47E-09
CEO	19	56	74.67	1.314	0.2516
CAO	3	4	57.14	0.041	0.8395
CFO	49	125	71.83	1.213	0.2892
Chairman	7	4	36.36	3.674	0.0554
Director	1220	2231	64.65	42.943	5.64E-11
EVP	25	54	68.35	0.003	1
President	35	79	69.3	0.0053	0.8712
Secretary	5	8	61.54	0.036	0.8483

We formulated null and alternative hypotheses which state:

H0: There is no relationship between the position of an individual in a company and the number of share prices that increase or decrease

H1: There is a relationship between the position of an individual in a company and the number of share prices that increase or decrease

We then performed the Chi-squared test of independence for individuals with a transaction of more than 10,000 shares based on Fig. 9. Table 3 was devised to contain results from the test.

#Decrease and #Increase columns give the number of shares whose prices have increased or decreased over one month. %Increase shows the increase in percentage for the share prices after one month. Looking at the chi-squared values for all positions, it is evident Ten-Percent Owners and Directors seem to have the strongest relationship with the increase and decrease of share prices based on the chi-squared values. Additionally, the p-value of Director is the lowest of all positions followed by Ten-Percent Owner which further corroborates the hypothesis that they have a strong impact on the number of share prices that increase or decrease. Thus, we reject the null hypothesis and accept the alternative hypothesis.

Although the Directors and Ten-Percent Owners have the highest chi-squared values, the percentage increase of share prices is the highest for CEOs followed by Ten-Percent Owners and CFOs. This is not strange as there are a lot more observations for Directors and Ten-Percent Owners as compared to other positions. Even with a lesser number of observations, CEOs and CFOs seem to be performing exceptionally well by portraying an increasing trend in the price of shares they invest in.

Hence, it would be recommended to investigate the investing trends of these positions.

Individuals or entities with high profitability

We further filtered out the dataset based on the highest % increase of share prices after a month specific to CEOs, CFOs, Directors, and Ten-Percent Owners. This resulted in a list of individuals/entities shown in table 3 who have the highest profits in purchase transactions over 10,000 shares after a month.

Table 4

Individual/Entity	Role/Designation	Company	No. of Transaction	Profitable after 1 month
Wells Scott	CEO	Clear Channel Outdoor Holdings Inc	3	3
Saba Capital Management	Ten Percent Owner	Western Asset Global High-Income Fund, etc.	168	131
Handok, Inc.	Ten Percent Owner	Rezolute Inc.	36	36
Smith Clarence Edward	Director	ProtoKinetix Inc	27	26
Huckins Scott E.	CFO	Sunopta Inc	5	4

One can follow these entities and their purchase trends to get a better understanding of managing personal investments.

Discussion

From the results, we found that we can use the insider purchase transactions data to trade a company's stocks over a month's timeline. We learned that transactions, which were more than 10,000 shares are traded by CEO, Director, CFO, or 10% owners, are the most profitable.

These results can be used by an individual who wants to or already trades in the open market. Such transactions are an indication that the share prices of the respected company are likely to increase in a month. Also, anyone who's looking to purchase a company's stocks can use the results of our project to make a better-informed decision.

Our results also include a list of a few individuals and entities who are highly profitable. A beginner trader can follow these individuals and entities to trade in the open market.

The project can further analyze insider sales data and its effect on the share prices of their companies.

Statement of contributions

- Nihal Desai: Data Collection (Web Scraping), Preliminary Analysis, General trends analysis and visualization.
- Ramya Kondrakunta: Data Transformation, Analysis on the relationship (role) and change in share prices, visualization.
- Santosh Saranyan: Data Transformation, Analysis on the relationship (role), number of shares and change in share prices, visualization.
- Srinath Iyer: Analysis on transaction size (in dollars) and change in share price, visualization.

For the presentation and report, everyone worked on the portion that they did the analysis and generated visualizations.

References

[1] "EDGAR Application Programming Interfaces." https://www.sec.gov/edgar/sec-api-documentation.

