BA 870 – Group Project

Financial and Accounting Analytics of Stock Returns Around Covid-19 Market Shock

Shangkun(Sherry) Zuo, szuo@bu.edu;
Yuqi(Yoki) Liu, yokiliu@bu.edu;
Yanni Lan, lanyanni@bu.edu;
Jiayuan Zou, jyzou@bu.edu;
Ziyan Pei, ziyanpei@bu.edu;
Siqi Zhang, zhangsq@bu.edu

Cohort B
April 19, 2020

2019 Financial Accounting Characteristics that Explain Varying Exposures to Covid-19 Market Shock

Upload data to Pandas Dataframes

Import 2019 Data

```
from google.colab import files
uploaded = files.upload()
```

Choose Files 2019_ratios

- **2019_ratios.csv**(text/csv) - 178969 bytes, last modified: 4/17/2020 - 100% done

```
Saving 2019_ratios.csv to 2019_ratios.csv
```

```
import pandas as pd
import numpy as np
df = pd.read_csv('2019_ratios.csv')
pd.DataFrame.from_records(df)
df.head()
```

	Global Company Key	Data Date	Data Year - Fiscal	Ticker Symbol	Company Name	Current Assets - Total	Assets - Total	Cash	Cash and Short-Term Investments	Cost of Goods Sold	Long-Term Debt - Total	Earnings Before Interest and Taxes
0	1045	20191231	2019	AAL	AMERICAN AIRLINES GROUP INC	8206.00	59995.000	280.000	3984.000	32027.00	28875.00	3706.00
1	1075	20191231	2019	PNW	PINNACLE WEST CAPITAL CORP	1030.03	18479.247	10.283	10.283	2208.32	4884.43	671.96
2	1078	20191231	2019	ABT	ABBOTT LABORATORIES	15667.00	67887.000	3860.000	4140.000	11949.00	17416.00	5041.00
3	1161	20191231	2019	AMD	ADVANCED MICRO DEVICES	4597.00	6028.000	1466.000	1507.000	3721.00	685.00	583.00
4	1209	20190930	2019	APD	AIR PRODUCTS & CHEMICALS INC	4618.30	18942.800	2248.700	2414.700	4892.70	3227.40	2120.50

```
df.shape
```

```
(1217, 21)
```

Import 2020 Data

```
from google.colab import files
uploaded = files.upload()
```

Choose Files BA870 Project...ta 2020.xlsx

- **BA870 Project Data 2020.xlsx**(application/vnd.openxmlformats-officedocument.spreadsheetml.sheet) - 340388 bytes, last modified: 4/14/2020 - 100% done

```
Saving BA870 Project Data 2020.xlsx to BA870 Project Data 2020.xlsx
```

```
df2 = pd.read_excel('BA870 Project Data 2020.xlsx')
df2.head()
```

	tic	conm_x	Last Price	YTD Ret	1-mth Ret	3-mth Ret	12-mth Ret	BothSP-NASDAQ	SP500	Ticker	gvkey	datadate	fyear	cusip
0	A	Agilent Technologies	70.42	-0.1745	-0.1233	-0.1673	-0.1406	0	1	A	126554.0	20191031.0	2019.0	00846U101
1	AAL	American Airlines Gp	9.39	-0.6726	-0.4739	-0.6604	-0.7214	1	1	AAL	1045.0	20191231.0	2019.0	02376R102
2	AAON	Aaon Inc	44.91	-0.0911	-0.2006	-0.1080	-0.0194	0	0	AAON	21542.0	20191231.0	2019.0	360206
3	AAP	Advance Auto Parts Inc	84.65	-0.4715	-0.3505	-0.4690	-0.5070	0	1	AAP	145977.0	20191231.0	2019.0	00751Y106
4	AAPL	Apple Inc	241.41	-0.1779	-0.1656	-0.1883	0.2358	1	1	AAPL	1690.0	20190930.0	2019.0	37833100

```
#only leave those columns we will use for analysis
df3=df2[['tic','YTD Ret']]
#rename columns
df3=df3.rename(columns={"tic": "Ticker", "YTD Ret":"YTD"})
df3.head()
```

	Ticker	YTD
0	A	-0.1745
1	AAL	-0.6726
2	AAON	-0.0911
3	AAP	-0.4715
4	AAPL	-0.1779

Data Cleaning

```
import warnings
warnings.filterwarnings('ignore')
```

In this section, we will have a detailed data cleaning for the general dataset, as well as seperate into 9 industrial catergories.

Remove unnecessary variables and rename the columns

```
# Remove unnecessary data columns
cols_remove = [0,1,2,4]
df.drop(df.columns[cols_remove],axis=1, inplace=True)
# Rename variables
renamed_columns = ["Ticker", "CA", "TA", "Cash", "CHE", "COGS", "Long_Term_Debt", "EBIT", "Employees", "Inventory", "
df.columns = renamed_columns
df.head()
```

	Ticker	CA	TA	Cash	CHE	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI
0	AAL	8206.00	59995.000	280.000	3984.000	32027.00	28875.00	3706.00	133.70	1851.00	18311.000	1686.00
1	PNW	1030.03	18479.247	10.283	10.283	2208.32	4884.43	671.96	6.21	345.92	2078.365	538.32
2	ABT	15667.00	67887.000	3860.000	4140.000	11949.00	17416.00	5041.00	107.00	4316.00	10863.000	3687.00
3	AMD	4597.00	6028.000	1466.000	1507.000	3721.00	685.00	583.00	11.40	982.00	2359.000	341.00
4	APD	4618.30	18942.800	2248.700	2414.700	4892.70	3227.40	2120.50	17.70	388.30	1820.900	1760.00

Rearrange Industry Calssification

```
#check frequency count for SIC
#df['SIC'].value_counts().to_dict()
#could uncomment the code line below to view
```

We want to classify these companies to different industries, which could give us specific overviews for the different industries. The first two digits in range 01-09 is Ariculture, Foresty, and Fishing. However, since there is only 1 row which the SIC is 100 (also is the 0100) in that range, we decide to delete that row since it could not stand for an entire industry.

```
df.drop(df[df["SIC"]==100].index, inplace=True)
```

```
# Change 4-digit SIC code to 2-digit SIC code
df['SIC'] = (df['SIC']).astype(str).str[:2]
```

```
#transfer SIC from string to integer
df['SIC'] = df['SIC'].astype(int)
```

```
df.head()
```

	Ticker	CA	TA	Cash	CHE	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI
0	AAL	8206.00	59995.000	280.000	3984.000	32027.00	28875.00	3706.00	133.70	1851.00	18311.000	1686.00
1	PNW	1030.03	18479.247	10.283	10.283	2208.32	4884.43	671.96	6.21	345.92	2078.365	538.32
2	ABT	15667.00	67887.000	3860.000	4140.000	11949.00	17416.00	5041.00	107.00	4316.00	10863.000	3687.00
3	AMD	4597.00	6028.000	1466.000	1507.000	3721.00	685.00	583.00	11.40	982.00	2359.000	341.00
4	APD	4618.30	18942.800	2248.700	2414.700	4892.70	3227.40	2120.50	17.70	388.30	1820.900	1760.00

SIC Code Classification

> 01-09: Agriculture, Forestry, and Fishing (ignore since only 1 record)
> 10-14: Mining (--> group 1)
> 15-17: Construction (--> group 2)
> 20-39: Manufacturing (--> group 3)
> 40-49: Transportation, Communications, Electric, Gas, and Sanitary Services (--> group 4)
> 50-51: Wholesale Trade (--> group 5)
> 52-59: Retail Trade (--> group 6)
> 60-67: Finance, Insurance, and Real Estate (--> group 7)
> 70-89: Services (--> group 8)
> 90-99: Public Administration (--> group 9)

```
df['Group'] = pd.cut(df['SIC'], [10,14,17,39,49,51,59,67,89,99], labels=["1","2","3","4","5","6","7","8","9"], includ
df.head()
```

	Ticker	CA	TA	Cash	CHE	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI
0	AAL	8206.00	59995.000	280.000	3984.000	32027.00	28875.00	3706.00	133.70	1851.00	18311.000	1686.00
1	PNW	1030.03	18479.247	10.283	10.283	2208.32	4884.43	671.96	6.21	345.92	2078.365	538.32
2	ABT	15667.00	67887.000	3860.000	4140.000	11949.00	17416.00	5041.00	107.00	4316.00	10863.000	3687.00
3	AMD	4597.00	6028.000	1466.000	1507.000	3721.00	685.00	583.00	11.40	982.00	2359.000	341.00
4	APD	4618.30	18942.800	2248.700	2414.700	4892.70	3227.40	2120.50	17.70	388.30	1820.900	1760.00

```
#check frequency count for Group this time
df['Group'].value_counts().to_dict()
```

```
{'1': 33,
 '2': 10,
 '3': 478,
 '4': 108,
 '5': 23,
 '6': 62,
 '7': 273,
 '8': 225,
 '9': 4}
```

From our observation, most of companines are in the `Manufacturing`, `Finance, Insurance and Real Estate` and `Service` group.

Check missing values

```
#check any NA in each variable
df.isna().any()
```

```
Ticker              False
CA                   True
TA                  False
Cash                 True
CHE                 False
COGS                False
Long_Term_Debt       True
EBIT                False
Employees            True
Inventory            True
CL                   True
NI                  False
PP&E                 True
Sales               False
Interest_Expense     True
Zip                  True
SIC                 False
Group               False
dtype: bool
```

We notice that there is a great amount NaNs in the `Interest_Expense`. Since other information lacks, we assume that some companies have great operation so they don't have that much debt. We decide to replace the NAs with 0.

```
df['Interest_Expense'].fillna(0, inplace=True)
```

Replace all missing values in `Cash` with alternative variables in WRDS `CHE(Cash and Short-Term Investments)`. For example, in banking industry they don't classify their liquity asset as Cash.

```
df.Cash.fillna(df.CHE,inplace=True)
```

```
#remove CHE column since we want to focus on Cash
del df['CHE']
```

```
#counting missing values
df.isna().sum()
```

```
Ticker                0
CA                  250
TA                    0
Cash                  0
COGS                  0
Long_Term_Debt        7
EBIT                  0
Employees            33
Inventory            19
CL                  250
NI                    0
PP&E                 30
Sales                 0
Interest_Expense      0
Zip                   6
SIC                   0
Group                 0
dtype: int64
```

Subset Dataset based on `Group`

```
mining=df[df.Group == '1']
```

```
construction=df[df.Group == '2']
manufacturing=df[df.Group == '3']
transportation=df[df.Group == '4']
wholesale=df[df.Group == '5']
retail=df[df.Group == '6']
finance=df[df.Group == '7']
service=df[df.Group == '8']
administration=df[df.Group == '9']
```

In the next following step, we are going to look deeper into other missing values and replace the values with reasonable explanations.

Replace Missing Values

Mining Industry

```
#counting missing values
mining.isna().sum()
```

```
Ticker              0
CA                  0
TA                  0
Cash                0
COGS                0
Long_Term_Debt      0
EBIT                0
Employees           0
Inventory           1
CL                  0
NI                  0
PP&E                0
Sales               0
Interest_Expense    0
Zip                 0
SIC                 0
Group               0
dtype: int64
```

```
#replace missing value with column median
mining.fillna(mining.median(), inplace=True)
```

Construction Industry

```
#counting missing values
construction.isna().sum()
```

```
Ticker              0
CA                  5
TA                  0
Cash                0
COGS                0
Long_Term_Debt      0
EBIT                0
Employees           0
Inventory           1
CL                  5
NI                  0
PP&E                0
Sales               0
Interest_Expense    0
Zip                 0
SIC                 0
Group               0
dtype: int64
```

```
construction[construction['Inventory'].isnull()]
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&E
144	J	4111.768	11462.711	631.068	10086.602	1201.245	760.086	52.0	NaN	3073.706	847.184	308.143

The company that missing the inventory record is `Jacobs Engineering Group Inc`. It is an American international technical professional services firm. The company provides technical, professional and construction services, so this company does not have inventory, thus we replace with 0.

```
construction['Inventory'].fillna(0, inplace=True)
```

```
construction['Inventory'].fillna(0, inplace=True)
```

```
#replace missing value with column median
construction.fillna(construction.median(), inplace=True)
```

Manufacturing Industry

```
#counting missing values
manufacturing.isna().sum()
```

```
Ticker              0
CA                  3
TA                  0
Cash                0
COGS                0
Long_Term_Debt      4
EBIT                0
Employees           7
Inventory           1
CL                  3
NI                  0
PP&E                0
Sales               0
Interest_Expense    0
Zip                 4
SIC                 0
Group               0
dtype: int64
```

```
manufacturing[manufacturing['Zip'].isnull()]
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI
135	TT	6217.200	20492.300	1303.600	10967.900	5317.300	2235.500	50.0	1712.200	4861.900	1410.900
146	JCI	12393.000	42287.000	2805.000	15450.000	6708.000	1223.000	104.0	1814.000	9070.000	5674.000
1071	CMPR	240.300	1868.376	35.279	1280.829	1070.422	184.911	13.0	66.310	520.749	95.052
1073	SIMO	565.199	697.729	323.166	217.806	0.000	73.444	NaN	88.439	125.763	64.399

After reviewing the company information, `TT (TRANE TECHNOLOGIES PLC)` and `JCI(JOHNSON CONTROLS INTL PLC)` are both in Ireland, so they do not have Zip code, thus replace with '00000', this does not affect that much because we just don't want to remove records for our ratios analysis, but we will remove foreign countires for our geo analysis; company `CMPR(CIMPRESS PLC)` zip code is 02451, company `SIMO (SILICON MOTION TECH)` zip code is 95131.

```
manufacturing.loc[manufacturing['Ticker'] == "CMPR", 'Zip'] = "02451"
manufacturing.loc[manufacturing['Ticker'] == "SIMO", 'Zip'] = "95131"
manufacturing.loc[manufacturing['Ticker'] == "TT", 'Zip'] = "00000"
manufacturing.loc[manufacturing['Ticker'] == "JCI", 'Zip'] = "00000"
```

```
manufacturing[manufacturing['Inventory'].isnull()]
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&E	Sal
472	SAGE	1035.093	1084.15	126.705	712.709	26.848	-708.124	0.675	NaN	112.128	-680.238	42.897	6.8

SAGE(SAGE THERAPEUTICS INC) is a biopharmaceutical company, so the company should have inventory, thus we will replace with column median.

```
#replace missing value with column median
manufacturing.fillna(manufacturing.median(), inplace=True)
```

Transportation Industry

```
#counting missing values
transportation.isna().sum()
```

```
Ticker              0
CA                  0
TA                  0
Cash                0
COGS                0
Long_Term_Debt      0
EBIT                0
Employees           2
Inventory           3
CL                  0
NI                  0
PP&E                1
Sales               0
Interest_Expense    0
Zip                 0
SIC                 0
Group               0
dtype: int64
```

```
#replace missing value with column median
transportation.fillna(transportation.median(), inplace=True)
```

Wholesale Trade Industry

```
#counting missing values
wholesale.isna().sum()
```

```
Ticker              0
CA                  1
TA                  0
Cash                0
COGS                0
Long_Term_Debt      0
EBIT                0
Employees           0
Inventory           0
CL                  1
NI                  0
PP&E                0
Sales               0
Interest_Expense    0
Zip                 0
SIC                 0
Group               0
dtype: int64
```

```
#replace missing value with column median
wholesale.fillna(wholesale.median(), inplace=True)
```

Retail Trade Industry

```
#counting missing values
retail.isna().sum()
```

```
Ticker              0
CA                  0
TA                  0
Cash                0
COGS                0
Long_Term_Debt      0
EBIT                0
Employees           3
Inventory           2
CL                  0
NI                  0
PP&E                0
Sales               0
Interest_Expense    0
Zip                 0
SIC                 0
Group               0
dtype: int64
```

```
#replace missing value with column median
retail.fillna(retail.median(), inplace=True)
```

Finance Industry

We try to download the dataset by set up Industry Format with `FS` for finance industry, but there is no difference with previous data. We also try to download `CHS`, but it's empty.

```
#counting missing values
finance.isna().sum()
```

```
Ticker              0
CA                231
TA                  0
Cash                0
COGS                0
Long_Term_Debt      1
EBIT                0
Employees           8
Inventory           7
CL                231
NI                  0
PP&E               29
Sales               0
Interest_Expense    0
Zip                 0
SIC                 0
Group               0
dtype: int64
```

```
#replace missing value with column median
finance.fillna(finance.median(), inplace=True)
```

Important Notice: We replace the NaNs with industry median because we didn't find other better solution to replace the missing values. We also tried the method that professor suggest on the Q&A, but with the limiation on total assets, sales, and sic, WRDS returns no matching data.

Service Industry

```
#counting missing values
service.isna().sum()
```

```
Ticker              0
CA                  7
TA                  0
Cash                0
COGS                0
Long_Term_Debt      2
EBIT                0
Employees          13
Inventory           4
CL                  7
NI                  0
PP&E                0
Sales               0
Interest_Expense    0
Zip                 2
SIC                 0
Group               0
dtype: int64
```

```
service[service['Zip'].isnull()]
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&E
778	AFYA	276.584	724.557	234.651	63.198	66.815	51.051	NaN	0.978	82.90	38.291	102.894
1119	MLCO	1894.672	9488.422	1394.982	3637.915	4744.284	761.068	23.078	43.959	1525.46	373.173	5834.952

After reviewing the company information, AFYA(AFYA LTD) is in Brazil, MLCO(MELCO RESORTS & ENTERTAINMEN) is in Hongkong, so they do not have Zip code, thus replace with '00000'.

```
service.loc[service['Ticker'] == "AFYA", 'Zip'] = "00000"
service.loc[service['Ticker'] == "MLCO", 'Zip'] = "00000"
```

```
#replace missing value with column median
service.fillna(service.median(), inplace=True)
```

Public Administration Industry

```
#counting missing values
administration.isna().sum()
```

```
Ticker              0
CA                  3
TA                  0
Cash                0
COGS                0
Long_Term_Debt      0
EBIT                0
Employees           0
Inventory           0
CL                  3
NI                  0
PP&E                0
Sales               0
Interest_Expense    0
Zip                 0
SIC                 0
Group               0
dtype: int64
```

```
#replace missing value with column median
administration.fillna(administration.median(), inplace=True)
```

Financial Ratios Formula

Cash Ratios

> $$CashToAssets = \frac{Cash}{Total Assets}$$
> $$CashToLiability = \frac{Cash}{Current Liabilities}$$
> $$CashToInterest = \frac{Cash}{InterestCharges}$$

Liquidity Ratios

> $$CurrentRatio = \frac{TotalCurrentAssets}{TotalCurrentLiabilities}$$
> $$QuickRatio = \frac{CurrentAssets - Inventories}{CurrentLiabilities}$$

Asset Management Ratios

> $$InventoryTurnover = \frac{CostofGoodsSold}{Inventories}$$
> $$FixedAssetsTurnover = \frac{AnnualSales}{NetFixedAssets}$$
> $$TotalAssetsTurnover = \frac{AnnualSales}{TotalAssets}$$

Debt Management Ratios

> $$LongTermDebtRatio = \frac{TotalLong-TermDebt}{TotalAssets}$$
> $$InterestCoverage = \frac{EBIT}{InterestCharges}$$

Operation Ratios

> $$ROA = \frac{NetIncome}{TotalAssets}$$

> $$SalesPerEmployee = \frac{Sales}{NumberOfEmployees}$$

The ratio we try to look at are all list above, we will measure the company from its liquidity, asset management, profit management, and liability aspects, and try to highlight the most statistically and economically significant findings in each industry.
Notice: We revise those 4 ratios (Cash To Interest, Inventory Turnover, Interest Coverage, Sales Per Employee) for our calculation in case those 0 be in the denominator bring infinite numbers for ratios, but we will explain those in the correct way.