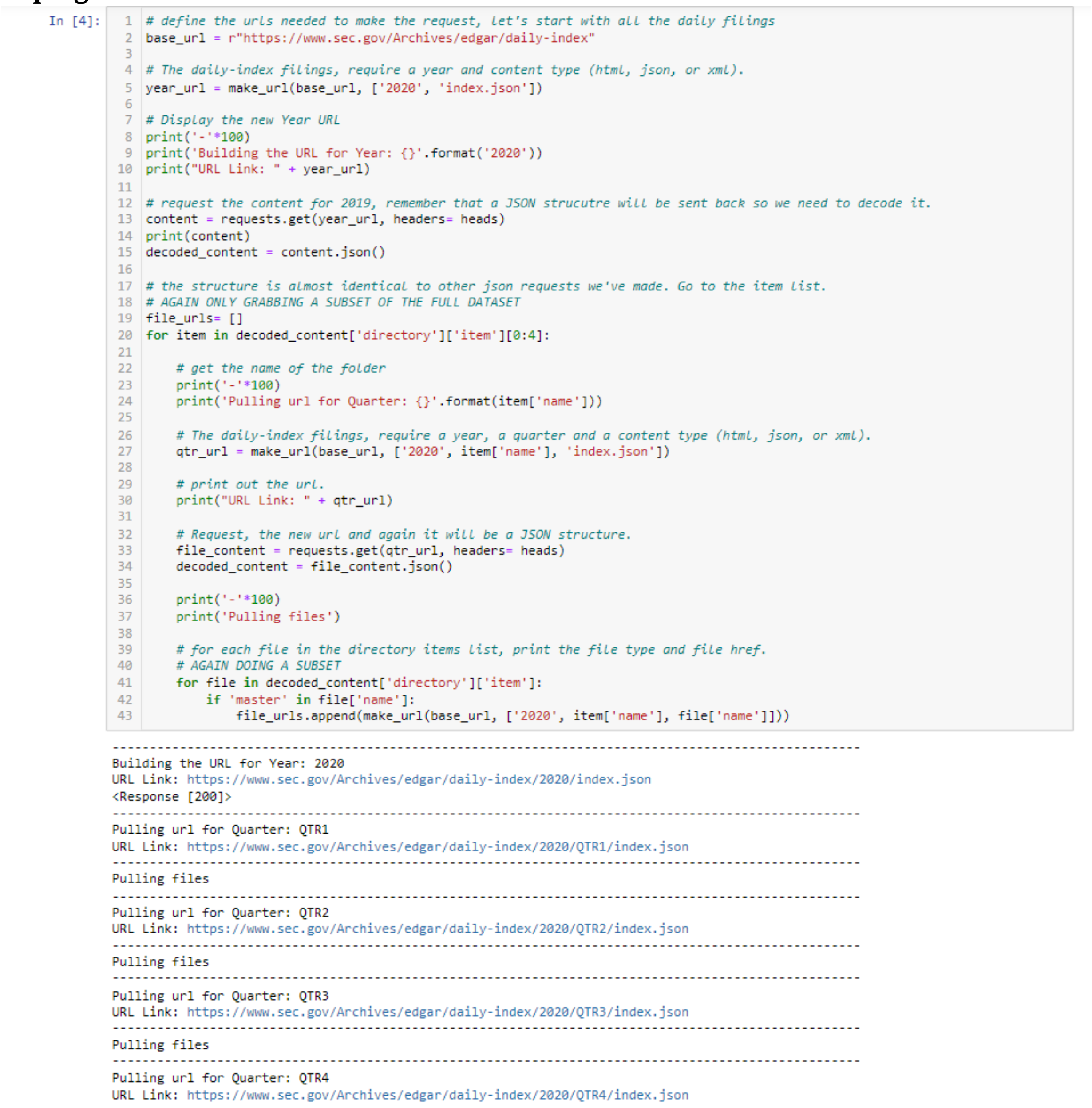
[2] "Yahoo Finance." https://www.sec.gov/edgar/sec-api-documentation.

[3] A. Tamersoy, B. Xie, S. L. Lenkey, B. R. Routledge, D. H. Chau, and S. B. Navathe, "Inside insider trading: Patterns & discoveries from a large scale exploratory analysis," *Proc. 2013 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2013*, pp. 797–804, 2013, doi: 10.1145/2492517.2500288.

[4] A. Tamersoy *et al.*, "Large-scale insider trading analysis: patterns and discoveries," *Soc. Netw. Anal. Min.*, vol. 4, no. 1, pp. 1–17, 2014, doi: 10.1007/s13278-014-0201-9.

Appendix:

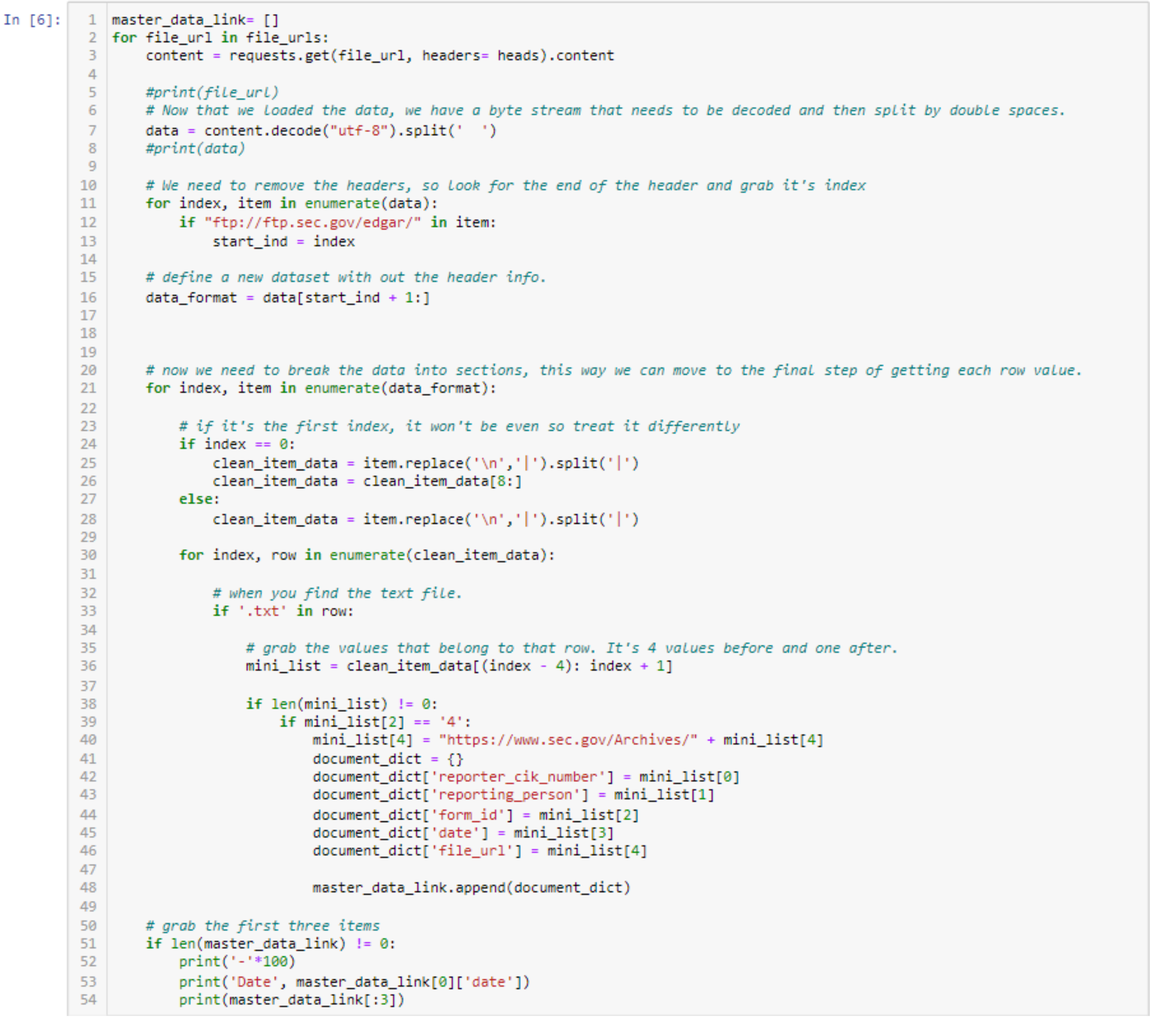
Web scraping Code:



```python
# define the urls needed to make the request, let's start with all the daily filings
base_url = r"https://www.sec.gov/Archives/edgar/daily-index"

# The daily-index filings, require a year and content type (html, json, or xml).
year_url = make_url(base_url, ['2020', 'index.json'])

# Display the new Year URL
print('-'*100)
print('Building the URL for Year: {}'.format('2020'))
print("URL Link: " + year_url)

# request the content for 2019, remember that a JSON strucutre will be sent back so we need to decode it.
content = requests.get(year_url, headers= heads)
print(content)
decoded_content = content.json()

# the structure is almost identical to other json requests we've made. Go to the item list.
# AGAIN ONLY GRABBING A SUBSET OF THE FULL DATASET
file_urls= []
for item in decoded_content['directory']['item'][0:4]:

    # get the name of the folder
    print('-'*100)
    print('Pulling url for Quarter: {}'.format(item['name']))

    # The daily-index filings, require a year, a quarter and a content type (html, json, or xml).
    qtr_url = make_url(base_url, ['2020', item['name'], 'index.json'])

    # print out the url.
    print("URL Link: " + qtr_url)

    # Request, the new url and again it will be a JSON structure.
    file_content = requests.get(qtr_url, headers= heads)
    decoded_content = file_content.json()

    print('-'*100)
    print('Pulling files')

    # for each file in the directory items list, print the file type and file href.
    # AGAIN DOING A SUBSET
    for file in decoded_content['directory']['item']:
        if 'master' in file['name']:
            file_urls.append(make_url(base_url, ['2020', item['name'], file['name']]))
```

```
----------------------------------------------------------------------------------------------------
Building the URL for Year: 2020
URL Link: https://www.sec.gov/Archives/edgar/daily-index/2020/index.json
<Response [200]>
----------------------------------------------------------------------------------------------------
Pulling url for Quarter: QTR1
URL Link: https://www.sec.gov/Archives/edgar/daily-index/2020/QTR1/index.json
----------------------------------------------------------------------------------------------------
Pulling files
----------------------------------------------------------------------------------------------------
Pulling url for Quarter: QTR2
URL Link: https://www.sec.gov/Archives/edgar/daily-index/2020/QTR2/index.json
----------------------------------------------------------------------------------------------------
Pulling files
----------------------------------------------------------------------------------------------------
Pulling url for Quarter: QTR3
URL Link: https://www.sec.gov/Archives/edgar/daily-index/2020/QTR3/index.json
----------------------------------------------------------------------------------------------------
Pulling files
----------------------------------------------------------------------------------------------------
Pulling url for Quarter: QTR4
URL Link: https://www.sec.gov/Archives/edgar/daily-index/2020/QTR4/index.json
```