Analytics Based on Industry

Mining Industry

Data Merge

```
#merge industry 2019 and 2020 data
```

```
mining2 = pd.merge(mining, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
mining2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
mining2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	HES	3156.000	21782.000	1545.000	3106.000	7733.000	637.000	1.775	261.000	2510.000	-408.000	175
1	APA	1961.000	18107.000	247.000	3465.000	8752.000	-1332.000	3.163	502.000	1855.000	-3553.000	141
2	HAL	11212.000	25377.000	2268.000	18498.000	11265.000	2058.000	55.000	3139.000	4878.000	-1131.000	83
3	HP	1115.086	5839.515	347.943	2014.913	479.356	-12.269	8.510	149.653	410.238	-33.656	45
4	MRO	2135.000	20245.000	858.000	1652.000	5608.000	437.000	2.000	72.000	1745.000	480.000	171

Check Outliers

```
#statistics description
mining2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	C
count	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.00000
mean	4040.744697	20496.967394	1004.910485	4907.731242	5340.165515	786.001879	12.239121	825.148212	2722.77742
std	5630.964495	24133.782956	1427.390920	6520.079495	7464.320258	1591.760630	22.940608	1360.087766	3931.48027
min	50.634000	361.603000	0.963000	60.536000	43.709000	-1756.000000	0.089000	0.000000	40.01800
25%	319.585000	3461.682000	64.615000	668.688000	784.734000	39.319000	0.791000	30.000000	271.67000
50%	1362.000000	13717.000000	247.000000	2070.000000	3172.357000	259.498000	2.900000	125.479000	1182.00000
75%	5273.339000	24732.000000	1545.000000	5231.000000	6942.000000	1067.400000	9.173000	767.297000	2510.00000
max	18681.000000	109330.000000	5190.200000	25838.000000	39391.000000	6330.000000	105.000000	4608.000000	14949.00000

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in mining2.columns[1:]:
  mining2[col] = winsorize(mining2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
mining2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL
count	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000
mean	3988.126545	19321.352818	1001.856152	4685.493788	4623.035636	724.696697	11.120545	810.663364	2667.437485
std	5493.358738	20311.647662	1418.076112	5877.115723	4720.740266	1348.020329	18.793272	1320.511186	3760.570505
min	82.235000	382.322000	2.370000	66.700000	82.423000	-1362.071000	0.176000	0.000000	64.800000
25%	319.585000	3461.682000	64.615000	668.688000	784.734000	39.319000	0.791000	30.000000	271.670000
50%	1362.000000	13717.000000	247.000000	2070.000000	3172.357000	259.498000	2.900000	125.479000	1182.000000
75%	5273.339000	24732.000000	1545.000000	5231.000000	6942.000000	1067.400000	9.173000	767.297000	2510.000000
max	16913.000000	70514.000000	5088.000000	18498.000000	15687.000000	3913.000000	68.000000	4130.000000	13098.000000

```
# Add important useful varialbes back
mining2['SIC']=mining['SIC'].values
mining2['Group']=mining['Group'].values
mining2['Zip']=mining['Zip'].values
mining2.head()
```

```
mining2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	HES	3156.000	21782.000	1545.000	3106.000	7733.000	637.000	1.775	261.000	2510.000	-408.000	1756
1	APA	1961.000	18107.000	247.000	3465.000	8752.000	-1332.000	3.163	502.000	1855.000	-3553.000	1415
2	HAL	11212.000	25377.000	2268.000	18498.000	11265.000	2058.000	55.000	3139.000	4878.000	-1131.000	836
3	HP	1115.086	5839.515	347.943	2014.913	479.356	-12.269	8.510	149.653	410.238	-33.656	450
4	MRO	2135.000	20245.000	858.000	1652.000	5608.000	437.000	2.000	72.000	1745.000	480.000	1719

Calculate Financial Ratios

```
# Calculate financial ratios
mining2['CashToAssets'] = (mining2.Cash/mining2.TA)
mining2['CashToInterest'] = 1/(mining2.Cash/mining2['Interest_Expense'])
mining2['CashToLiability'] = (mining2.Cash/mining2.CL)
mining2['CurrentRatio'] = (mining2.CA/mining2.CL)
mining2['QuickRatio'] = (mining2.CA-mining2.Inventory)/mining2.CL
mining2['InventoryTurnover'] = 1/(mining2.COGS/mining2.Inventory)
mining2['FixedAssetsTurnover'] = (mining2.Sales/mining2['PP&E'])
mining2['TotalAssetsTurnover'] = (mining2.Sales/mining2.TA)
mining2['LongTermDebtRatio'] = (mining2['Long_Term_Debt']/mining2.TA)
mining2['InterestCoverage'] = 1/(mining2.EBIT/mining2['Interest_Expense'])
mining2['ROA'] = (mining2.NI/mining2.TA)
mining2['SalesPerEmployee']=1/(mining2.Sales/(mining2.Employees*1000))
```

```
mining2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	HES	3156.000	21782.000	1545.000	3106.000	7733.000	637.000	1.775	261.000	2510.000	-408.000	1756
1	APA	1961.000	18107.000	247.000	3465.000	8752.000	-1332.000	3.163	502.000	1855.000	-3553.000	1415
2	HAL	11212.000	25377.000	2268.000	18498.000	11265.000	2058.000	55.000	3139.000	4878.000	-1131.000	836
3	HP	1115.086	5839.515	347.943	2014.913	479.356	-12.269	8.510	149.653	410.238	-33.656	450
4	MRO	2135.000	20245.000	858.000	1652.000	5608.000	437.000	2.000	72.000	1745.000	480.000	1719

```
# Delete unnecessary variables after using
mining2.drop(mining2.columns[1:14],axis=1,inplace=True)
mining2.head()
```

	Ticker	YTD	SIC	Group	Zip	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	Inventor
0	HES	-0.4983	13	1	10036	0.070930	0.270550	0.615538	1.257371	1.153386	
1	APA	-0.7898	13	1	77056	0.013641	1.769231	0.133154	1.057143	0.786523	
2	HAL	-0.6890	13	1	77032	0.089372	0.261023	0.464945	2.298483	1.654982	
3	HP	-0.6597	13	1	74119	0.059584	0.067505	0.848149	2.718144	2.353349	
4	MRO	-0.7312	13	1	77056-2723	0.042381	0.326340	0.491691	1.223496	1.182235	

```
#statistics ratio description check
mining2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	33.00000	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000	33.000000
mean	-0.60460	12.787879	0.049169	3.084887	0.393976	1.549595	1.241453	0.118953
std	0.21513	0.960390	0.054288	7.244795	0.380207	0.770671	0.619125	0.113662
min	-0.89980	10.000000	0.000533	0.067505	0.006561	0.472083	0.436552	0.000000
25%	-0.73120	13.000000	0.011063	0.199683	0.086807	1.057143	0.841652	0.020835
50%	-0.66560	13.000000	0.039234	0.290220	0.324446	1.291266	1.116031	0.076976
75%	-0.49830	13.000000	0.059584	1.943089	0.507109	1.998443	1.628490	0.181402
max	-0.00340	14.000000	0.216338	35.683966	1.905864	3.526235	3.341049	0.381587

Industry Overview

Here are 33 companies in total under mining catergory. Mining industry invest heavily on fixed assest, so their fixed assest turnover is 0.9 on average. However, its mean ROA is negative, which indicates that mining industry is facing downsizing and losing the sales since 2019.

More importantly, the mean YTD is -0.60, which is the greatest across all industries we analyzed. It means that Mining industry has been nearly crushed by COVID-19 event in 2020.

OLS

```
import statsmodels.api as sm
```

```
#mining
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = mining2.iloc[:, i]
  y = mining2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=mining2.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	-0.522139	0.464828	0.017361
1	CashToInterest	-0.005132	0.336172	0.029867
2	CashToLiability	0.074653	0.464229	0.017407
3	CurrentRatio	0.100430	0.039735	0.129438
4	QuickRatio	0.074650	0.229890	0.046155
5	InventoryTurnover	0.798620	0.014451	0.178036
6	FixedAssetsTurnover	-0.021534	0.621540	0.007957
7	TotalAssetsTurnover	-0.232295	0.269901	0.039117
8	LongTermDebtRatio	-0.514747	0.052234	0.116189
9	InterestCoverage	0.015900	0.467813	0.017132
10	ROA	0.892612	0.001356	0.285636
11	SalesPerEmployee	0.017776	0.490409	0.015471

Analysis:

According to the summary result above, we noticed that `Current Ratio`,`Inventory Turnover` and `ROA` are statistically and economically significant determinants of stock returns. AS we can see from the results above that `ROA` affects the stock returns positively, which indicates the higher profit that mining industry make, the higher stock returns will get.

Meanwhile, the reverse of inventory turnover ratio is positivly correlated to `YTD`. It indicates that the longer the companies in mining convert the inventories to revenues, the higher `YTD` it will be. Apparently, the situation of mining industry is different from most of other industries which need to have lower inventory turnover ratios in order to generate `YTD`. The reason is that mining industry is not an efficient industry, which perform slowly in selling its inventories in short term. Also, with the start of WFH policy, the uncertainties across the supply chain affecting commodity prices. Most mining operations around the world have been placed into shutdown.

```
# Multi Regression
X = mining2.iloc[:,5:17]
y = mining2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.513
Model:	OLS	**Adj. R-squared:**	0.221
Method:	Least Squares	**F-statistic:**	1.757
Date:	Sun, 19 Apr 2020	**Prob (F-statistic):**	0.128
Time:	23:21:37	**Log-Likelihood:**	16.265
No. Observations:	33	**AIC:**	-6.531
Df Residuals:	20	**BIC:**	12.92
Df Model:	12		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-0.5852	0.156	-3.743	0.001	-0.911	-0.259
CashToAssets	-0.1557	1.522	-0.102	0.919	-3.330	3.018
CashToInterest	-0.0031	0.006	-0.559	0.582	-0.015	0.009
CashToLiability	0.0607	0.320	0.190	0.851	-0.607	0.728
CurrentRatio	0.0970	0.218	0.446	0.661	-0.357	0.551
QuickRatio	-0.0459	0.273	-0.168	0.868	-0.615	0.523
InventoryTurnover	0.5227	0.691	0.756	0.458	-0.919	1.964
FixedAssetsTurnover	-0.0056	0.066	-0.086	0.933	-0.143	0.132
TotalAssetsTurnover	-0.2762	0.313	-0.882	0.388	-0.929	0.377
LongTermDebtRatio	-0.0852	0.311	-0.274	0.787	-0.734	0.564
InterestCoverage	0.0167	0.021	0.788	0.440	-0.027	0.061
ROA	0.5961	0.371	1.607	0.124	-0.177	1.370
SalesPerEmployee	-0.0163	0.033	-0.486	0.632	-0.086	0.054

Omnibus:	14.211	**Durbin-Watson:**	2.151
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	16.413
Skew:	1.203	**Prob(JB):**	0.000273
Kurtosis:	5.480	**Cond. No.**	370.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Analysis:
For the last step, we also applied multiple regression on the mining industry, which means we keep all other variables constant to see the performance of each individual variable. Comparing with our previous single regression results, no variable is significant under multiple regression.

Specifically, though mining industry has very low `YTD` since 2020, its multiple Adjusted r-squared is around 22% which means those ratios still explains 22% of `YTD` even with an unforseen COVID-19 situation in 2020. Our assumption is that maybe mining industry does not perform very well in 2019 with a negative mean ROA, so its prediction to 2020 `YTD` is still acceptable.

(For the last step, we also applied multiple regression on the mining industry, which means we keep all other variables the same to see the performance of each individual variable. Combining the concluion from the previous single regression and the p-value, we notice that for both models, `Quick Ratio`,`Total Asset Turnover` and `Long Term Debt Ratio` are significantly affecting the minning industry. However, the R square score is 0.9, which means mining industry doesn't effected by COVID-19 that much, because we still get a relative accurate score by using the historial data from 2019. That might also because the industry hasn't perform so well in 2019 neither due to the global sources and competition.Moreover, nearly half of the ratios have negative effect on stock return.)

Construction Industry

Data Merge

```
#merge industry 2019 and 2020 data
construction2 = pd.merge(construction, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
construction2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
construction2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&I
0	AEGN	450.215	995.513	64.874	925.033	299.882	64.315	4.900	57.193	234.041	-20.892	172.55
1	J	4111.768	11462.711	631.068	10086.602	1201.245	760.086	52.000	0.000	3073.706	847.184	308.14
2	LEN	912.707	29359.511	1445.996	17564.046	7014.438	2461.181	10.106	19738.586	670.448	1849.052	279.52
3	PHM	912.707	10715.597	1217.913	7788.885	2835.121	1348.353	5.245	8513.027	670.448	1016.700	181.74
4	NVR	912.707	3809.815	1140.585	5829.044	690.806	1051.175	5.700	2253.264	670.448	878.539	135.25

Check Outliers

```
#statistics description
construction2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	
count	10.000000	10.000000	10.000000	10.00000	10.00000	10.000000	10.00000	10.000000	10.000000	
mean	1433.021100	8502.800900	674.516500	7089.04820	1748.03790	905.876000	14.01200	4531.423700	995.748400	
std	1352.405841	8952.964025	595.001396	5818.10369	1994.93253	829.919553	17.41855	6917.127805	915.191838	
min	450.215000	995.513000	38.345000	687.00000	299.88200	64.315000	0.95300	0.000000	234.041000	
25%	912.707000	1707.202500	131.414000	1723.75700	692.15550	236.153500	4.98625	56.087500	670.448000	
50%	912.707000	6070.748500	529.034000	6808.96450	1097.12250	723.111500	7.30800	899.074000	670.448000	
75%	912.707000	11275.932500	1198.581000	10219.19600	1717.75400	1274.058500	10.00450	6948.086250	670.448000	
max	4111.768000	29359.511000	1494.300000	17564.04600	7014.43800	2461.181000	52.00000	19738.586000	3073.706000	1

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in construction2.columns[1:]:
  construction2[col] = winsorize(construction2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
construction2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	
count	10.000000	10.000000	10.000000	10.00000	10.00000	10.000000	10.00000	10.000000	10.000000	
mean	1433.021100	8502.800900	674.516500	7089.04820	1748.03790	905.876000	14.01200	4531.423700	995.748400	
std	1352.405841	8952.964025	595.001396	5818.10369	1994.93253	829.919553	17.41855	6917.127805	915.191838	
min	450.215000	995.513000	38.345000	687.00000	299.88200	64.315000	0.95300	0.000000	234.041000	
25%	912.707000	1707.202500	131.414000	1723.75700	692.15550	236.153500	4.98625	56.087500	670.448000	
50%	912.707000	6070.748500	529.034000	6808.96450	1097.12250	723.111500	7.30800	899.074000	670.448000	
75%	912.707000	11275.932500	1198.581000	10219.19600	1717.75400	1274.058500	10.00450	6948.086250	670.448000	
max	4111.768000	29359.511000	1494.300000	17564.04600	7014.43800	2461.181000	52.00000	19738.586000	3073.706000	1

```
# Add important useful varialbes back
construction2['SIC']=construction['SIC'].values
```

```
construction2['Group']=construction['Group'].values
construction2['Zip']=construction['Zip'].values
construction2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&I
0	AEGN	450.215	995.513	64.874	925.033	299.882	64.315	4.900	57.193	234.041	-20.892	172.55
1	J	4111.768	11462.711	631.068	10086.602	1201.245	760.086	52.000	0.000	3073.706	847.184	308.14
2	LEN	912.707	29359.511	1445.996	17564.046	7014.438	2461.181	10.106	19738.586	670.448	1849.052	279.52
3	PHM	912.707	10715.597	1217.913	7788.885	2835.121	1348.353	5.245	8513.027	670.448	1016.700	181.74
4	NVR	912.707	3809.815	1140.585	5829.044	690.806	1051.175	5.700	2253.264	670.448	878.539	135.25

Calculate Financial Ratios

```
# Calculate financial ratios
construction2['CashToAssets'] = (construction2.Cash/construction2.TA)
construction2['CashToInterest'] = 1/(construction2.Cash/construction2['Interest_Expense'])
construction2['CashToLiability'] = (construction2.Cash/construction2.CL)
construction2['CurrentRatio'] = (construction2.CA/construction2.CL)
construction2['QuickRatio'] = (construction2.CA-construction2.Inventory)/construction2.CL
construction2['InventoryTurnover'] = 1/(construction2.COGS/construction2.Inventory)
construction2['FixedAssetsTurnover'] = (construction2.Sales/construction2['PP&E'])
construction2['TotalAssetsTurnover'] = (construction2.Sales/construction2.TA)
construction2['LongTermDebtRatio'] = (construction2['Long_Term_Debt']/construction2.TA)
construction2['InterestCoverage'] =1/ (construction2.EBIT/construction2['Interest_Expense'])
construction2['ROA'] = (construction2.NI/construction2.TA)
construction2['SalesPerEmployee']=1/(construction2.Sales/(construction2.Employees*1000))
```

```
construction2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&I
0	AEGN	450.215	995.513	64.874	925.033	299.882	64.315	4.900	57.193	234.041	-20.892	172.55
1	J	4111.768	11462.711	631.068	10086.602	1201.245	760.086	52.000	0.000	3073.706	847.184	308.14
2	LEN	912.707	29359.511	1445.996	17564.046	7014.438	2461.181	10.106	19738.586	670.448	1849.052	279.52
3	PHM	912.707	10715.597	1217.913	7788.885	2835.121	1348.353	5.245	8513.027	670.448	1016.700	181.74
4	NVR	912.707	3809.815	1140.585	5829.044	690.806	1051.175	5.700	2253.264	670.448	878.539	135.25

```
# Delete unnecessary variables after using
construction2.drop(construction2.columns[1:14],axis=1,inplace=True)
construction2.head()
```

	Ticker	YTD	SIC	Group	Zip	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	Inventor
0	AEGN	-0.3330	16	2	63005-1195	0.065166	0.215834	0.277191	1.923659	1.679287	
1	J	-0.1209	16	2	75201	0.055054	0.132865	0.205312	1.337723	1.337723	
2	LEN	-0.4015	15	2	33172	0.049251	0.292331	2.156761	1.361339	-28.079551	
3	PHM	-0.5124	15	2	30326	0.113658	0.135230	1.816566	1.361339	-11.336181	
4	NVR	-0.3795	15	2	20190	0.299381	0.023201	1.701228	1.361339	-1.999494	

```
#statistics ratio description recheck
construction2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	-0.356220	15.500000	0.112837	0.279948	0.986837	1.438874	-5.329983	0.480804
std	0.116808	0.707107	0.113542	0.336448	0.919582	0.204270	10.366950	0.516377
min	-0.513200	15.000000	0.019780	0.023201	0.057193	1.266484	-28.079551	0.000000
25%	-0.398800	15.000000	0.050702	0.133456	0.185886	1.361339	-9.002009	0.019529
50%	-0.368150	15.000000	0.065440	0.156138	0.725134	1.361339	0.020334	0.242241
75%	-0.311925	16.000000	0.109180	0.273207	1.787731	1.361339	1.319913	1.055878
max	-0.120900	17.000000	0.341600	1.188030	2.228808	1.923659	1.679287	1.123806

Industry Overview

Here are only 10 companies in total under the Construction category. The `YTD` for construction category is around -0.35, showing that it received a great negative stock returns since 2020 due to COVID-19.

For its ratios since 2019, the mean Quick Ratio is -5.32, which shows that construction industry's companies are not able to fully pay off its current liabilities in the short term. The scale of ratio number is also large, it might be caused by the fact that we only have 10 observations left in this category.

OLS

```
import statsmodels.api as sm
```

```
#construction
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = construction2.iloc[:, i]
  y = construction2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=construction2.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	-0.121588	0.745044	0.013969
1	CashToInterest	-0.133949	0.270829	0.148857
2	CashToLiability	-0.057253	0.191096	0.203161
3	CurrentRatio	0.114501	0.579128	0.040094
4	QuickRatio	0.004582	0.243559	0.165347
5	InventoryTurnover	-0.178346	0.006732	0.621608
6	FixedAssetsTurnover	-0.000851	0.090879	0.315823
7	TotalAssetsTurnover	0.080055	0.496093	0.059755
8	LongTermDebtRatio	-0.171659	0.406876	0.087415
9	InterestCoverage	-0.304528	0.625214	0.031240
10	ROA	-0.499783	0.434820	0.077905
11	SalesPerEmployee	0.061347	0.003567	0.674682

Analysis:

According to the summary result above, we noticed that `Inverntory Turnover` and `SalesPerEmployee` are statistically and economically significant determinants of stock returns. But the increase in `Inventory Turnover` will decrease the stock returns. The employees in the construction industry by default is skilled workers, electricians, engineers and more. Additionally, according to the report from Construction Dive, the Economist estimated that 30% of building products and raw materials for Construction Industry in the U.S are imported from China, which is the nation's largest single supplier. With global affection, material shortfall seems to impact construction in the US.

```
construction2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	-0.356220	15.500000	0.112837	0.279948	0.986837	1.438874	-5.329983	0.480804
std	0.116808	0.707107	0.113542	0.336448	0.919582	0.204270	10.366950	0.516377
min	-0.513200	15.000000	0.019780	0.023201	0.057193	1.266484	-28.079551	0.000000
25%	-0.398800	15.000000	0.050702	0.133456	0.185886	1.361339	-9.002009	0.019529
50%	-0.368150	15.000000	0.065440	0.156138	0.725134	1.361339	0.020334	0.242241
75%	-0.311925	16.000000	0.109180	0.273207	1.787731	1.361339	1.319913	1.055878
max	-0.120900	17.000000	0.341600	1.188030	2.228808	1.923659	1.679287	1.123806

```
# Multi Regression
X = construction2.iloc[:,[5, 6, 11,12,13,14,15,16]]
y = construction2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.950
Model:	OLS	**Adj. R-squared:**	0.546
Method:	Least Squares	**F-statistic:**	2.355
Date:	Sun, 19 Apr 2020	**Prob (F-statistic):**	0.467
Time:	23:22:23	**Log-Likelihood:**	22.749
No. Observations:	10	**AIC:**	-27.50
Df Residuals:	1	**BIC:**	-24.77
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.8059	0.433	-1.861	0.314	-6.309	4.697
CashToAssets	-1.3738	1.599	-0.859	0.548	-21.688	18.940
CashToInterest	0.2281	0.481	0.474	0.718	-5.880	6.337
FixedAssetsTurnover	-0.0029	0.003	-1.098	0.470	-0.037	0.031
TotalAssetsTurnover	-0.1307	0.138	-0.945	0.518	-1.888	1.626
LongTermDebtRatio	-0.4738	0.858	-0.552	0.679	-11.371	10.424
InterestCoverage	2.8271	3.186	0.887	0.538	-37.649	43.303
ROA	5.2544	3.694	1.422	0.390	-41.682	52.191
SalesPerEmployee	0.0832	0.031	2.660	0.229	-0.314	0.481

Omnibus:	2.711	**Durbin-Watson:**	1.634
Prob(Omnibus):	0.258	**Jarque-Bera (JB):**	0.310
Skew:	-0.023	**Prob(JB):**	0.856
Kurtosis:	3.861	**Cond. No.**	1.83e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.83e+04. This might indicate that there are strong multicollinearity or other numerical problems.

Analysis:

Since we only have 10 observations, instead of using all ratios, we adopt 8 financial ratios which are not calculated by `CA` and `CL` and do not have extreme value of financial ratios from single variable OLS analysis into multiple regression model. In other words, during the data cleaning stage, since `CA` and `CL` have 5 NAs, we replaced these NAs by industry mean. Meanwhile, since we have too less data points, such substitutions might pose more unpredictive errors into our model. Therefore we did not put `CashToLiability`, `CurrentRatio` and `Quick Ratio` into model.

The result shows that no variables in the multiple regression are significant. However, though our variables are not significant, the overall R-squared adjusted is 54.6%, which is much higher than usual multiple regression models in other industries. We think that this situation is caused by overfitting problem, since we only have 10 observations in the dataset. Even though R-squared adjusted is plausible, the model itself is not predictive. Therefore, we think we nee do obtain more observations in the future to better predict construction industry's `YTD` in 2020.

Manufacturing Industry

Data Merge

```
#merge industry 2019 and 2020 data
manufacturing2 = pd.merge(manufacturing, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
manufacturing2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
manufacturing2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	ABT	15667.000	67887.000	3860.000	11949.000	17416.000	5041.000	107.000	4316.000	10863.000	3687.000	897
1	AMD	4597.000	6028.000	1466.000	3721.000	685.000	583.000	11.400	982.000	2359.000	341.000	70
2	APD	4618.300	18942.800	2248.700	4892.700	3227.400	2120.500	17.700	388.300	1820.900	1760.000	1033
3	ATRI	132.338	262.031	45.048	73.525	0.268	40.529	0.616	42.093	11.274	36.761	8
4	SWKS	2234.600	4839.600	851.300	1353.000	0.000	1042.100	9.000	609.700	374.000	853.600	120

Check Outliers

```
#statistics description
manufacturing2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	
count	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.
mean	4491.602207	14522.359408	1227.360960	5639.507105	3936.748379	1267.901833	17.283195	1093.265937	3238.
std	12540.774663	36900.634988	3698.413693	18058.999422	9754.081894	3930.142591	30.860213	3960.669928	10426.
min	4.324000	10.113000	1.593000	0.000000	0.000000	-939.431000	0.006000	0.000000	1.
25%	380.115500	609.991000	95.641750	162.729250	41.594250	-26.475250	0.578000	18.588000	83.
50%	996.062000	2144.296000	266.421000	787.448000	476.874000	148.973000	4.074000	219.400000	352.
75%	3375.900000	11259.358750	833.401000	3706.668000	3449.375000	1064.965500	18.000000	871.250000	1868.
max	162819.000000	362597.000000	48844.000000	211152.000000	102408.000000	63930.000000	267.000000	76622.000000	105718.

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in manufacturing2.columns[1:]:
  manufacturing2[col] = winsorize(manufacturing2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
manufacturing2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL
count	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.00000	478.000000	478.000000
mean	3252.930362	10833.331121	815.457918	3362.393615	3179.283615	910.790636	15.24141	770.864833	2128.058010
std	5119.159679	18424.269519	1265.323813	5448.262644	5729.367957	1616.665020	22.86501	1206.583850	3932.977719
min	103.951000	147.985000	20.707000	15.822000	0.744000	-220.545000	0.09700	0.000000	17.002000
25%	380.115500	609.991000	95.641750	162.729250	41.594250	-26.475250	0.57800	18.588000	83.848250
50%	996.062000	2144.296000	266.421000	787.448000	476.874000	148.973000	4.07400	219.400000	352.594000
75%	3375.900000	11259.358750	833.401000	3706.668000	3449.375000	1064.965500	18.00000	871.250000	1868.225000
max	19780.000000	69396.000000	4825.430000	20276.500000	22155.800000	5929.000000	80.00000	4316.000000	15322.000000

```
# Add important useful varialbes back
manufacturing2['SIC']=manufacturing['SIC'].values
manufacturing2['Group']=manufacturing['Group'].values
manufacturing2['Zip']=manufacturing['Zip'].values
manufacturing2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	ABT	15667.000	67887.000	3860.000	11949.000	17416.000	5041.000	80.000	4316.000	10863.000	3687.000	897
1	AMD	4597.000	6028.000	1466.000	3721.000	685.000	583.000	11.400	982.000	2359.000	341.000	70
2	APD	4618.300	18942.800	2248.700	4892.700	3227.400	2120.500	17.700	388.300	1820.900	1760.000	1033
3	ATRI	132.338	262.031	45.048	73.525	0.744	40.529	0.616	42.093	17.002	36.761	8
4	SWKS	2234.600	4839.600	851.300	1353.000	0.744	1042.100	9.000	609.700	374.000	853.600	120

Calculate Financial Ratios

```
# Calculate financial ratios
manufacturing2['CashToAssets'] = (manufacturing2.Cash/manufacturing2.TA)
manufacturing2['CashToInterest'] = 1/(manufacturing2.Cash/manufacturing2['Interest_Expense'])
manufacturing2['CashToLiability'] = (manufacturing2.Cash/manufacturing2.CL)
manufacturing2['CurrentRatio'] = (manufacturing2.CA/manufacturing2.CL)
manufacturing2['QuickRatio'] = (manufacturing2.CA-manufacturing2.Inventory)/manufacturing2.CL
manufacturing2['InventoryTurnover'] = 1/(manufacturing2.COGS/manufacturing2.Inventory)
manufacturing2['FixedAssetsTurnover'] = (manufacturing2.Sales/manufacturing2['PP&E'])
manufacturing2['TotalAssetsTurnover'] = (manufacturing2.Sales/manufacturing2.TA)
manufacturing2['LongTermDebtRatio'] = (manufacturing2['Long_Term_Debt']/manufacturing2.TA)
manufacturing2['InterestCoverage'] = 1/(manufacturing2.EBIT/manufacturing2['Interest_Expense'])
manufacturing2['ROA'] = (manufacturing2.NI/manufacturing2.TA)
manufacturing2['SalesPerEmployee']=1/(manufacturing2.Sales/(manufacturing2.Employees*1000))
```

```
manufacturing2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	ABT	15667.000	67887.000	3860.000	11949.000	17416.000	5041.000	80.000	4316.000	10863.000	3687.000	897
1	AMD	4597.000	6028.000	1466.000	3721.000	685.000	583.000	11.400	982.000	2359.000	341.000	70
2	APD	4618.300	18942.800	2248.700	4892.700	3227.400	2120.500	17.700	388.300	1820.900	1760.000	1033
3	ATRI	132.338	262.031	45.048	73.525	0.744	40.529	0.616	42.093	17.002	36.761	8
4	SWKS	2234.600	4839.600	851.300	1353.000	0.744	1042.100	9.000	609.700	374.000	853.600	120

```
# Delete unnecessary variables after using
manufacturing2.drop(manufacturing2.columns[1:14],axis=1,inplace=True)
manufacturing2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000	478.000000
mean	-0.279433	31.378661	0.176831	0.255843	1.473923	3.922866	3.385347	0.302313
std	0.189446	5.261398	0.182799	0.719715	2.324561	4.411587	4.540315	0.405663
min	-0.595600	20.000000	0.002358	0.000000	0.009158	0.269628	-9.415753	0.000000
25%	-0.417800	28.000000	0.058925	0.007289	0.309443	1.538205	1.056604	0.107394
50%	-0.286050	28.500000	0.121826	0.082403	0.753826	2.690610	1.904644	0.219856
75%	-0.154375	36.000000	0.212118	0.241213	1.569809	5.063452	4.528078	0.381757
max	0.098000	39.000000	0.973483	11.580645	21.210511	66.088990	65.659864	4.454815

Industry Overview

Here are 478 companies in total under the Manufacturing Industry category. Firstly, we notice that its `YTD` is -0.27, which means that on average, there is an investment loss on manufacturing companies overall. However, compared with other 8 industries, the scale of `YTD` is the smallest. It may caused by increasing demand of facial masks and other medical and protection products during COVID-19 situation.

We observe that the means of `Interest Coverage Ratio` and `ROA` are also negative. These facts demonstrate that manufacturing companies overall are losing money across the year of 2019, and its ability to pay its interest expense is not good.Meanwhile, manufacturing has only 0.17 mean of `Cash to Assets Ratio`, which is less than 1. It means that its ability of covering liabilities in short term is weak. This situation is normal for manufacturing industry, since manufacturing industry invests heavily on fixed assets and is expecting to gain the returns for over a year.

OLS

```
import statsmodels.api as sm
```

```
#manufacturing
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = manufacturing2.iloc[:, i]
  y = manufacturing2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=manufacturing2.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	0.024926	0.599903	0.000578
1	CashToInterest	-0.009806	0.416418	0.001388
2	CashToLiability	0.008705	0.019504	0.011408
3	CurrentRatio	0.005527	0.004826	0.016567
4	QuickRatio	0.005676	0.002880	0.018505
5	InventoryTurnover	0.050815	0.017322	0.011840
6	FixedAssetsTurnover	0.000784	0.505188	0.000933
7	TotalAssetsTurnover	-0.060516	0.003666	0.017597
8	LongTermDebtRatio	-0.054490	0.107849	0.005423
9	InterestCoverage	-0.001122	0.694896	0.000323
10	ROA	0.076923	0.052768	0.007858
11	SalesPerEmployee	-0.000609	0.033799	0.009429

Analysis:

According to the summary result above, we noticed that `CashToLiability` , `Current Ratio` , `Quick Ratio`, `Inventory Turnover` , `TotalAsset Turnover` and `SalesPerEmployee` are statistically and economically significant determinants of stock returns. However, even though those ratios are significant, they cannot explained the `YTD` based on their nearly 0% r-squared. In addtion, as we mentioned before, the manufacturing companies are lossing money over the year 2019, and its overall performance of total asset turnover and sales generated by each employee are not good. It reconciles with the results above that the `TotalAsset Turnover` and `SalesPerEmployee` has negative effects on stock returns.

```
# Multi Regression
X = manufacturing2.iloc[:,5:17]
y = manufacturing2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.086
Model:	OLS	**Adj. R-squared:**	0.062
Method:	Least Squares	**F-statistic:**	3.631
Date:	Sun, 19 Apr 2020	**Prob (F-statistic):**	3.22e-05
Time:	23:23:01	**Log-Likelihood:**	138.88
No. Observations:	478	**AIC:**	-251.8
Df Residuals:	465	**BIC:**	-197.6
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.2253	0.027	-8.237	0.000	-0.279	-0.172
CashToAssets	-0.0682	0.080	-0.848	0.397	-0.226	0.090
CashToInterest	-0.0027	0.012	-0.217	0.828	-0.027	0.022
CashToLiability	0.0119	0.007	1.746	0.081	-0.001	0.025
CurrentRatio	-0.0201	0.011	-1.757	0.080	-0.043	0.002
QuickRatio	0.0214	0.011	1.983	0.048	0.000	0.043
InventoryTurnover	0.0429	0.022	1.973	0.049	0.000	0.086
FixedAssetsTurnover	-5.62e-05	0.001	-0.047	0.963	-0.002	0.002
TotalAssetsTurnover	-0.0794	0.026	-3.006	0.003	-0.131	-0.027
LongTermDebtRatio	-0.0326	0.034	-0.952	0.341	-0.100	0.035
InterestCoverage	-0.0004	0.003	-0.147	0.883	-0.006	0.005
ROA	0.1521	0.052	2.944	0.003	0.051	0.254
SalesPerEmployee	-0.0006	0.000	-2.168	0.031	-0.001	-5.87e-05

Omnibus:	16.735	**Durbin-Watson:**	1.906
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	11.737
Skew:	0.266	**Prob(JB):**	0.00283
Kurtosis:	2.446	**Cond. No.**	311.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Analysis:

For multiple regression, we can see that `Quick Ratio`, `TotalAssets Turnover`, `Inventory Turnover` , `ROA` and `SalesPerEmployee` are statistically and economically significant determinants of stock returns.