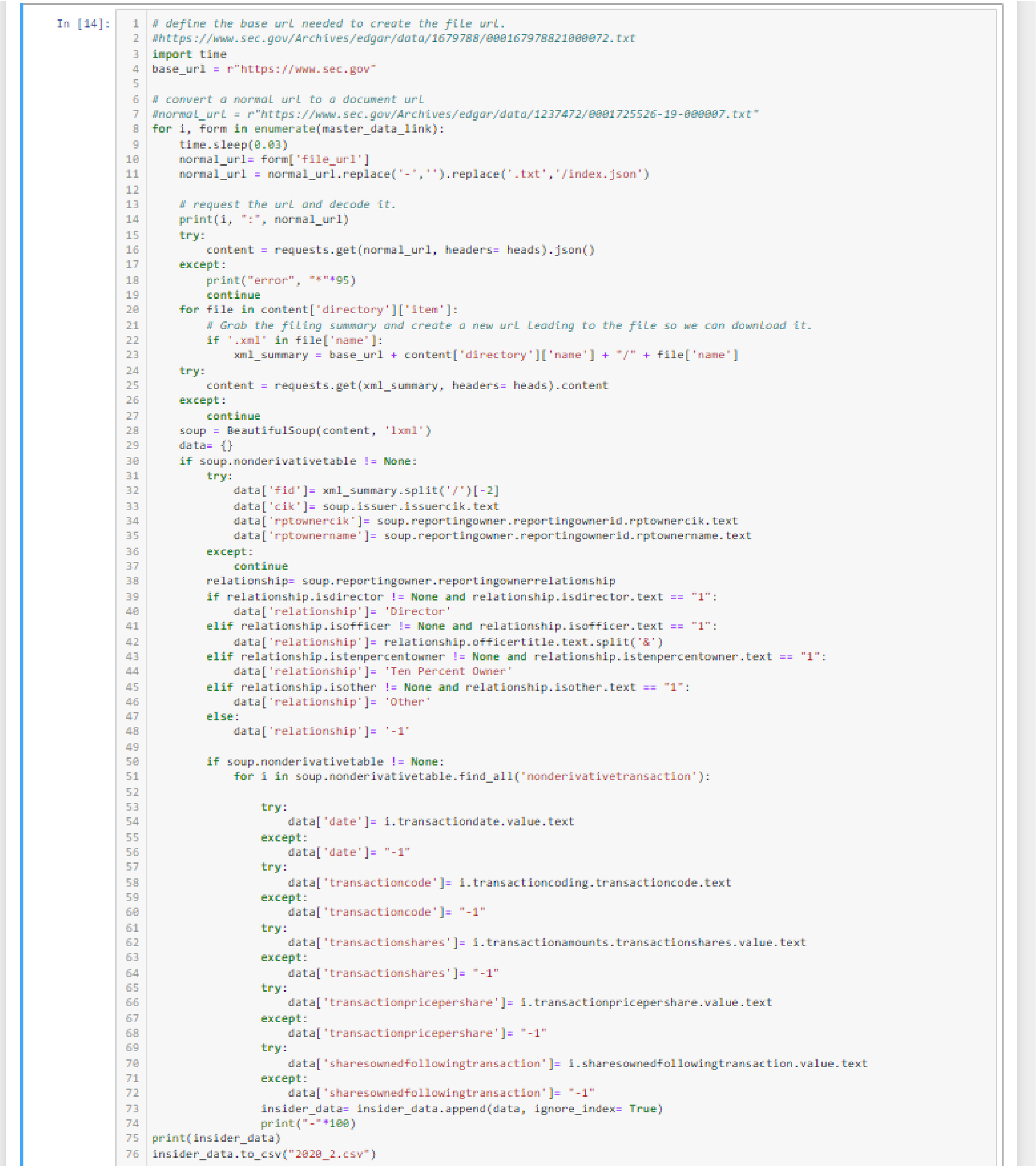
Figure 10: Information of all the filings in a given Quarter