Overall the variables we chosen explained 6.2% variability of `YTD` in the beginning of 2020. The results are not plausible. Moreover, among all multiple regression, manufacturing industry has the lowest adjusted R squared, which further highlights the fact that both positive and negative dramatic effects brought by COVID-19 to this industry make its `YTD` cannot be fully explained by 2019 financial ratios. Combined with analysis of single linear regression, we assume that manufacturing industry received huge influence from COVID-19. Its `YTD` becomes less

Transportation Industry

Data Merge

```
#merge industry 2019 and 2020 data
transportation2 = pd.merge(transportation, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
transportation2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
transportation2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP
0	AAL	8206.000	59995.000	280.000	32027.000	28875.000	3706.000	133.700	1851.00	18311.000	1686.00	43732.0
1	PNW	1030.030	18479.247	10.283	2208.320	4884.430	671.960	6.210	345.92	2078.365	538.32	14377.7
2	ALK	2037.000	12993.000	221.000	6379.000	2703.000	1107.000	24.134	72.00	3201.000	769.00	8613.0
3	ECOL	332.971	2231.244	41.281	421.373	822.104	74.094	3.800	0.00	180.770	33.14	536.1
4	AEP	4077.800	75892.300	246.800	10454.600	26110.600	2592.300	17.408	1169.20	10299.100	1921.10	61095.5

Check Outliers

```
#statistics description
transportation2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	
count	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.0000
mean	4071.209000	39733.981602	905.693972	8454.817167	14661.794417	2529.558019	29.321176	412.904148	5361.0199
std	7435.049702	70039.432149	1740.570691	14358.589861	24439.510500	4838.914928	62.073499	878.937397	9301.8020
min	29.133000	301.940000	0.248000	73.939000	0.000000	-2702.480000	0.000000	0.000000	33.6920
25%	638.298500	3674.802250	67.187250	1387.421000	1331.386250	189.550750	4.566250	1.106250	503.3480
50%	2046.100000	17983.623500	245.269000	3591.000000	7939.450000	1212.934500	11.558500	81.000000	2428.6825
75%	3889.025000	46462.500000	862.750000	8241.500000	18352.500000	2639.250000	22.814750	440.650000	5547.3657
max	54761.000000	551669.000000	12130.000000	84141.000000	173113.000000	31069.000000	495.000000	6246.000000	68911.0000

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in transportation2.columns[1:]:
  transportation2[col] = winsorize(transportation2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
transportation2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL
count	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.00000
mean	3125.055102	33087.032074	733.497120	7082.499565	11750.211537	1977.658509	23.569306	326.311556	4398.40505
std	3409.206334	39301.956629	1048.261154	9015.737117	12612.501949	2256.806929	34.047045	498.505868	5187.31160
min	151.814000	932.182000	7.726000	264.667000	173.774000	23.189000	0.352000	0.000000	76.32600
25%	638.298500	3674.802250	67.187250	1387.421000	1331.386250	189.550750	4.566250	1.106250	503.34800
50%	2046.100000	17983.623500	245.269000	3591.000000	7939.450000	1212.934500	11.558500	81.000000	2428.68250
75%	3889.025000	46462.500000	862.750000	8241.500000	18352.500000	2639.250000	22.814750	440.650000	5547.36575
max	12037.000000	148188.000000	3561.000000	34044.000000	43413.000000	8150.000000	133.700000	1768.000000	18311.00000

```
# Add important useful varialbes back
transportation2['SIC']=transportation['SIC'].values
transportation2['Group']=transportation['Group'].values
transportation2['Zip']=transportation['Zip'].values
transportation2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP
0	AAL	8206.000	59995.000	280.000	32027.000	28875.000	3706.000	133.700	1768.00	18311.000	1686.00	43732.0
1	PNW	1030.030	18479.247	10.283	2208.320	4884.430	671.960	6.210	345.92	2078.365	538.32	14377.7
2	ALK	2037.000	12993.000	221.000	6379.000	2703.000	1107.000	24.134	72.00	3201.000	769.00	8613.0
3	ECOL	332.971	2231.244	41.281	421.373	822.104	74.094	3.800	0.00	180.770	33.14	536.1
4	AEP	4077.800	75892.300	246.800	10454.600	26110.600	2592.300	17.408	1169.20	10299.100	1921.10	61095.5

Calculate Financial Ratios

```
# Calculate financial ratios
transportation2['CashToAssets'] = (transportation2.Cash/transportation2.TA)
transportation2['CashToInterest'] = 1/(transportation2.Cash/transportation2['Interest_Expense'])
transportation2['CashToLiability'] = (transportation2.Cash/transportation2.CL)
transportation2['CurrentRatio'] = (transportation2.CA/transportation2.CL)
transportation2['QuickRatio'] = (transportation2.CA-transportation2.Inventory)/transportation2.CL
transportation2['InventoryTurnover'] = 1/(transportation2.COGS/transportation2.Inventory)
transportation2['FixedAssetsTurnover'] = (transportation2.Sales/transportation2['PP&E'])
transportation2['TotalAssetsTurnover'] = (transportation2.Sales/transportation2.TA)
transportation2['LongTermDebtRatio'] = (transportation2['Long_Term_Debt']/transportation2.TA)
transportation2['InterestCoverage'] = 1/(transportation2.EBIT/transportation2['Interest_Expense'])
transportation2['ROA'] = (transportation2.NI/transportation2.TA)
transportation2['SalesPerEmployee']=1/(transportation2.Sales/(transportation2.Employees*1000))
```

```
transportation2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP
0	AAL	8206.000	59995.000	280.000	32027.000	28875.000	3706.000	133.700	1768.00	18311.000	1686.00	43732.0
1	PNW	1030.030	18479.247	10.283	2208.320	4884.430	671.960	6.210	345.92	2078.365	538.32	14377.7
2	ALK	2037.000	12993.000	221.000	6379.000	2703.000	1107.000	24.134	72.00	3201.000	769.00	8613.0
3	ECOL	332.971	2231.244	41.281	421.373	822.104	74.094	3.800	0.00	180.770	33.14	536.1
4	AEP	4077.800	75892.300	246.800	10454.600	26110.600	2592.300	17.408	1169.20	10299.100	1921.10	61095.5

```
# Delete unnecessary variables after using
transportation2.drop(transportation2.columns[1:14],axis=1,inplace=True)
transportation2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000	108.000000
mean	-0.316982	46.787037	0.041617	3.258854	0.327216	1.098626	1.022044	0.043761
std	0.200373	2.771858	0.063590	6.099523	0.556575	0.773169	0.790226	0.048108
min	-0.714000	40.000000	0.000375	0.000855	0.004280	0.146189	0.125806	0.000000
25%	-0.435450	45.000000	0.005975	0.343851	0.045212	0.628352	0.504458	0.001430
50%	-0.275000	48.000000	0.016950	1.019857	0.153308	0.826427	0.722943	0.029940
75%	-0.159700	49.000000	0.059083	2.934685	0.357372	1.427473	1.344188	0.068471
max	-0.031700	49.000000	0.428481	36.210277	4.453163	5.303900	5.303900	0.209274

Industry Overview

The `YTD`'s mean of transportation industry is negative for Transportation industry, and the scale is -0.31, which is relatively large, indicating that transportation industry is facing negative influences by COVID-19 situation. For example, according to government regulations, most citizens cannot go outside.

We notice that the reverse of `Inventory Turnover` is around 0.04, which indicates that there are around 25 days for transportation industry to convert its inventory into sales. Due to properties of transportation industry, its inventory turnover ratio is nearly the highest around all industries.

OLS

```
import statsmodels.api as sm
```

```
#transportation
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = transportation2.iloc[:, i]
  y = transportation2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=transportation2.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	0.177606	0.562319	0.003177
1	CashToInterest	0.004579	0.150257	0.019426
2	CashToLiability	0.048014	0.168816	0.017787
3	CurrentRatio	0.038383	0.126095	0.021935
4	QuickRatio	0.032633	0.184364	0.016563
5	InventoryTurnover	0.857053	0.032639	0.042343
6	FixedAssetsTurnover	-0.001027	0.859334	0.000298
7	TotalAssetsTurnover	0.023061	0.533889	0.003661
8	LongTermDebtRatio	-0.107448	0.357815	0.007983
9	InterestCoverage	0.017772	0.390264	0.006972
10	ROA	0.295204	0.404057	0.006577
11	SalesPerEmployee	-0.013674	0.087432	0.027307

Analysis:

Overall, we noticed that only `Inventory Turnover` and `Sales Per Employee` are statistically significant determinants of stock returns. However, the scale of `Sales Per Employee` is very small, around -0.01, which indicates there is no meaningful economical effect on stock returns, and due to its properties, Transportation industry's performance may not rely on the average revenue generated by each employee very much. Meanwhile, the reverse of Inventory Turnover ratio has positive effects on `YTD`, which means that differnt from other industries requires low inventory turnover ratio, for transportation industry, the longer days of converting invetories into revenue, the more likely the company is having higher `YTD`.

```
# Multi Regression
X = transportation2.iloc[:,5:17]
y = transportation2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.201
Model:	OLS	**Adj. R-squared:**	0.100
Method:	Least Squares	**F-statistic:**	1.995
Date:	Sun, 19 Apr 2020	**Prob (F-statistic):**	0.0329
Time:	23:23:37	**Log-Likelihood:**	33.011
No. Observations:	108	**AIC:**	-40.02
Df Residuals:	95	**BIC:**	-5.153
Df Model:	12		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	-0.3814	0.087	-4.362	0.000	-0.555	-0.208
CashToAssets	-1.2650	0.740	-1.708	0.091	-2.735	0.205
CashToInterest	0.0040	0.003	1.181	0.241	-0.003	0.011
CashToLiability	0.1587	0.113	1.402	0.164	-0.066	0.383
CurrentRatio	-0.4210	0.335	-1.256	0.212	-1.087	0.245
QuickRatio	0.4670	0.345	1.353	0.179	-0.218	1.152
InventoryTurnover	1.5582	0.713	2.185	0.031	0.142	2.974
FixedAssetsTurnover	-0.0129	0.008	-1.528	0.130	-0.030	0.004
TotalAssetsTurnover	0.1524	0.068	2.233	0.028	0.017	0.288
LongTermDebtRatio	-0.1570	0.130	-1.210	0.229	-0.415	0.101
InterestCoverage	-0.0017	0.024	-0.069	0.945	-0.050	0.046
ROA	0.1833	0.372	0.492	0.624	-0.556	0.923
SalesPerEmployee	-0.0158	0.008	-1.878	0.063	-0.033	0.001

Omnibus:	3.058	**Durbin-Watson:**	2.019
Prob(Omnibus):	0.217	**Jarque-Bera (JB):**	3.006
Skew:	-0.359	**Prob(JB):**	0.222
Kurtosis:	2.609	**Cond. No.**	321.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Analysis:

For multiple regression, the results are similar to single OLS regression for transportation industry. Controlling with all ratios, `Sales per Employee`, `Total assets Turnover`, `Inventory Turnover` and `Cash to Assets` are statistically significant, which shows that the ability of generating revenue by total assets, and the ability of paying short-term liabilities are positively correlated with stock returns of Trnasportation industry.

Overall, our model has 10.0% adjusted R-squared value, which is not plausible and indicates the industry receives unforseen effects from COVID-19.

Wholesale Trade Industry

Data Merge

```
#merge industry 2019 and 2020 data
wholesale2 = pd.merge(wholesale, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
wholesale2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
wholesale2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	AVT	6876.336	8564.556	546.105	16851.648	1419.922	611.451	15.5	3008.424	2578.576	176.337	45
1	CAH	25747.000	40963.000	2531.000	138260.000	7579.000	1678.000	49.5	12822.000	24109.000	1363.000	235
2	GPC	7938.616	14645.629	276.992	13066.428	3627.623	1132.602	55.0	3831.183	6394.120	621.085	229
3	GWW	3555.000	6005.000	360.000	6939.000	2085.000	1388.000	25.3	1655.000	1678.000	849.000	162
4	SYY	8141.505	17966.522	513.460	47995.335	8122.058	2656.752	69.0	3216.034	6103.183	1674.271	450

Check Outliers

```
#statistics description
wholesale2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL
count	23.000000	23.000000	23.00000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000
mean	5312.137587	8841.764739	458.71613	21856.750957	2094.475217	745.636087	18.357696	2156.661522	4168.586413
std	7367.656074	10959.681956	828.21377	44306.992332	2295.810443	679.179847	19.252880	3310.627948	7531.201358
min	127.515000	390.360000	7.97300	268.751000	5.412000	9.451000	1.000000	20.941000	90.629000
25%	1303.646500	2046.140500	31.29150	1936.946000	522.850000	184.172500	5.150000	328.515000	501.361000
50%	2781.139500	4715.000000	154.00000	6593.092000	1335.803000	611.451000	11.261000	771.000000	1559.387500
75%	5782.077500	10672.256000	505.60450	13899.464000	2997.246500	1094.301000	21.974000	2890.600500	3034.988000
max	28132.054000	40963.000000	3374.19400	174543.941000	8122.058000	2656.752000	69.000000	12822.000000	29581.294000

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in wholesale2.columns[1:]:
  wholesale2[col] = winsorize(wholesale2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
wholesale2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	C
count	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.00000
mean	5209.478935	8781.828391	422.321913	20281.411696	2070.889696	715.928000	17.801174	2080.757261	3931.52219
std	7040.651932	10710.626440	702.237853	38943.703329	2232.908799	593.773574	17.691149	3063.344853	6735.85532
min	151.420000	802.844000	14.100000	319.889000	6.003000	68.969000	2.200000	36.889000	110.44600
25%	1303.646500	2046.140500	31.291500	1936.946000	522.850000	184.172500	5.150000	328.515000	501.36100
50%	2781.139500	4715.000000	154.000000	6593.092000	1335.803000	611.451000	11.261000	771.000000	1559.38750
75%	5782.077500	10672.256000	505.604500	13899.464000	2997.246500	1094.301000	21.974000	2890.600500	3034.98800
max	25747.000000	39171.980000	2531.000000	138260.000000	7579.000000	1913.948000	55.000000	11060.254000	24109.00000

```
# Add important useful varialbes back
wholesale2['SIC']=wholesale['SIC'].values
wholesale2['Group']=wholesale['Group'].values
wholesale2['Zip']=wholesale['Zip'].values
wholesale2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	AVT	6876.336	8564.556	546.105	16851.648	1419.922	611.451	15.5	3008.424	2578.576	176.337	45
1	CAH	25747.000	39171.980	2531.000	138260.000	7579.000	1678.000	49.5	11060.254	24109.000	1363.000	235
2	GPC	7938.616	14645.629	276.992	13066.428	3627.623	1132.602	55.0	3831.183	6394.120	621.085	229
3	GWW	3555.000	6005.000	360.000	6939.000	2085.000	1388.000	25.3	1655.000	1678.000	849.000	162
4	SYY	8141.505	17966.522	513.460	47995.335	7579.000	1913.948	55.0	3216.034	6103.183	1363.000	254

Calculate Financial Ratios

```
# Calculate financial ratios
wholesale2['CashToAssets'] = (wholesale2.Cash/wholesale2.TA)
wholesale2['CashToInterest'] = 1/(wholesale2.Cash/wholesale2['Interest_Expense'])
wholesale2['CashToLiability'] = (wholesale2.Cash/wholesale2.CL)
wholesale2['CurrentRatio'] = (wholesale2.CA/wholesale2.CL)
wholesale2['QuickRatio'] = (wholesale2.CA-wholesale2.Inventory)/wholesale2.CL
wholesale2['InventoryTurnover'] = 1/(wholesale2.COGS/wholesale2.Inventory)
wholesale2['FixedAssetsTurnover'] = (wholesale2.Sales/wholesale2['PP&E'])
wholesale2['TotalAssetsTurnover'] = (wholesale2.Sales/wholesale2.TA)
wholesale2['LongTermDebtRatio'] = (wholesale2['Long_Term_Debt']/wholesale2.TA)
wholesale2['InterestCoverage'] = 1/(wholesale2.EBIT/wholesale2['Interest_Expense'])
wholesale2['ROA'] = (wholesale2.NI/wholesale2.TA)
wholesale2['SalesPerEmployee']=1/(wholesale2.Sales/(wholesale2.Employees*1000))
```

```
wholesale2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	AVT	6876.336	8564.556	546.105	16851.648	1419.922	611.451	15.5	3008.424	2578.576	176.337	45
1	CAH	25747.000	39171.980	2531.000	138260.000	7579.000	1678.000	49.5	11060.254	24109.000	1363.000	235
2	GPC	7938.616	14645.629	276.992	13066.428	3627.623	1132.602	55.0	3831.183	6394.120	621.085	229
3	GWW	3555.000	6005.000	360.000	6939.000	2085.000	1388.000	25.3	1655.000	1678.000	849.000	162
4	SYY	8141.505	17966.522	513.460	47995.335	7579.000	1913.948	55.0	3216.034	6103.183	1363.000	254

```
# Delete unnecessary variables after using
wholesale2.drop(wholesale2.columns[1:14],axis=1,inplace=True)
wholesale2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000	23.000000
mean	-0.325178	50.347826	0.042860	0.697742	0.209778	1.961242	1.189793	0.176079
std	0.186448	0.486985	0.049680	1.180446	0.372864	0.954535	0.596714	0.124039
min	-0.597100	50.000000	0.007211	0.000000	0.009136	1.067941	0.609181	0.028944
25%	-0.454350	50.000000	0.016200	0.093917	0.047140	1.337310	0.839351	0.073502
50%	-0.361500	50.000000	0.027444	0.246974	0.099164	1.745066	1.038374	0.178524
75%	-0.167000	51.000000	0.061680	0.567215	0.213163	2.135219	1.364622	0.240441
max	-0.043400	51.000000	0.245800	4.792377	1.787089	4.693270	3.019463	0.511837

Industry Overview

There are 23 observations in Wholesale industry. `YTD`'s mean is -0.32, indicating negative stock returns since 2020 with outbreak of COVID-19.

Moreover, the mean of reverse of inventory turnover ratio `inventory turnover` is 0.17, which means on average, it has inventory turnover ratio around 5 to 6 for whole sale companies to convert its products into sales.

However, the mean cash to assets ratio is only 0.69, indicating that the liquidity of wholesale industry is relatively low.

OLS

```
import statsmodels.api as sm
```

```
#wholesale
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = wholesale2.iloc[:, i]
  y = wholesale2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=wholesale2.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	1.659697	0.034604	0.195574
1	CashToInterest	-0.065698	0.048373	0.173013
2	CashToLiability	0.146749	0.174111	0.086127
3	CurrentRatio	0.039957	0.349121	0.041846
4	QuickRatio	0.025718	0.708878	0.006775
5	InventoryTurnover	-0.066151	0.841962	0.001937
6	FixedAssetsTurnover	0.002511	0.133434	0.103987
7	TotalAssetsTurnover	0.063140	0.020902	0.228969
8	LongTermDebtRatio	-0.350825	0.128628	0.106469
9	InterestCoverage	-0.329797	0.127971	0.106816
10	ROA	0.562179	0.381373	0.036676
11	SalesPerEmployee	-0.031743	0.360563	0.039942

Analysis:
According to the summary result above, we noticed that `CashToAssets`, `CashToInterest` and `TotalAssets Turnover` are statistically and economically significant determinants of stock returns. Because of the stay home policy since the breakout of COV-19, the need of wholesale industry has increased rapidly which proves these affects, which is correspond to the `TotalAssets Turnover` explained the stock returns most (23% r-squared).

```
# Multi Regression
X = wholesale2.iloc[:,5:17]
y = wholesale2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.784
Model:	OLS	**Adj. R-squared:**	0.525
Method:	Least Squares	**F-statistic:**	3.029
Date:	Mon, 20 Apr 2020	**Prob (F-statistic):**	0.0444
Time:	00:48:38	**Log-Likelihood:**	24.142
No. Observations:	23	**AIC:**	-22.28
Df Residuals:	10	**BIC:**	-7.523
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.4646	0.212	-2.191	0.053	-0.937	0.008
CashToAssets	5.0574	2.671	1.893	0.088	-0.894	11.009
CashToInterest	0.0993	0.054	1.837	0.096	-0.021	0.220
CashToLiability	-0.2464	0.407	-0.605	0.559	-1.154	0.661
CurrentRatio	0.0247	0.148	0.167	0.871	-0.306	0.355
QuickRatio	-0.1481	0.215	-0.688	0.507	-0.628	0.331
InventoryTurnover	0.0806	0.771	0.105	0.919	-1.637	1.798
FixedAssetsTurnover	-0.0004	0.003	-0.157	0.879	-0.006	0.006
TotalAssetsTurnover	0.0695	0.037	1.861	0.092	-0.014	0.153
LongTermDebtRatio	-0.7120	0.304	-2.345	0.041	-1.388	-0.036
InterestCoverage	-0.0876	0.215	-0.408	0.692	-0.567	0.391
ROA	1.6599	0.702	2.364	0.040	0.095	3.225
SalesPerEmployee	-0.0177	0.052	-0.343	0.739	-0.133	0.098

Omnibus:	4.207	**Durbin-Watson:**	2.104
Prob(Omnibus):	0.122	**Jarque-Bera (JB):**	2.461
Skew:	-0.749	**Prob(JB):**	0.292
Kurtosis:	3.568	**Cond. No.**	3.49e+03

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.49e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

With the application of multiple regression on the wholesale industry, we get a fairly high adjusted R-squared of 52.5% compared with other industries, indicating that 2019 ratios can predict 2020 `YTD` with 52.5% explained variability. However, we cannot rule out the effect of overfitting, since we only have 23 observations for this industry. In order to have more accurate result, we will need to implement this regression with more data in the future.

We notice that `cash to assets`, `cash to interest`, `total assets turnover`, `long-term debt ratio` and `ROA` are significant. Except `long-term debt ratio`, other four significant variables are all positively related to `YTD`. Specifically, `ROA` has the greatest scale of coefficients, indicating with every 1% increase of `ROA`, the companies in wholesale industry will have 1.65% increase of `YTD`.

For `Long-term debt ratio`, which has negative correlation with `YTD`, it indicates that for wholesale industry companies, the less long-term debt ratio is, the higher its `YTD`. This correlation can be reconciled with the fact that wholesale companies with lower long-term debt ratios are healthier and performing better.

Retail Trade Industry

Data Merge

```
#merge industry 2019 and 2020 data
retail2 = pd.merge(retail, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
retail2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
retail2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	BBY	8857.000	15591.000	2229.000	32778.000	3395.000	2053.000	125.0	5174.000	8060.000	1541.000	503
1	CBRL	242.380	1581.225	36.884	2528.987	400.000	282.844	73.0	154.958	392.474	223.401	116
2	WEN	554.047	4994.529	300.195	597.530	3636.145	265.297	13.3	3.891	349.698	136.940	203
3	TGT	12902.000	42779.000	2577.000	54617.000	13613.000	4641.000	368.0	8992.000	14487.000	3281.000	2851
4	DG	5177.868	22825.084	240.320	18760.108	10731.121	2333.304	143.0	4676.848	4543.560	1712.555	1207

Check Outliers

```
#statistics description
retail2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	
count	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.0
mean	7209.600355	22720.151935	1563.693984	26363.661274	8524.109242	2157.712403	141.320419	3610.083645	7193.8
std	15764.594257	49125.561724	4825.030296	61080.035076	16321.534253	3991.268130	303.326452	6886.154134	16386.2
min	30.190000	166.113000	3.372000	41.105000	0.000000	-59.778000	0.412000	0.315000	32.9
25%	484.777000	1995.855000	86.473000	1413.407750	828.209750	153.499750	13.500000	135.338000	361.3
50%	1921.602000	5260.134000	270.257500	4464.149000	2538.202500	622.140500	46.250000	1290.350500	1377.4
75%	5140.572250	19485.850000	892.550000	15533.625000	8065.525000	2026.500000	128.000000	4123.584750	4527.0
max	96334.000000	236495.000000	36092.000000	383618.000000	83625.000000	19152.000000	2200.000000	44435.000000	87812.0

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in retail2.columns[1:]:
  retail2[col] = winsorize(retail2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
retail2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	
count	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.0000
mean	4990.176774	14970.984145	969.352968	20781.406694	7390.636823	1981.411984	106.084919	2993.668435	4914.118
std	6850.657397	19367.271399	1556.885514	36908.284245	12046.213907	3344.238639	128.375586	4251.093905	7552.1347
min	181.546000	730.721000	27.911000	352.481000	63.711000	23.495000	1.114000	3.891000	157.9230
25%	484.777000	1995.855000	86.473000	1413.407750	828.209750	153.499750	13.500000	135.338000	361.3522
50%	1921.602000	5260.134000	270.257500	4464.149000	2538.202500	622.140500	46.250000	1290.350500	1377.494
75%	5140.572250	19485.850000	892.550000	15533.625000	8065.525000	2026.500000	128.000000	4123.584750	4527.095
max	23485.000000	67598.000000	5683.000000	131394.000000	46875.900000	12702.000000	415.700000	14531.000000	25769.0000

```
# Add important useful varialbes back
retail2['SIC']=retail['SIC'].values
retail2['Group']=retail['Group'].values
retail2['Zip']=retail['Zip'].values
retail2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	BBY	8857.000	15591.000	2229.000	32778.000	3395.000	2053.000	125.0	5174.000	8060.000	1541.000	503
1	CBRL	242.380	1581.225	36.884	2528.987	400.000	282.844	73.0	154.958	392.474	223.401	116
2	WEN	554.047	4994.529	300.195	597.530	3636.145	265.297	13.3	3.891	349.698	136.940	203
3	TGT	12902.000	42779.000	2577.000	54617.000	13613.000	4641.000	368.0	8992.000	14487.000	3281.000	2851
4	DG	5177.868	22825.084	240.320	18760.108	10731.121	2333.304	143.0	4676.848	4543.560	1712.555	1207

Calculate Financial Ratios

```
# Calculate financial ratios
retail2['CashToAssets'] = (retail2.Cash/retail2.TA)
retail2['CashToInterest'] = 1/(retail2.Cash/retail2['Interest_Expense'])
retail2['CashToLiability'] = (retail2.Cash/retail2.CL)
retail2['CurrentRatio'] = (retail2.CA/retail2.CL)
retail2['QuickRatio'] = (retail2.CA-retail2.Inventory)/retail2.CL
retail2['InventoryTurnover'] = 1/(retail2.COGS/retail2.Inventory)
retail2['FixedAssetsTurnover'] = (retail2.Sales/retail2['PP&E'])
retail2['TotalAssetsTurnover'] = (retail2.Sales/retail2.TA)
retail2['LongTermDebtRatio'] = (retail2['Long_Term_Debt']/retail2.TA)
retail2['InterestCoverage'] = 1/(retail2.EBIT/retail2['Interest_Expense'])
retail2['ROA'] = (retail2.NI/retail2.TA)
retail2['SalesPerEmployee']=1/(retail2.Sales/(retail2.Employees*1000))
```

```
retail2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	BBY	8857.000	15591.000	2229.000	32778.000	3395.000	2053.000	125.0	5174.000	8060.000	1541.000	503
1	CBRL	242.380	1581.225	36.884	2528.987	400.000	282.844	73.0	154.958	392.474	223.401	116
2	WEN	554.047	4994.529	300.195	597.530	3636.145	265.297	13.3	3.891	349.698	136.940	203
3	TGT	12902.000	42779.000	2577.000	54617.000	13613.000	4641.000	368.0	8992.000	14487.000	3281.000	2851
4	DG	5177.868	22825.084	240.320	18760.108	10731.121	2333.304	143.0	4676.848	4543.560	1712.555	1207

```
# Delete unnecessary variables after using
retail2.drop(retail2.columns[1:14],axis=1,inplace=True)
retail2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000	62.000000
mean	-0.351305	56.306452	0.070625	0.420825	0.303614	1.388556	0.703901	0.271819
std	0.224729	2.440026	0.051753	0.597944	0.259738	0.758658	0.633395	0.514098
min	-0.746100	52.000000	0.002761	0.000000	0.009041	0.353909	-0.657679	0.005858
25%	-0.503625	54.000000	0.024381	0.054929	0.119469	0.913356	0.272945	0.080374
50%	-0.355700	57.000000	0.059038	0.211332	0.227453	1.149586	0.526408	0.157861
75%	-0.221325	58.000000	0.102465	0.601228	0.418536	1.602452	0.938837	0.269055
max	0.039900	59.000000	0.233922	3.785898	1.066491	3.993714	2.700082	3.660766

Industry Overview:

We have 62 companies for Retailing Industry. Overall, the `YTD`'s mean is negative with -0.35, which indicates that Retailing Industry is hurt severely by COVID-19 situation since 2020 even more than transportation industry.

The reverse of inventory turnover ratio(`InventoryTurnover`) for retailing is around 0.27, meaning that it is efficient. Also, the fixed assets turnover ratio has mean of 4.95, which demonstrates the fact that retailing industry is very efficient compared with other industries.

However, the mean of interest coverage ratio is only 0.22, which is much lower than 1.5, indicating the overall retailing industry does not have a good ability of paying its interest expense in short term.

OLS

```
import statsmodels.api as sm
```

```
#retail
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = retail2.iloc[:, i]
  y = retail2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=retail2.iloc[:, i].name
```

```
    name=retail2.iloc[:,i].name
    params=model.params[1]
    pvalue=model.pvalues[1]
    rsquared = model.rsquared
    data1.append(name)
    data2.append(params)
    data3.append(pvalue)
    data4.append(rsquared)
    summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	-0.121284	0.829385	0.000780
1	CashToInterest	0.043914	0.365783	0.013652
2	CashToLiability	-0.061611	0.582328	0.005071
3	CurrentRatio	0.024926	0.515536	0.007081
4	QuickRatio	0.001522	0.973598	0.000018
5	InventoryTurnover	0.003732	0.947497	0.000073
6	FixedAssetsTurnover	0.004920	0.321838	0.016359
7	TotalAssetsTurnover	0.116071	0.007029	0.114935
8	LongTermDebtRatio	-0.037961	0.677856	0.002896
9	InterestCoverage	-0.105760	0.093122	0.046279
10	ROA	0.548613	0.265493	0.020622
11	SalesPerEmployee	-0.005980	0.138212	0.036261

Analysis:
For single variabel OLS analysis, we can see that only `Total assets turnover` and `Interest Coverage` rates are significant. Also, its scales have economical effect on `YTD`. With higher total assets turnover, the companies in Retailing will have higher stock returns. In contrast, the higher Interest Coverage ratio will result in lower stock returns. In other words, in general,for companies in retailing industry, the harder the companies are able to cover its interest expenses, the higher `YTD` they will be. It reconciles the fact that most retailing companies are having low interest coverage ratios.

```
# Multi Regression
X = retail2.iloc[:,5:17]
y = retail2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.265
Model:	OLS	**Adj. R-squared:**	0.085
Method:	Least Squares	**F-statistic:**	1.470
Date:	Mon, 20 Apr 2020	**Prob (F-statistic):**	0.168
Time:	00:49:42	**Log-Likelihood:**	14.617
No. Observations:	62	**AIC:**	-3.234
Df Residuals:	49	**BIC:**	24.42
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.7386	0.165	-4.480	0.000	-1.070	-0.407
CashToAssets	-0.9978	1.193	-0.836	0.407	-3.395	1.400
CashToInterest	0.0578	0.060	0.968	0.338	-0.062	0.178
CashToLiability	0.0618	0.286	0.217	0.829	-0.512	0.636
CurrentRatio	0.0635	0.076	0.835	0.408	-0.089	0.216
QuickRatio	0.0412	0.118	0.348	0.730	-0.197	0.279
InventoryTurnover	0.0708	0.101	0.704	0.485	-0.131	0.273
FixedAssetsTurnover	-0.0024	0.007	-0.345	0.732	-0.017	0.012
TotalAssetsTurnover	0.1982	0.072	2.753	0.008	0.054	0.343
LongTermDebtRatio	0.0898	0.119	0.753	0.455	-0.150	0.330
InterestCoverage	-0.0851	0.079	-1.076	0.287	-0.244	0.074
ROA	-0.1233	0.663	-0.186	0.853	-1.455	1.209
SalesPerEmployee	-0.0019	0.005	-0.411	0.683	-0.011	0.007

Omnibus:	0.565	**Durbin-Watson:**	2.050
Prob(Omnibus):	0.754	**Jarque-Bera (JB):**	0.682
Skew:	0.087	**Prob(JB):**	0.711
Kurtosis:	2.517	**Cond. No.**	485.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Analysis:

For multiple regression, controlling all ratios, no variables are significant except `Total Asset Turnover`, which reconciles our single variable OLS analysis.

However, the adjusted Rsquared for retailing industry is only 0.085, indicating that our regression did a poor estimate for `YTD` in 2020 using 2019 data. It unveils the fact that COVID-19, as a black swan, has a huge unforseen effect on Retailing industry.

Finance Industry

Data Merge

```
#merge industry 2019 and 2020 data
finance2 = pd.merge(finance, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
finance2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
finance2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	AXP	1220.1695	198321.000	23932.000	36250.000	43003.000	9582.000	64.500	0.000	1270.0465	6759.000	48
1	AFL	1220.1695	152768.000	4896.000	17594.000	6565.000	4735.000	11.729	0.000	1270.0465	3304.000	5
2	AIG	1220.1695	525064.000	2856.000	37687.000	36083.000	6617.000	46.000	0.000	1270.0465	3348.000	26
3	ANAT	1220.1695	28597.566	452.001	3322.606	157.997	694.377	2.123	0.000	1270.0465	620.363	1
4	BPOP	1220.1695	52115.324	388.311	534.878	1146.637	1457.182	8.560	181.275	1270.0465	671.135	7

Check Outliers

```
#statistics description
finance2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	
count	273.000000	2.730000e+02	273.000000	273.000000	273.000000	273.000000	273.000000	273.000000	273.0
mean	2430.459661	8.593982e+04	2298.575890	5346.007718	9245.877853	2129.955315	12.839564	4516.533659	2398.6
std	6499.997589	2.967833e+05	9581.950661	17133.796898	32432.891105	6311.719856	36.610811	35540.591601	6738.6
min	58.826000	2.572000e+02	0.300000	1.888000	0.000000	-861.000000	0.000000	0.000000	13.4
25%	1220.169500	5.620319e+03	78.618000	97.257000	234.664000	147.499000	0.779000	0.000000	1270.0
50%	1220.169500	1.340062e+04	215.469000	493.031000	1117.151000	384.532000	2.123000	2.141000	1270.0
75%	1220.169500	3.788536e+04	808.036000	2071.955000	5531.000000	1401.601000	7.365000	58.162000	1270.0
max	67979.000000	2.687379e+06	133546.000000	184557.000000	261720.000000	63197.000000	325.000000	402390.000000	68816.0

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in finance2.columns[1:]:
  finance2[col] = winsorize(finance2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
finance2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL
count	273.000000	273.000000	273.000000	273.000000	273.000000	273.000000	273.000000	273.000000	273.000000
mean	1512.229467	49849.359084	1356.439121	3619.390176	5102.355974	1369.175033	8.856850	233.366297	1507.583603
std	1327.460206	92217.713061	2901.729528	7607.727369	8890.785488	2158.824038	15.458293	646.097789	1216.433989
min	607.600000	1338.986000	19.032000	27.078000	20.964000	39.728000	0.077000	0.000000	305.558000
25%	1220.169500	5620.319000	78.618000	97.257000	234.664000	147.499000	0.779000	0.000000	1270.046500
50%	1220.169500	13400.618000	215.469000	493.031000	1117.151000	384.532000	2.123000	2.141000	1270.046500
75%	1220.169500	37885.361000	808.036000	2071.955000	5531.000000	1401.601000	7.365000	58.162000	1270.046500
max	7068.000000	381508.000000	12123.000000	29360.000000	36083.000000	8144.000000	56.600000	2661.000000	6436.451000