In [6]:

```python
master_data_link= []
for file_url in file_urls:
    content = requests.get(file_url, headers= heads).content

    #print(file_url)
    # Now that we loaded the data, we have a byte stream that needs to be decoded and then split by double spaces.
    data = content.decode("utf-8").split('  ')
    #print(data)

    # We need to remove the headers, so look for the end of the header and grab it's index
    for index, item in enumerate(data):
        if "ftp://ftp.sec.gov/edgar/" in item:
            start_ind = index

    # define a new dataset with out the header info.
    data_format = data[start_ind + 1:]



    # now we need to break the data into sections, this way we can move to the final step of getting each row value.
    for index, item in enumerate(data_format):

        # if it's the first index, it won't be even so treat it differently
        if index == 0:
            clean_item_data = item.replace('\n','|').split('|')
            clean_item_data = clean_item_data[8:]
        else:
            clean_item_data = item.replace('\n','|').split('|')

        for index, row in enumerate(clean_item_data):

            # when you find the text file.
            if '.txt' in row:

                # grab the values that belong to that row. It's 4 values before and one after.
                mini_list = clean_item_data[(index - 4): index + 1]

                if len(mini_list) != 0:
                    if mini_list[2] == '4':
                        mini_list[4] = "https://www.sec.gov/Archives/" + mini_list[4]
                        document_dict = {}
                        document_dict['reporter_cik_number'] = mini_list[0]
                        document_dict['reporting_person'] = mini_list[1]
                        document_dict['form_id'] = mini_list[2]
                        document_dict['date'] = mini_list[3]
                        document_dict['file_url'] = mini_list[4]

                        master_data_link.append(document_dict)

    # grab the first three items
    if len(master_data_link) != 0:
        print('-'*100)
        print('Date', master_data_link[0]['date'])
        print(master_data_link[:3])
```

Figure 11: Filter and store information of only Form 4



In [14]:

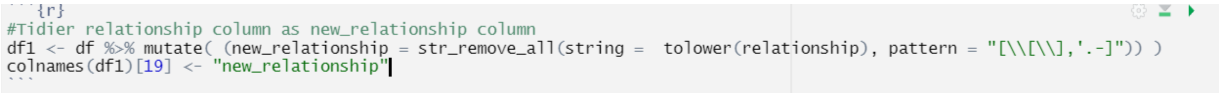
```python
# define the base url needed to create the file url.
#https://www.sec.gov/Archives/edgar/data/1679788/000167978821000072.txt
import time
base_url = r"https://www.sec.gov"

# convert a normal url to a document url
#normal_url = r"https://www.sec.gov/Archives/edgar/data/1237472/0001725526-19-000007.txt"
for i, form in enumerate(master_data_link):
    time.sleep(0.03)
    normal_url= form['file_url']
    normal_url = normal_url.replace('-','').replace('.txt','/index.json')

    # request the url and decode it.
    print(i, ":", normal_url)
    try:
        content = requests.get(normal_url, headers= heads).json()
    except:
        print("error", "*"*95)
        continue
    for file in content['directory']['item']:
        # Grab the filing summary and create a new url leading to the file so we can download it.
        if '.xml' in file['name']:
            xml_summary = base_url + content['directory']['name'] + "/" + file['name']
    try:
        content = requests.get(xml_summary, headers= heads).content
    except:
        continue
    soup = BeautifulSoup(content, 'lxml')
    data= {}
    if soup.nonderivativetable != None:
        try:
            data['fid']= xml_summary.split('/')[-2]
            data['cik']= soup.issuer.issuercik.text
            data['rptownercik']= soup.reportingowner.reportingownerid.rptownercik.text
            data['rptownername']= soup.reportingowner.reportingownerid.rptownername.text
        except:
            continue
        relationship= soup.reportingowner.reportingownerrelationship
        if relationship.isdirector != None and relationship.isdirector.text == "1":
            data['relationship']= 'Director'
        elif relationship.isofficer != None and relationship.isofficer.text == "1":
            data['relationship']= relationship.officertitle.text.split('&')
        elif relationship.istenpercentowner != None and relationship.istenpercentowner.text == "1":
            data['relationship']= 'Ten Percent Owner'
        elif relationship.isother != None and relationship.isother.text == "1":
            data['relationship']= 'Other'
        else:
            data['relationship']= '-1'

        if soup.nonderivativetable != None:
            for i in soup.nonderivativetable.find_all('nonderivativetransaction'):

                try:
                    data['date']= i.transactiondate.value.text
                except:
                    data['date']= "-1"
                try:
                    data['transactioncode']= i.transactioncoding.transactioncode.text
                except:
                    data['transactioncode']= "-1"
                try:
                    data['transactionshares']= i.transactionamounts.transactionshares.value.text
                except:
                    data['transactionshares']= "-1"
                try:
                    data['transactionpricepershare']= i.transactionpricepershare.value.text
                except:
                    data['transactionpricepershare']= "-1"
                try:
                    data['sharesownedfollowingtransaction']= i.sharesownedfollowingtransaction.value.text
                except:
                    data['sharesownedfollowingtransaction']= "-1"
                insider_data= insider_data.append(data, ignore_index= True)
                print("-"*100)
print(insider_data)
insider_data.to_csv("2020_2.csv")
```

Figure 12: Store required information from each Form 4

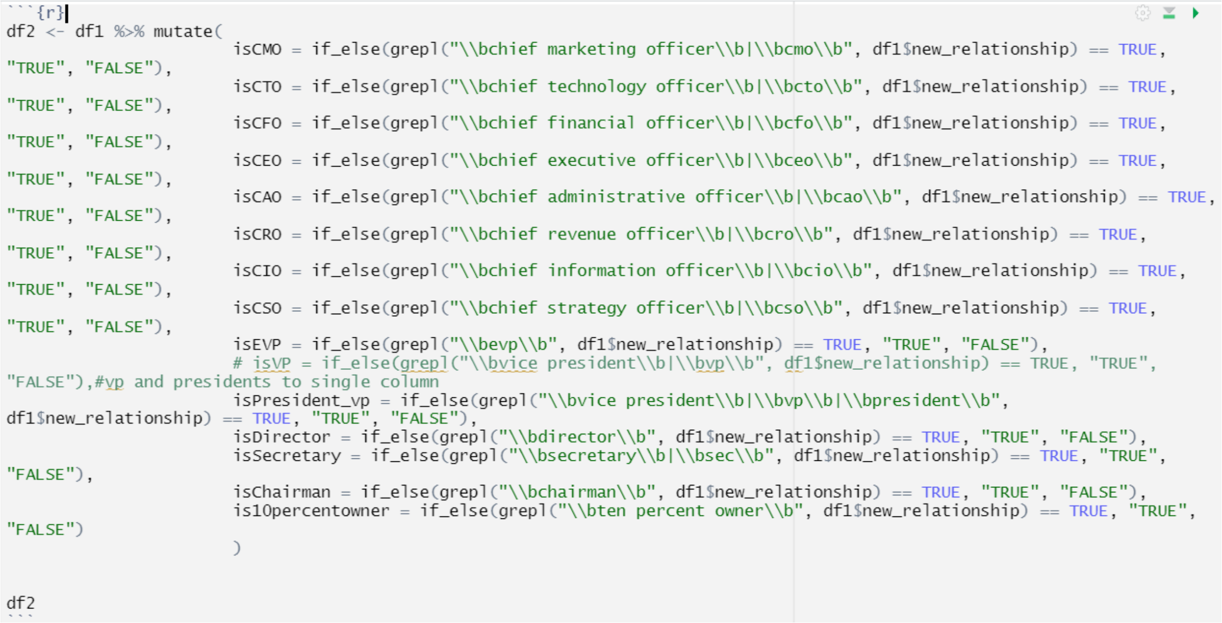
Fig 10 Collect share prices after the transaction

Data Transformation and Tidying

Code:

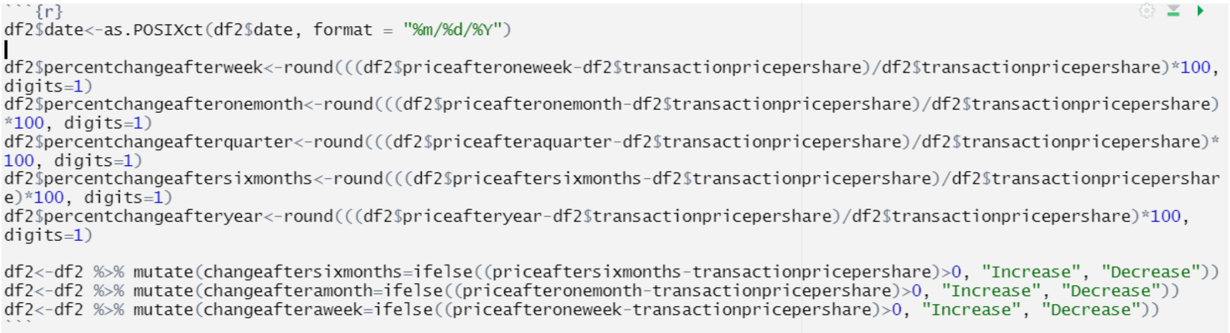


~~~r
```{r}
#Tidier relationship column as new_relationship column
df1 <- df %>% mutate( (new_relationship = str_remove_all(string =  tolower(relationship), pattern = "[\\[\\],'.-]")) )
colnames(df1)[19] <- "new_relationship"
~~~
```