```
# Add important useful varialbes back
finance2['SIC']=finance['SIC'].values
finance2['Group']=finance['Group'].values
finance2['Zip']=finance['Zip'].values
finance2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	AXP	1220.1695	198321.000	12123.000	29360.000	36083.000	8144.000	56.600	0.000	1270.0465	5369.000	48
1	AFL	1220.1695	152768.000	4896.000	17594.000	6565.000	4735.000	11.729	0.000	1270.0465	3304.000	5
2	AIG	1220.1695	381508.000	2856.000	29360.000	36083.000	6617.000	46.000	0.000	1270.0465	3348.000	26
3	ANAT	1220.1695	28597.566	452.001	3322.606	157.997	694.377	2.123	0.000	1270.0465	620.363	1
4	BPOP	1220.1695	52115.324	388.311	534.878	1146.637	1457.182	8.560	181.275	1270.0465	671.135	7

Calculate Financial Ratios

```
# Calculate financial ratios
finance2['CashToAssets'] = (finance2.Cash/finance2.TA)
finance2['CashToInterest'] = 1/(finance2.Cash/finance2['Interest_Expense'])
finance2['CashToLiability'] = (finance2.Cash/finance2.CL)
finance2['CurrentRatio'] = (finance2.CA/finance2.CL)
```

```
finance2['QuickRatio'] = (finance2.CA-finance2.Inventory)/finance2.CL
finance2['InventoryTurnover'] = 1/(finance2.COGS/finance2.Inventory)
finance2['FixedAssetsTurnover'] = (finance2.Sales/finance2['PP&E'])
finance2['TotalAssetsTurnover'] = (finance2.Sales/finance2.TA)
finance2['LongTermDebtRatio'] = (finance2['Long_Term_Debt']/finance2.TA)
finance2['InterestCoverage'] = 1/(finance2.EBIT/finance2['Interest_Expense'])
finance2['ROA'] = (finance2.NI/finance2.TA)
finance2['SalesPerEmployee']=1/(finance2.Sales/(finance2.Employees*1000))
```

```
finance2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	
0	AXP	1220.1695	198321.000	12123.000	29360.000	36083.000	8144.000	56.600	0.000	1270.0465	5369.000	48
1	AFL	1220.1695	152768.000	4896.000	17594.000	6565.000	4735.000	11.729	0.000	1270.0465	3304.000	5
2	AIG	1220.1695	381508.000	2856.000	29360.000	36083.000	6617.000	46.000	0.000	1270.0465	3348.000	26
3	ANAT	1220.1695	28597.566	452.001	3322.606	157.997	694.377	2.123	0.000	1270.0465	620.363	1
4	BPOP	1220.1695	52115.324	388.311	534.878	1146.637	1457.182	8.560	181.275	1270.0465	671.135	7

```
# Delete unnecessary variables after using
finance2.drop(finance2.columns[1:14],axis=1,inplace=True)
finance2.head()
```

	Ticker	YTD	SIC	Group	Zip	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	Inventor
0	AXP	-0.4088	61	7	10285	0.061128	0.067862	9.545320	0.960728	0.960728	
1	AFL	-0.3873	63	7	31999	0.032049	0.046569	3.854977	0.960728	0.960728	
2	AIG	-0.6014	63	7	10038	0.007486	0.288058	2.248737	0.960728	0.960728	
3	ANAT	-0.4510	63	7	77550	0.015806	0.000000	0.355893	0.960728	0.960728	
4	BPOP	-0.4596	60	7	918	0.007451	0.000000	0.305745	0.960728	0.817997	

```
#statistics description
finance2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	273.000000	273.000000	273.000000	273.000000	273.000000	273.000000	273.000000	273.00000
mean	-0.382985	62.175824	0.036116	0.619733	0.953641	1.045438	0.866102	0.36560
std	0.152047	2.641273	0.063457	1.574531	2.125765	0.350554	0.624916	1.07264
min	-0.610000	60.000000	0.001255	0.000000	0.014985	0.465368	-1.134471	0.00000
25%	-0.497800	60.000000	0.009421	0.000000	0.069807	0.960728	0.913397	0.00000
50%	-0.407900	61.000000	0.014618	0.000000	0.169654	0.960728	0.959042	0.01081
75%	-0.285100	63.000000	0.031777	0.273782	0.590530	0.960728	0.960728	0.31001
max	-0.043500	67.000000	0.462440	11.241099	9.545320	4.270368	4.160778	11.10176

Industry Overview

Here are 273 companies in total under Finance catergory.It is one of the largest group in the dataset. `YTD` of financial industry has a mean of negative -0.38, indicating the lowest stock return since 2020 by effects from COVID-19 across other industries.

Meanwhile, we think the ratios related to the interests, current assets(`CA`) and current liability(`CL`) wouldn't be so accurate, because we are missing some of the data from WRDS data resources. Although we tried lots of different method to replace the NaNs, we didn't find any better solution than replace with industrial median.

Overall, finance companies has a great operation in liquity and assest management. 2019 is still a bull stock market.

OLS

```
import statsmodels.api as sm
```

```
#finance
data1=[]
```

```
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = finance2.iloc[:, i]
  y = finance2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=finance2.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	0.492619	0.000631	0.042270
1	CashToInterest	-0.004718	0.421406	0.002387
2	CashToLiability	-0.007191	0.097350	0.010109
3	CurrentRatio	0.044019	0.094228	0.010300
4	QuickRatio	0.030343	0.039481	0.015553
5	InventoryTurnover	-0.013149	0.126282	0.008605
6	FixedAssetsTurnover	-0.000155	0.247684	0.004928
7	TotalAssetsTurnover	0.166319	0.000027	0.063042
8	LongTermDebtRatio	0.049621	0.268319	0.004520
9	InterestCoverage	-0.021565	0.243984	0.005006
10	ROA	1.206439	0.000010	0.069630
11	SalesPerEmployee	-0.000793	0.893852	0.000066

Analysis:

According to the summary result above, we notice that `Cash to Assets`, `Quick Ratio`, `Total Assets Turnover` and `ROA` are statistically and economically significant (with 0.05 threshold) determinants of finance stock returns. They are moving in the same direction with the stock returns because with the global spreading of the decrease, the economic development has been sharply harmed. Financial companies are mostly working closely with investment and trading. Their total assets are shrinking rapidly due to the stock market and global layoff. Thus, the balance sheet variables related to the current operations are affected mostly by the stock market crash. Special notes for the result, because we replaced lots of Current Asset and Current Liability missing data with median, the model result might not be so accurate. We think the real situation might be even worse than what we currently have.

```
# Multi Regression
X = finance2.iloc[:,5:17]
y = finance2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.115
Model:	OLS	**Adj. R-squared:**	0.074
Method:	Least Squares	**F-statistic:**	2.824
Date:	Mon, 20 Apr 2020	**Prob (F-statistic):**	0.00118
Time:	00:50:20	**Log-Likelihood:**	144.07
No. Observations:	273	**AIC:**	-262.1
Df Residuals:	260	**BIC:**	-215.2
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.3889	0.035	-11.093	0.000	-0.458	-0.320
CashToAssets	0.2400	0.210	1.142	0.254	-0.174	0.654
CashToInterest	-0.0079	0.008	-1.033	0.302	-0.023	0.007
CashToLiability	-0.0114	0.006	-1.959	0.051	-0.023	5.95e-05
CurrentRatio	-0.0018	0.041	-0.043	0.966	-0.083	0.079
QuickRatio	-0.0145	0.026	-0.567	0.571	-0.065	0.036
InventoryTurnover	-0.0112	0.010	-1.149	0.252	-0.030	0.008
FixedAssetsTurnover	-2.569e-05	0.000	-0.154	0.878	-0.000	0.000
TotalAssetsTurnover	0.0801	0.049	1.620	0.106	-0.017	0.178
LongTermDebtRatio	0.0467	0.065	0.721	0.471	-0.081	0.174
InterestCoverage	-0.0226	0.025	-0.890	0.374	-0.073	0.027
ROA	0.5989	0.358	1.671	0.096	-0.107	1.304
SalesPerEmployee	-0.0005	0.006	-0.074	0.941	-0.013	0.012

Omnibus:	6.215	**Durbin-Watson:**	2.103
Prob(Omnibus):	0.045	**Jarque-Bera (JB):**	6.380
Skew:	0.359	**Prob(JB):**	0.0412
Kurtosis:	2.786	**Cond. No.**	3.00e+03

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Analysis:
As we expected, the finance industry adjusted r squared runs nearly the worst among all 9 industries we analyzed with (0.074, only slightly higher than manufacturing industry). Based on the P-value result from the multiple regression, reversed `Cash to Liability`,`ROA` and `Total Asset Turnover` ratios are related to stock return. That also reveals the financial industry is highly related to the stock market. The result is reasonable because with the COVID-19, in these short four months, we saw lots of unpredictable events happened at the same time. President Trumps's announcements, CDC's actions, and oil prices drop all affected the investors' confidence in the stock market.

Service Industry

Data Merge

```
#merge industry 2019 and 2020 data
service2 = pd.merge(service, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
service2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
service2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP
0	ADSK	2659.300	6179.300	1774.700	238.600	2046.800	366.800	10.10	0.000	3219.200	214.500	600.5
1	ADP	34342.300	41887.700	1949.200	7677.600	2002.200	3080.800	58.00	0.000	32627.700	2292.800	764.2
2	RHI	1628.849	2311.408	270.478	3479.649	202.200	620.390	221.60	0.000	940.692	454.433	369.4
3	MGLN	1673.043	3092.173	325.249	6932.257	724.542	95.657	10.10	44.962	910.147	55.902	188.8
4	CHDN	221.200	2551.000	96.200	880.300	1491.100	233.800	0.55	0.000	301.200	137.500	937.3

Check Outliers

```
#statistics description
service2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	
count	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	225.00
mean	4231.873404	10616.314031	907.609129	2398.350738	2917.347538	1000.198622	23.902340	55.964667	2428.79
std	16661.303517	31035.316260	2231.266928	6386.557421	7703.751195	4320.652040	61.317261	255.953781	6971.18
min	5.520000	17.837000	0.402000	0.000000	0.000000	-1226.225000	0.037000	0.000000	12.13
25%	352.457000	1014.552000	93.628000	167.740000	120.497000	11.933000	1.987000	0.000000	204.54
50%	920.707000	2370.900000	236.232000	487.410000	572.944000	128.063000	5.949500	0.000000	573.13
75%	2134.200000	7335.358000	686.000000	1744.919000	2062.670000	566.612000	17.500000	5.738000	1752.45
max	175552.000000	286556.000000	19079.000000	60245.000000	79107.000000	42959.000000	492.000000	2063.000000	69420.00

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in service2.columns[1:]:
  service2[col] = winsorize(service2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
service2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	
count	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	
mean	2273.599391	7121.107227	684.402982	1629.133760	2170.480684	473.888529	15.496411	15.473787	1619.823844	
std	3558.218092	10611.670669	1073.063985	2470.575844	3497.489900	789.536866	21.727740	37.148995	2605.605093	
min	147.223000	372.279000	18.032000	46.200000	0.109000	-170.946000	0.779000	0.000000	55.754000	-
25%	352.457000	1014.552000	93.628000	167.740000	120.497000	11.933000	1.987000	0.000000	204.544000	
50%	920.707000	2370.900000	236.232000	487.410000	572.944000	128.063000	5.949500	0.000000	573.139000	
75%	2134.200000	7335.358000	686.000000	1744.919000	2062.670000	566.612000	17.500000	5.738000	1752.453000	
max	14584.700000	41887.700000	4226.000000	8729.100000	12502.700000	2812.000000	81.000000	141.518000	10382.000000	2

```
# Add important useful varialbes back
service2['SIC']=service['SIC'].values
service2['Group']=service['Group'].values
service2['Zip']=service['Zip'].values
service2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP
0	ADSK	2659.300	6179.300	1774.700	238.600	2046.800	366.800	10.100	0.000	3219.200	214.500	600.5
1	ADP	14584.700	41887.700	1949.200	7677.600	2002.200	2812.000	58.000	0.000	10382.000	2292.800	764.2
2	RHI	1628.849	2311.408	270.478	3479.649	202.200	620.390	81.000	0.000	940.692	454.433	369.4
3	MGLN	1673.043	3092.173	325.249	6932.257	724.542	95.657	10.100	44.962	910.147	55.902	188.8
4	CHDN	221.200	2551.000	96.200	880.300	1491.100	233.800	0.779	0.000	301.200	137.500	937.3

Calculate Financial Ratios

```
# Calculate financial ratios
service2['CashToAssets'] = (service2.Cash/service2.TA)
service2['CashToInterest'] = 1/(service2.Cash/service2['Interest_Expense'])
service2['CashToLiability'] =(service2.Cash/service2.CL)
service2['CurrentRatio'] = (service2.CA/service2.CL)
```

```
service2['QuickRatio'] = (service2.CA-service2.Inventory)/service2.CL
service2['InventoryTurnover'] = 1/(service2.COGS/service2.Inventory)
service2['FixedAssetsTurnover'] = (service2.Sales/service2['PP&E'])
service2['TotalAssetsTurnover'] = (service2.Sales/service2.TA)
service2['LongTermDebtRatio'] = (service2['Long_Term_Debt']/service2.TA)
service2['InterestCoverage'] = 1/(service2.EBIT/service2['Interest_Expense'])
service2['ROA'] = (service2.NI/service2.TA)
service2['SalesPerEmployee']=1/(service2.Sales/(service2.Employees*1000))
```

```
service2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP
0	ADSK	2659.300	6179.300	1774.700	238.600	2046.800	366.800	10.100	0.000	3219.200	214.500	600.5
1	ADP	14584.700	41887.700	1949.200	7677.600	2002.200	2812.000	58.000	0.000	10382.000	2292.800	764.2
2	RHI	1628.849	2311.408	270.478	3479.649	202.200	620.390	81.000	0.000	940.692	454.433	369.4
3	MGLN	1673.043	3092.173	325.249	6932.257	724.542	95.657	10.100	44.962	910.147	55.902	188.8
4	CHDN	221.200	2551.000	96.200	880.300	1491.100	233.800	0.779	0.000	301.200	137.500	937.3

```
# Delete unnecessary variables after using
service2.drop(service2.columns[1:14],axis=1,inplace=True)
service2.head()
```

	Ticker	YTD	SIC	Group	Zip	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	Inventor
0	ADSK	-0.2423	73	8	94903	0.287201	0.000000	0.551286	0.826075	0.826075	
1	ADP	-0.2459	73	8	7068	0.046534	0.066643	0.187748	1.404806	1.404806	
2	RHI	-0.3987	73	8	94025	0.117019	0.000000	0.287531	1.731543	1.731543	
3	MGLN	-0.4667	87	8	85034	0.105185	0.111155	0.357359	1.838212	1.788811	
4	CHDN	-0.4136	79	8	40222	0.037711	0.737006	0.319389	0.734396	0.734396	

```
service2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover
count	225.00000	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000	225.000000
mean	-0.27907	75.191111	0.137476	0.296136	0.696980	1.895291	1.871889	0.018344
std	0.19964	4.392964	0.124663	0.744919	0.764347	1.184515	1.177046	0.059854
min	-0.63090	70.000000	0.002741	0.000000	0.023658	0.468324	0.468324	0.000000
25%	-0.42880	73.000000	0.048775	0.018683	0.244905	1.169627	1.134980	0.000000
50%	-0.26910	73.000000	0.101171	0.100593	0.431943	1.515876	1.513464	0.000000
75%	-0.14720	75.000000	0.180226	0.268475	0.848138	2.416341	2.398153	0.012084
max	0.08720	87.000000	0.634871	9.061038	5.171215	9.153507	9.153507	0.535278

Industry Overview

There are 225 companies in Service industry. The `YTD` mean is -0.27, indicating that it has negative stock returns since 2020, and implies that it receives negative effects from COVID. However, compared with other industries' YTD in 2020, Service Industry has the smallest decrease YTD.

Specifically, service industry has very low reservse of inventory turnover ratio (`InventoryTurnover`), whose mean is 0.018. In other words, its inventory turnover ratio is around 55, which is the slowest one company compared with all industries in converting inventories into sales. We think it is due to the properties of service industry: most of services industry does not have too many inventories, except its storage of beverages and foods, such as Hotels.

Also, the reverse of interest coverage ratio (`interestCoverage`) has mean of 0.13, indicating its interest coverage ratio is around 7, indicating high abilities of paying liabilites in short term. Specifically, according to industry ratios in 2019, legal services in services industry has the greatest positive interest coverage ratio, which is around 47.87.

OLS

```
import statsmodels.api as sm
```

```
#service
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,17):
  X = service2.iloc[:, i]
  y = service2['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=service2.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	0.315510	2.998025e-03	0.038816
1	CashToInterest	-0.068123	1.156731e-04	0.064612
2	CashToLiability	0.035102	4.403135e-02	0.018061
3	CurrentRatio	0.035387	1.538949e-03	0.044083
4	QuickRatio	0.035962	1.378265e-03	0.044956
5	InventoryTurnover	0.068789	7.583411e-01	0.000425
6	FixedAssetsTurnover	0.001761	1.783473e-01	0.008108
7	TotalAssetsTurnover	-0.053618	1.224245e-01	0.010666
8	LongTermDebtRatio	-0.335047	3.239989e-08	0.128323
9	InterestCoverage	-0.041604	8.166161e-02	0.013532
10	ROA	0.072268	5.251999e-01	0.001813
11	SalesPerEmployee	-0.002392	3.014783e-01	0.004787

Analysis:

According to the summary result above, we noticed that `Cash to assets`,`Cash to Interest`, `LongTermDebtRatio` and `Current Ratio` are statistically and economically significant determinants of stock returns. From the above variables, `LongTermDebtRatio` is positively correlated to `YTD`. It indicates that it takes longer for the service company to convert the inventories to revenues. It moving in the same direction as `YTD`. Due to COVID-19, the service industry is facing huge challenges in maintaining their operations. Duet the loss of customers and closed for all physical stores. Even many big companies like AMC movie theater are plan to start their bankruptcy protection. Their long term debt will be big issues and some companies might not be able to pay back the money.

```
X = service2.iloc[:,5:17]
y = service2['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.216
Model:	OLS	**Adj. R-squared:**	0.172
Method:	Least Squares	**F-statistic:**	4.877
Date:	Mon, 20 Apr 2020	**Prob (F-statistic):**	4.35e-07
Time:	00:50:56	**Log-Likelihood:**	71.194
No. Observations:	225	**AIC:**	-116.4
Df Residuals:	212	**BIC:**	-71.98
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.2066	0.042	-4.870	0.000	-0.290	-0.123
CashToAssets	0.3944	0.166	2.371	0.019	0.067	0.722
CashToInterest	-0.0318	0.018	-1.759	0.080	-0.067	0.004
CashToLiability	-0.0741	0.034	-2.191	0.030	-0.141	-0.007
CurrentRatio	0.1900	0.279	0.680	0.497	-0.361	0.741
QuickRatio	-0.1563	0.280	-0.558	0.578	-0.709	0.396
InventoryTurnover	-0.2508	0.329	-0.762	0.447	-0.900	0.398
FixedAssetsTurnover	0.0015	0.001	1.016	0.311	-0.001	0.004
TotalAssetsTurnover	-0.1213	0.040	-3.007	0.003	-0.201	-0.042
LongTermDebtRatio	-0.2593	0.064	-4.024	0.000	-0.386	-0.132
InterestCoverage	-0.0127	0.023	-0.556	0.579	-0.058	0.032
ROA	0.1522	0.113	1.352	0.178	-0.070	0.374
SalesPerEmployee	0.0010	0.002	0.442	0.659	-0.004	0.006

Omnibus:	2.115	**Durbin-Watson:**	1.873
Prob(Omnibus):	0.347	**Jarque-Bera (JB):**	2.033
Skew:	0.160	**Prob(JB):**	0.362
Kurtosis:	2.661	**Cond. No.**	573.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Analysis: With the application of multiple regression on the service industry, `Cash to assets`, `Cash to Liability`, `LongTermDebt Ratio` and `TotalAssetsTurnover` are statistically and economically significant determinants of stock returns, which make sense in the service industry under the COVID-19 situation.

In reality, the service industry have suffered heavy losses in the epidemic. The "Stay-at-home" cost the company a large number of customers. Relatively speaking, this is an impact on the overall industry, so there will be no significant changes due to individual differences. This is why adjusted Rsquared is only 0.17.

Public Administration Industry

Data Merge

```
#merge industry 2019 and 2020 data
administration2 = pd.merge(administration, df3, how='left', on='Ticker')
#drop those columns are not affect by outliers
administration2.drop(['Zip','SIC','Group'],axis=1,inplace=True)
administration2
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&E	S
0	HON	24303.0	58679.0	9067.0	22975.0	11644.0	7397.0	113.000	4421.0	18098.0	6143.0	5998.0	36
1	BRK.B	24303.0	817729.0	64175.0	211678.0	98830.0	105481.0	391.539	19852.0	18098.0	81417.0	180282.0	327
2	GE	24303.0	266048.0	36394.0	67328.0	70403.0	8902.0	205.000	14104.0	18098.0	-4979.0	46186.0	95
3	IEP	24303.0	24639.0	3794.0	8280.0	8839.0	-1386.0	28.033	1812.0	18098.0	-1098.0	5163.0	8

Check Outliers

```
#statistics description
administration2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	
count	4.0	4.000000	4.00000	4.000000	4.000000	4.000000	4.000000	4.000000	4.0	
mean	24303.0	291773.750000	28357.50000	77565.250000	47429.000000	30098.500000	184.393000	10047.250000	18098.0	2037
std	0.0	366508.242904	27826.65126	92864.726462	44495.774339	50459.372307	155.862679	8407.623024	0.0	4095
min	24303.0	24639.000000	3794.00000	8280.000000	8839.000000	-1386.000000	28.033000	1812.000000	18098.0	-497
25%	24303.0	50169.000000	7748.75000	19301.250000	10942.750000	5201.250000	91.758250	3768.750000	18098.0	-206
50%	24303.0	162363.500000	22730.50000	45151.500000	41023.500000	8149.500000	159.000000	9262.500000	18098.0	252
75%	24303.0	403968.250000	43339.25000	103415.500000	77509.750000	33046.750000	251.634750	15541.000000	18098.0	2496
max	24303.0	817729.000000	64175.00000	211678.000000	98830.000000	105481.000000	391.539000	19852.000000	18098.0	8141

Note that there appears to be big outliers (common for financial ratios)
Evidence: Large difference in mean and median of variables. The maximum and minimum are really far away from the IQR.

Transform Variables

```
from scipy.stats.mstats import winsorize
from scipy import stats
# Convert variables
# Using Winsorize to convert outliers
for col in administration2.columns[1:]:
  administration2[col] = winsorize(administration2[col], limits=[0.05,0.05], inplace=True)
```

```
#statistics description recheck
administration2.describe()
```

	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	
count	4.0	4.000000	4.00000	4.000000	4.000000	4.000000	4.000000	4.000000	4.0	
mean	24303.0	291773.750000	28357.50000	77565.250000	47429.000000	30098.500000	184.393000	10047.250000	18098.0	2037
std	0.0	366508.242904	27826.65126	92864.726462	44495.774339	50459.372307	155.862679	8407.623024	0.0	4095
min	24303.0	24639.000000	3794.00000	8280.000000	8839.000000	-1386.000000	28.033000	1812.000000	18098.0	-497
25%	24303.0	50169.000000	7748.75000	19301.250000	10942.750000	5201.250000	91.758250	3768.750000	18098.0	-206
50%	24303.0	162363.500000	22730.50000	45151.500000	41023.500000	8149.500000	159.000000	9262.500000	18098.0	252
75%	24303.0	403968.250000	43339.25000	103415.500000	77509.750000	33046.750000	251.634750	15541.000000	18098.0	2496
max	24303.0	817729.000000	64175.00000	211678.000000	98830.000000	105481.000000	391.539000	19852.000000	18098.0	8141

```
# Add important useful varialbes back
administration2['SIC']=administration['SIC'].values
administration2['Group']=administration['Group'].values
administration2['Zip']=administration['Zip'].values
administration2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&E	
0	HON	24303.0	58679.0	9067.0	22975.0	11644.0	7397.0	113.000	4421.0	18098.0	6143.0	5998.0	36
1	BRK.B	24303.0	817729.0	64175.0	211678.0	98830.0	105481.0	391.539	19852.0	18098.0	81417.0	180282.0	327
2	GE	24303.0	266048.0	36394.0	67328.0	70403.0	8902.0	205.000	14104.0	18098.0	-4979.0	46186.0	95
3	IEP	24303.0	24639.0	3794.0	8280.0	8839.0	-1386.0	28.033	1812.0	18098.0	-1098.0	5163.0	8

Calculate Financial Ratios

```
# Calculate financial ratios
administration2['CashToAssets'] = (administration2.Cash/administration2.TA)
administration2['CashToInterest'] = 1/(administration2.Cash/administration2['Interest_Expense'])
administration2['CashToLiability'] =(administration2.Cash/administration2.CL)
administration2['CurrentRatio'] = (administration2.CA/administration2.CL)
administration2['QuickRatio'] = (administration2.CA-administration2.Inventory)/administration2.CL
administration2['InventoryTurnover'] = 1/(administration2.COGS/administration2.Inventory)
```

```
administration2['InventoryTurnover'] = 1/(administration2.COGS/administration2.Inventory)
administration2['FixedAssetsTurnover'] = (administration2.Sales/administration2['PP&E'])
administration2['TotalAssetsTurnover'] = (administration2.Sales/administration2.TA)
administration2['LongTermDebtRatio'] = (administration2['Long_Term_Debt']/administration2.TA)
administration2['InterestCoverage'] = 1/(administration2.EBIT/administration2['Interest_Expense'])
administration2['ROA'] = (administration2.NI/administration2.TA)
administration2['SalesPerEmployee']=1/(administration2.Sales/(administration2.Employees*1000))
```

```
administration2.head()
```

	Ticker	CA	TA	Cash	COGS	Long_Term_Debt	EBIT	Employees	Inventory	CL	NI	PP&E	S
0	HON	24303.0	58679.0	9067.0	22975.0	11644.0	7397.0	113.000	4421.0	18098.0	6143.0	5998.0	36
1	BRK.B	24303.0	817729.0	64175.0	211678.0	98830.0	105481.0	391.539	19852.0	18098.0	81417.0	180282.0	327
2	GE	24303.0	266048.0	36394.0	67328.0	70403.0	8902.0	205.000	14104.0	18098.0	-4979.0	46186.0	95
3	IEP	24303.0	24639.0	3794.0	8280.0	8839.0	-1386.0	28.033	1812.0	18098.0	-1098.0	5163.0	8

```
# Delete unnecessary variables after using
administration2.drop(administration2.columns[1:14],axis=1,inplace=True)
administration2.head()
```

	Ticker	YTD	SIC	Group	Zip	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	Inventor
0	HON	-0.2799	99	9	28202	0.154519	0.039374	0.500995	1.342856	1.098574	
1	BRK.B	-0.2126	99	9	68131	0.078480	0.061722	3.545972	1.342856	0.245939	
2	GE	-0.3970	99	9	2210	0.136795	0.109139	2.010940	1.342856	0.563543	
3	IEP	-0.2631	99	9	10153	0.153984	0.159462	0.209636	1.342856	1.242734	

```
administration2.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnover	Fixed
count	4.000000	4.0	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000	
mean	-0.288150	99.0	0.130944	0.092424	1.566886	1.342856	0.787698	0.178633	
std	0.077999	0.0	0.035932	0.053324	1.537554	0.000000	0.464561	0.057613	
min	-0.397000	99.0	0.078480	0.039374	0.209636	1.342856	0.245939	0.093784	
25%	-0.309175	99.0	0.122216	0.056135	0.428155	1.342856	0.484142	0.167766	
50%	-0.271500	99.0	0.145389	0.085430	1.255968	1.342856	0.831059	0.200954	
75%	-0.250475	99.0	0.154117	0.121720	2.394698	1.342856	1.134614	0.211822	
max	-0.212600	99.0	0.154519	0.159462	3.545972	1.342856	1.242734	0.218841	

Industry Overview

Here are only 4 companies in total under Public Administration catergory, which is the smallest group in the dataset. `YTD` of Public Administration industry has a mean of negative -0.28, indicating lower stock return since 2020 by effects from COVID-19. The mean of `Fixed Assets Turnover` is 2.92 which indicates the industry is pretty efficient on the utilization of fixed assets.

OLS

```
import statsmodels.api as sm
```

Since Public Administration Industry only has 4 tickers, and 3 of them are replaced by one when replace missing values in CA and CL. Thus we will not use those ratios since they are totally the same across the Administration industry, and are not helpful for our analysis.

```
## drop ratios contain inf or nans since those are not meaningful
administration3=administration2.drop(['CashToLiability','CurrentRatio','CurrentRatio'],axis=1)
```

```
#administration
data1=[]
data2=[]
data3=[]
data4=[]
for i in range(5,14):
```

```
for i in range(5,14):
  X = administration3.iloc[:, i]
  y = administration3['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=administration3.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	-0.988534	0.544611	0.207379
1	CashToInterest	-0.339642	0.767803	0.053916
2	QuickRatio	-0.003965	0.976384	0.000558
3	InventoryTurnover	-0.868379	0.358580	0.411419
4	FixedAssetsTurnover	0.000403	0.988948	0.000122
5	TotalAssetsTurnover	0.105938	0.824555	0.030781
6	LongTermDebtRatio	-0.296211	0.616758	0.146875
7	InterestCoverage	-0.143875	0.334009	0.443544
8	ROA	0.510217	0.489941	0.260160

Analysis:

Because there are only 4 observations in public administration industry which gives multiple regression too less data, we decide to do the single regression only.

According to the single variable OLS summary above, no variables are significant in statistical sense. From economical perspective, most variables are negative. For instance, cash to interest, cash to assets and interest coverage ratio. Such negative correlations indicating that administration industry have unusual behaviors compared with other industries. Specifically, its `YTD` return would increase as its ability of paying interest by cash increase. We assume that it is caused by the fact that we only have 4 observations. Also, since there are not enough obervations in this industry, the rsquared wouldn't be convincing enough, and we will be needing more data to do a deeper analysis to see how COV-19 is affecting the stock returns.

Combine Industry Datasets

```
data = pd.concat([mining2, construction2, manufacturing2, transportation2, wholesale2, retail2, finance2, service2, a
```

```
data.head()
```

	Ticker	YTD	SIC	Group	Zip	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	Inventor
0	HES	-0.4983	13	1	10036	0.070930	0.270550	0.615538	1.257371	1.153386	
1	APA	-0.7898	13	1	77056	0.013641	1.769231	0.133154	1.057143	0.786523	
2	HAL	-0.6890	13	1	77032	0.089372	0.261023	0.464945	2.298483	1.654982	
3	HP	-0.6597	13	1	74119	0.059584	0.067505	0.848149	2.718144	2.353349	
4	MRO	-0.7312	13	1	77056-2723	0.042381	0.326340	0.491691	1.223496	1.182235	

```
data.isna().sum()
```

```
Ticker                  0
YTD                     0
SIC                     0
Group                   0
Zip                     0
CashToAssets            0
CashToInterest          0
CashToLiability         0
CurrentRatio            0
QuickRatio              0
InventoryTurnover       0
FixedAssetsTurnover     0
TotalAssetsTurnover     0
LongTermDebtRatio       0
InterestCoverage        0
ROA                     0
SalesPerEmployee        0
dtype: int64
```

```
#statistics description check
data.describe()
```

	YTD	SIC	CashToAssets	CashToInterest	CashToLiability	CurrentRatio	QuickRatio	InventoryTurnov
count	1216.000000	1216.000000	1216.000000	1216.000000	1216.000000	1216.000000	1216.000000	1216.000
mean	-0.319963	48.985197	0.113857	0.704915	0.994922	2.391219	2.013177	0.233
std	0.196885	19.012401	0.146245	2.532957	1.866415	3.106430	3.319168	0.597
min	-0.899800	10.000000	0.000375	0.000000	0.004280	0.146189	-28.079551	0.000
25%	-0.457825	34.000000	0.019333	0.004107	0.146229	0.960728	0.907068	0.000
50%	-0.326800	49.000000	0.064136	0.104300	0.389704	1.406910	1.129765	0.079
75%	-0.184275	63.000000	0.144403	0.366929	1.047614	2.640582	2.152606	0.263
max	0.098000	99.000000	0.973483	36.210277	21.210511	66.088990	65.659864	11.101

Dataset Overview

In the last part of our analysis, we combined the subset as a whole and reviewed the changes that happened around us stock market. There are a total of 1216 companies in the dataset we are looking at. The mean return is negative due to the COVID-19 impact. As we can see from the statistics, the Cash the liquidity ratio is affected, with the global layoff and unemployment rate rising, the economic activities are low. The government is trying to apply fiscal policy to stimulate the economy. We saw a lot of things happen in these short four months, and still in an unstable period.

```
#data.to_csv("project_data.csv",index=False,header=True)
# could uncomment the code line below to download the data
```

```
#Check OLS for entire dataset
data1=[]
data2=[]
data3=[]
data4=[]
# all_inf_or_nan = data.isin([np.inf, -np.inf, np.nan]).all(axis='columns')
# data=data[~all_inf_or_nan]
for i in range(5,17):
  X = data.iloc[:, i]
  y = data['YTD']
  X = sm.add_constant(X) # adding a constant
  model = sm.OLS(y,X).fit()
  summary=model.summary()
  name=data.iloc[:, i].name
  params=model.params[1]
  pvalue=model.pvalues[1]
  rsquared = model.rsquared
  data1.append(name)
  data2.append(params)
  data3.append(pvalue)
  data4.append(rsquared)
  summary_7 = pd.DataFrame(list(zip(data1, data2,data3,data4)), columns =['variable', 'params','pvalue','rsquared'])
summary_7
```

	variable	params	pvalue	rsquared
0	CashToAssets	0.214327	2.385527e-08	0.025345
1	CashToInterest	-0.004818	3.066228e-02	0.003842
2	CashToLiability	0.008066	7.644353e-03	0.005846
3	CurrentRatio	0.010913	1.509160e-09	0.029648
4	QuickRatio	0.010090	2.386429e-09	0.028934
5	InventoryTurnover	-0.004596	6.268093e-01	0.000195
6	FixedAssetsTurnover	-0.000221	1.720352e-01	0.001536
7	TotalAssetsTurnover	0.023742	1.882319e-02	0.004537
8	LongTermDebtRatio	-0.065239	5.277208e-03	0.006392
9	InterestCoverage	-0.003272	2.507349e-01	0.001087
10	ROA	0.076019	3.789096e-02	0.003545
11	SalesPerEmployee	-0.000442	1.317555e-01	0.001870