Figure 13: Tidying the relationship attribute and creating new_relationship attribute
```



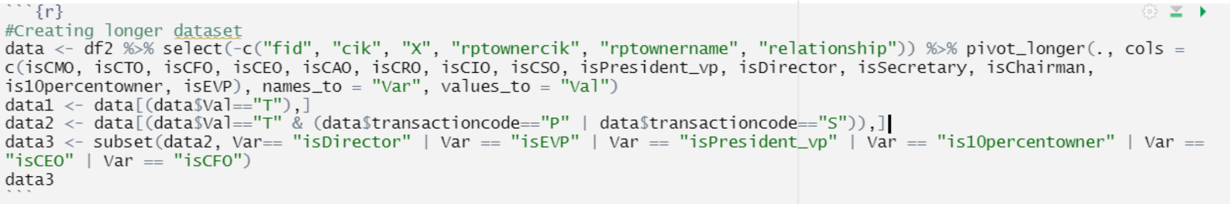
~~~
```{r}
df2 <- df1 %>% mutate(
                       isCMO = if_else(grepl("\\bchief marketing officer\\b|\\bcmo\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isCTO = if_else(grepl("\\bchief technology officer\\b|\\bcto\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isCFO = if_else(grepl("\\bchief financial officer\\b|\\bcfo\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isCEO = if_else(grepl("\\bchief executive officer\\b|\\bceo\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isCAO = if_else(grepl("\\bchief administrative officer\\b|\\bcao\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isCRO = if_else(grepl("\\bchief revenue officer\\b|\\bcro\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isCIO = if_else(grepl("\\bchief information officer\\b|\\bcio\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isCSO = if_else(grepl("\\bchief strategy officer\\b|\\bcso\\b", df1$new_relationship) == TRUE,
"TRUE", "FALSE"),
                       isEVP = if_else(grepl("\\bevp\\b", df1$new_relationship) == TRUE, "TRUE", "FALSE"),
                       # isVP = if_else(grepl("\\bvice president\\b|\\bvp\\b", df1$new_relationship) == TRUE, "TRUE",
"FALSE"),#vp and presidents to single column
                       isPresident_vp = if_else(grepl("\\bvice president\\b|\\bvp\\b|\\bpresident\\b",
df1$new_relationship) == TRUE, "TRUE", "FALSE"),
                       isDirector = if_else(grepl("\\bdirector\\b", df1$new_relationship) == TRUE, "TRUE", "FALSE"),
                       isSecretary = if_else(grepl("\\bsecretary\\b|\\bsec\\b", df1$new_relationship) == TRUE, "TRUE",
"FALSE"),
                       isChairman = if_else(grepl("\\bchairman\\b", df1$new_relationship) == TRUE, "TRUE", "FALSE"),
                       is10percentowner = if_else(grepl("\\bten percent owner\\b", df1$new_relationship) == TRUE, "TRUE",
"FALSE")
                       )


df2
```
~~~

Figure 14: Creating various insider position attributes to indicate the report owner's relationship with the company



~~~
```{r}
df2$date<-as.POSIXct(df2$date, format = "%m/%d/%Y")

df2$percentchangeafterweek<-round(((df2$priceafteroneweek-df2$transactionpricepershare)/df2$transactionpricepershare)*100,
digits=1)
df2$percentchangeafteronemonth<-round(((df2$priceafteronemonth-df2$transactionpricepershare)/df2$transactionpricepershare)
*100, digits=1)
df2$percentchangeafterquarter<-round(((df2$priceafteraquarter-df2$transactionpricepershare)/df2$transactionpricepershare)*
100, digits=1)
df2$percentchangeaftersixmonths<-round(((df2$priceaftersixmonths-df2$transactionpricepershare)/df2$transactionpricepershar
e)*100, digits=1)
df2$percentchangeafteryear<-round(((df2$priceafteryear-df2$transactionpricepershare)/df2$transactionpricepershare)*100,
digits=1)

df2<-df2 %>% mutate(changeaftersixmonths=ifelse((priceaftersixmonths-transactionpricepershare)>0, "Increase", "Decrease"))
df2<-df2 %>% mutate(changeafteramonth=ifelse((priceafteronemonth-transactionpricepershare)>0, "Increase", "Decrease"))
df2<-df2 %>% mutate(changeafteraweek=ifelse((priceafteroneweek-transactionpricepershare)>0, "Increase", "Decrease"))
```
~~~

Figure 15: Handling date attribute. Creating percentage change and change (in stock price at different intervals) attributes



~~~
```{r}
#Creating longer dataset
data <- df2 %>% select(-c("fid", "cik", "X", "rptownercik", "rptownername", "relationship")) %>% pivot_longer(., cols =
c(isCMO, isCTO, isCFO, isCEO, isCAO, isCRO, isCIO, isCSO, isPresident_vp, isDirector, isSecretary, isChairman,
is10percentowner, isEVP), names_to = "Var", values_to = "Val")
data1 <- data[(data$Val=="T"),]
data2 <- data[(data$Val=="T" & (data$transactioncode=="P" | data$transactioncode=="S")),]
data3 <- subset(data2, Var== "isDirector" | Var == "isEVP" | Var == "isPresident_vp" | Var == "is10percentowner" | Var ==
"isCEO" | Var == "isCFO")
data3
```
~~~

Figure 16: Transforming dataset to be longer w.r.t the insider position attributes for visualizations.