Analysis

When looked at the dataset as a whole, we are surprised to see all the variables are significant to the model. `Cash To Assets`, `Current Ratio` and `Quick Ratio` are contributed most to the model based on r squared result. Due to the combining dataset, the r square result overall is lower. Because we have the data from all 9 industries together and they are influences the model result accuracy.

```
#Multi Regression
X = data.iloc[:,5:17]
y = data['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.072
Model:	OLS	**Adj. R-squared:**	0.063
Method:	Least Squares	**F-statistic:**	7.788
Date:	Mon, 20 Apr 2020	**Prob (F-statistic):**	4.19e-14
Time:	00:52:11	**Log-Likelihood:**	296.72
No. Observations:	1216	**AIC:**	-567.4
Df Residuals:	1203	**BIC:**	-501.1
Df Model:	12		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.3500	0.012	-29.023	0.000	-0.374	-0.326
CashToAssets	0.2500	0.052	4.786	0.000	0.148	0.352
CashToInterest	-0.0008	0.002	-0.371	0.711	-0.005	0.004
CashToLiability	-0.0089	0.004	-2.164	0.031	-0.017	-0.001
CurrentRatio	0.0059	0.005	1.153	0.249	-0.004	0.016
QuickRatio	0.0041	0.005	0.881	0.378	-0.005	0.013
InventoryTurnover	-0.0053	0.009	-0.564	0.573	-0.024	0.013
FixedAssetsTurnover	-0.0002	0.000	-1.248	0.212	-0.001	0.000
TotalAssetsTurnover	0.0161	0.011	1.483	0.138	-0.005	0.037
LongTermDebtRatio	-0.0581	0.024	-2.461	0.014	-0.104	-0.012
InterestCoverage	-0.0003	0.003	-0.125	0.900	-0.006	0.005
ROA	0.1541	0.040	3.805	0.000	0.075	0.233
SalesPerEmployee	-0.0007	0.000	-2.318	0.021	-0.001	-0.000

Omnibus:	12.090	**Durbin-Watson:**	1.789
Prob(Omnibus):	0.002	**Jarque-Bera (JB):**	9.251
Skew:	0.114	**Prob(JB):**	0.00980
Kurtosis:	2.638	**Cond. No.**	366.

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Analysis

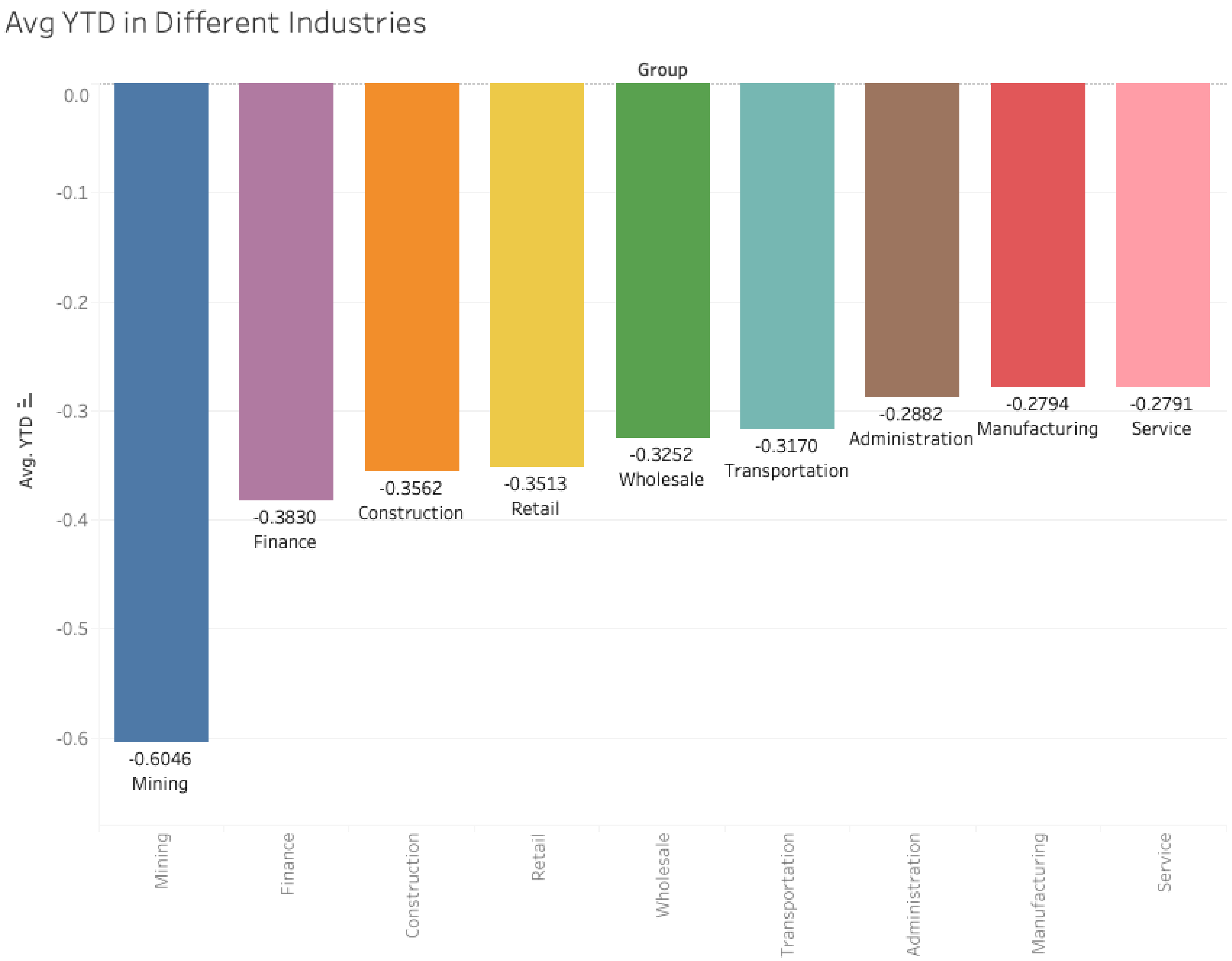
The whole dataset model r square is **0.072**. `Cash to Liability` and `Long-term Debt Ratio` negatively impact the model. In summary, in the industry level, the rsquared adjust for multiple regression:

1. Wholesales Industry **0.55**
2. Construction Industry **0.54**
3. Mining Industry **0.22**
4. Service Industry **0.17**
5. Transportation Industry **0.1**
6. Retail Industry **0.08**
7. Finance Industry **0.074**
8. Manufacturing Industry **0.06**

The whole dataset model predicts is not as well as industrial level prediction. But still, the influence is across all the industry. Some companies are struggling with surviving, but we also saw lots of possibilities in the zoom, CVS and grocery stores.

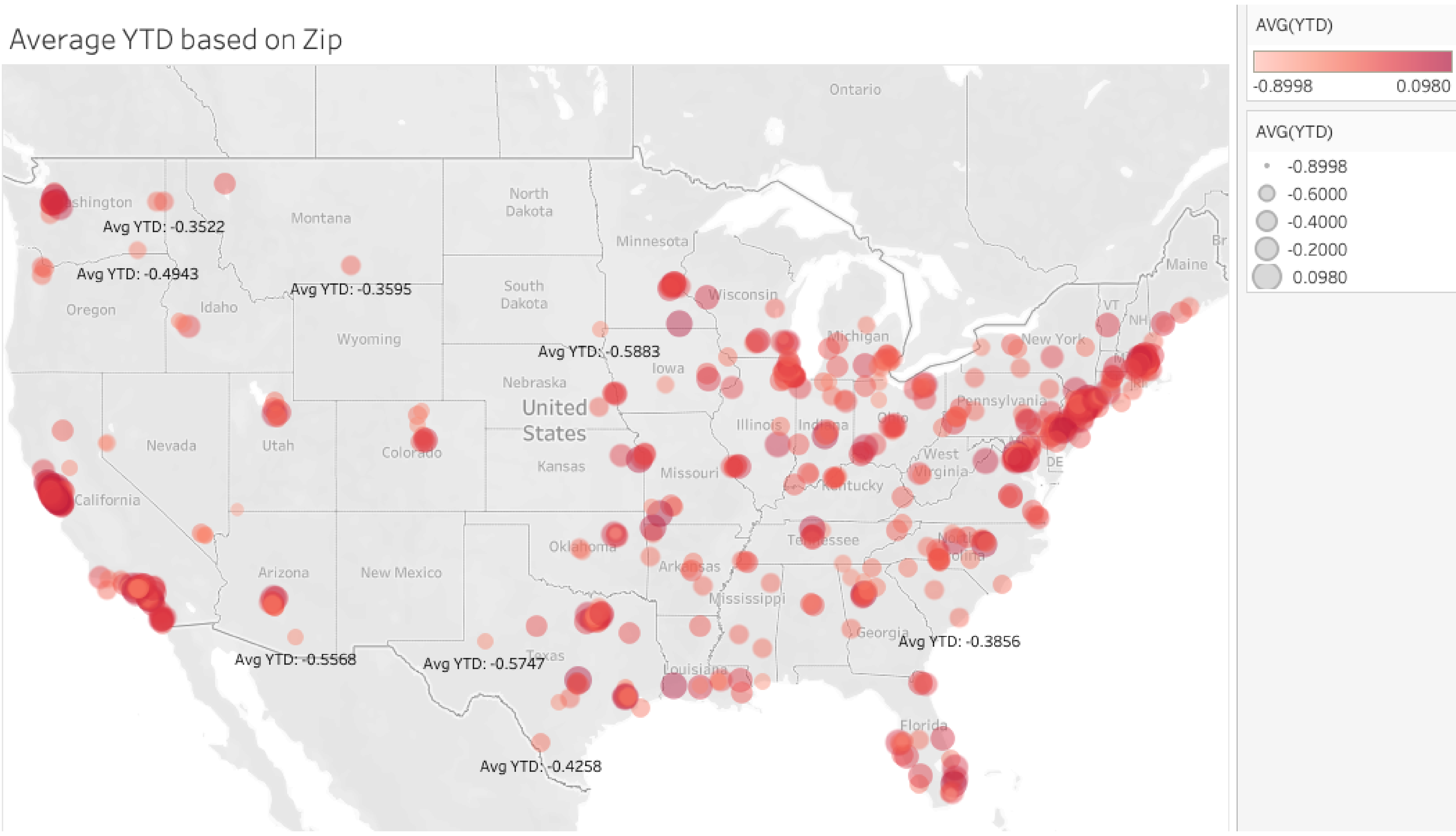
Visualization

Average YTD Comparison based on Industries



Based on this graph above, we could find out that on average, all industries are negatively affected by the COVID-19. Mining Industry has the largest negative effect by the COVID-19, and those followed by Finance Industry, Construction Industry, Retail Industry, Wholesale Industry, Transportation Industry, and Administration Industry. Manufacturing and Service Industry has the smallest negative effect by the COVID-19. Those findings connected the result for each industry finding that we have done.

Average YTD Comparison based on Zip



Based on the graph above, it shows what happened to YTD based on Zip Code. The lighter color area means the larger negative effect on the Stock market by the COVID-19. Overall, we can see most companies(tickers) have been negatively affected, and even though COVID-19 brings some positive effect to some companies, we can see it's very tiny since those AVG(YTD) are all less than 0.1 which is close to 0. Dive into deeply, 94949(Marin County, CA), 94041(Mountain View, CA), 98021(Snohomish County, WA) are some zip code have positive average YTD. As we guess, those may because there are more tech firms which are more important when people work remotely or work at home.

Average YTD Comparison based on States

Combine Data for State Level Visualization

```
from google.colab import files
uploaded = files.upload()
```

Choose Files state_level.csv

- **state_level.csv**(text/csv) - 36233 bytes, last modified: 4/18/2020 - 100% done

```
Saving state_level.csv to state_level.csv
```

```
import pandas as pd
import numpy as np
YTD = pd.read_csv('state_level.csv')
pd.DataFrame.from_records(YTD)
YTD.head()
```

	Ticker	YTD	SIC	Group	Zip	state	state2
0	HES	-0.4983	13	1	10036	NY	New York
1	APA	-0.7898	13	1	77056	TX	Texas
2	HAL	-0.6890	13	1	77032	TX	Texas
3	HP	-0.6597	13	1	74119	OK	Oklahoma
4	MRO	-0.7312	13	1	77056	TX	Texas

`state_level.csv` is a file we use after manually remove foreign countries since for the geolocation analysis we want to focus on only the United States. We also transfer those Zip into State in that file. Also to match that `States Reporting Cases of COVID-19 to CDC_.csv` file we find from the CDC web page that only has the records for the US.

```
from google.colab import files
uploaded = files.upload()
```

Choose Files States Report... to CDC_.csv

- **States Reporting Cases of COVID-19 to CDC_.csv**(text/csv) - 4991 bytes, last modified: 4/18/2020 - 100% done

```
Saving States Reporting Cases of COVID-19 to CDC_.csv to States Reporting Cases of COVID-19 to CDC_.csv
```

```
import pandas as pd
import numpy as np
cases = pd.read_csv('States Reporting Cases of COVID-19 to CDC_.csv')
pd.DataFrame.from_records(cases)
cases.head()
```

	Jurisdiction	Range	Cases Reported	Community Transmission	URL
0	Alabama	1001 to 5000	4572	Yes, defined area(s)	http://www.adph.org/
1	Alaska	101 to 1000	309	Yes, defined area(s)	http://dhss.alaska.gov/Pages/default.aspx
2	American Samoa	None	None	NaN	http://dhss.as/
3	Arizona	1001 to 5000	4507	Yes, widespread	http://www.azdhs.gov/
4	Arkansas	1001 to 5000	1702	Yes, widespread	https://www.healthy.arkansas.gov/

```
#merge industry 2019 and 2020 data
geo = pd.merge(YTD, cases, how='left', left_on=['state2'],right_on=['Jurisdiction'])
geo.head()
```

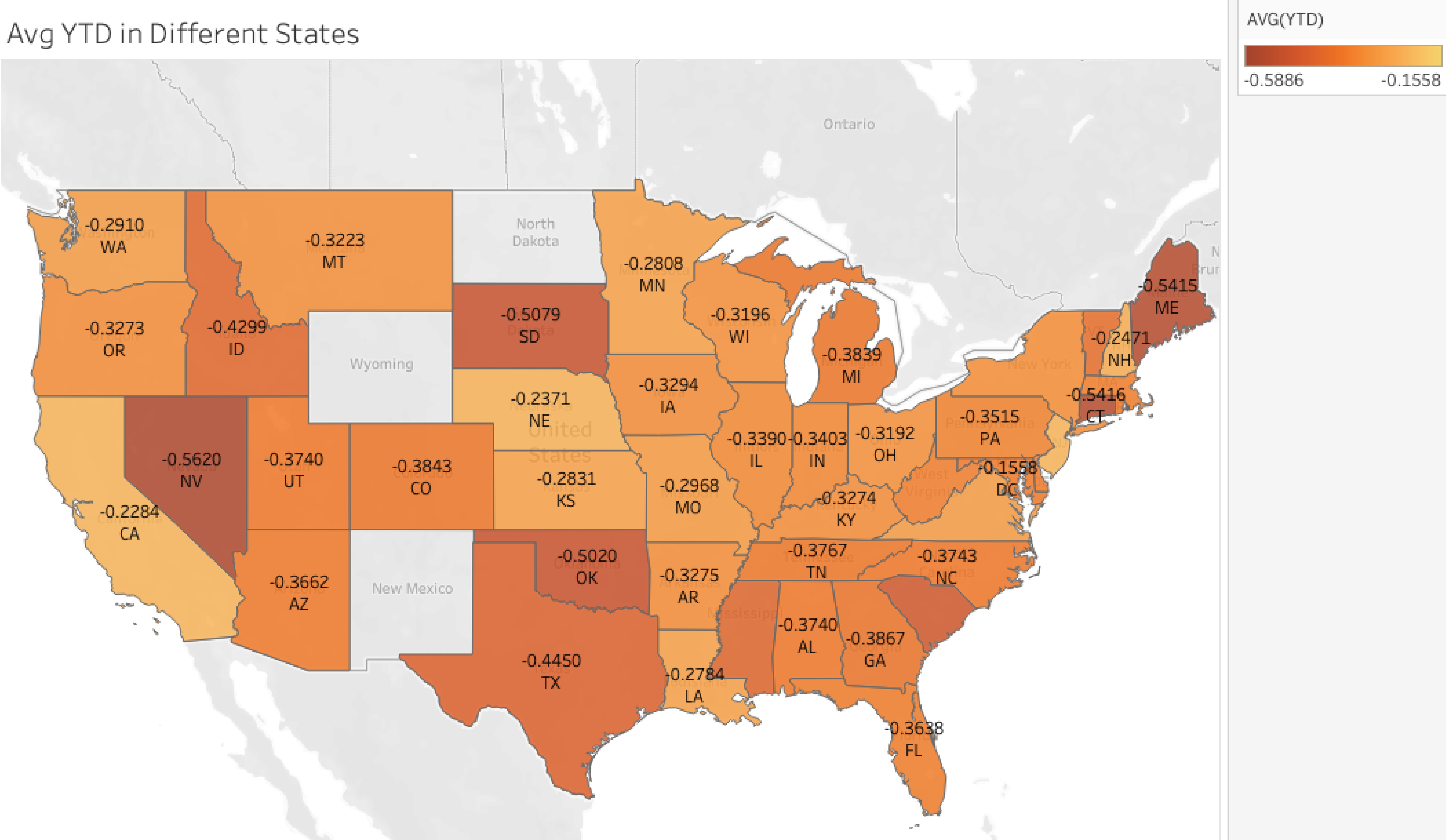
	Ticker	YTD	SIC	Group	Zip	state	state2	Jurisdiction	Range	Cases Reported	Community Transmission	UR
0	HES	-0.4983	13	1	10036	NY	New York	New York	10001 or more	224438	Yes, widespread	https://www.health.ny.go
1	APA	-0.7898	13	1	77056	TX	Texas	Texas	10001 or more	17371	Undetermined	https://www.dshs.state.tx.u
2	HAL	-0.6890	13	1	77032	TX	Texas	Texas	10001 or more	17371	Undetermined	https://www.dshs.state.tx.u
3	HP	-0.6597	13	1	74119	OK	Oklahoma	Oklahoma	1001 to 5000	2465	Yes, widespread	https://www.ok.gov/healtl

```
#only leave those columns we will use for analysis
geo=geo[['Ticker','YTD','Group','state2','Range','Cases Reported']]
#rename columns
geo=geo.rename(columns={"state2": "State", "Cases Reported":"Cases"})
geo.head()
```

	Ticker	YTD	Group	State	Range	Cases
0	HES	-0.4983	1	New York	10001 or more	224438
1	APA	-0.7898	1	Texas	10001 or more	17371
2	HAL	-0.6890	1	Texas	10001 or more	17371
3	HP	-0.6597	1	Oklahoma	1001 to 5000	2465
4	MRO	-0.7312	1	Texas	10001 or more	17371

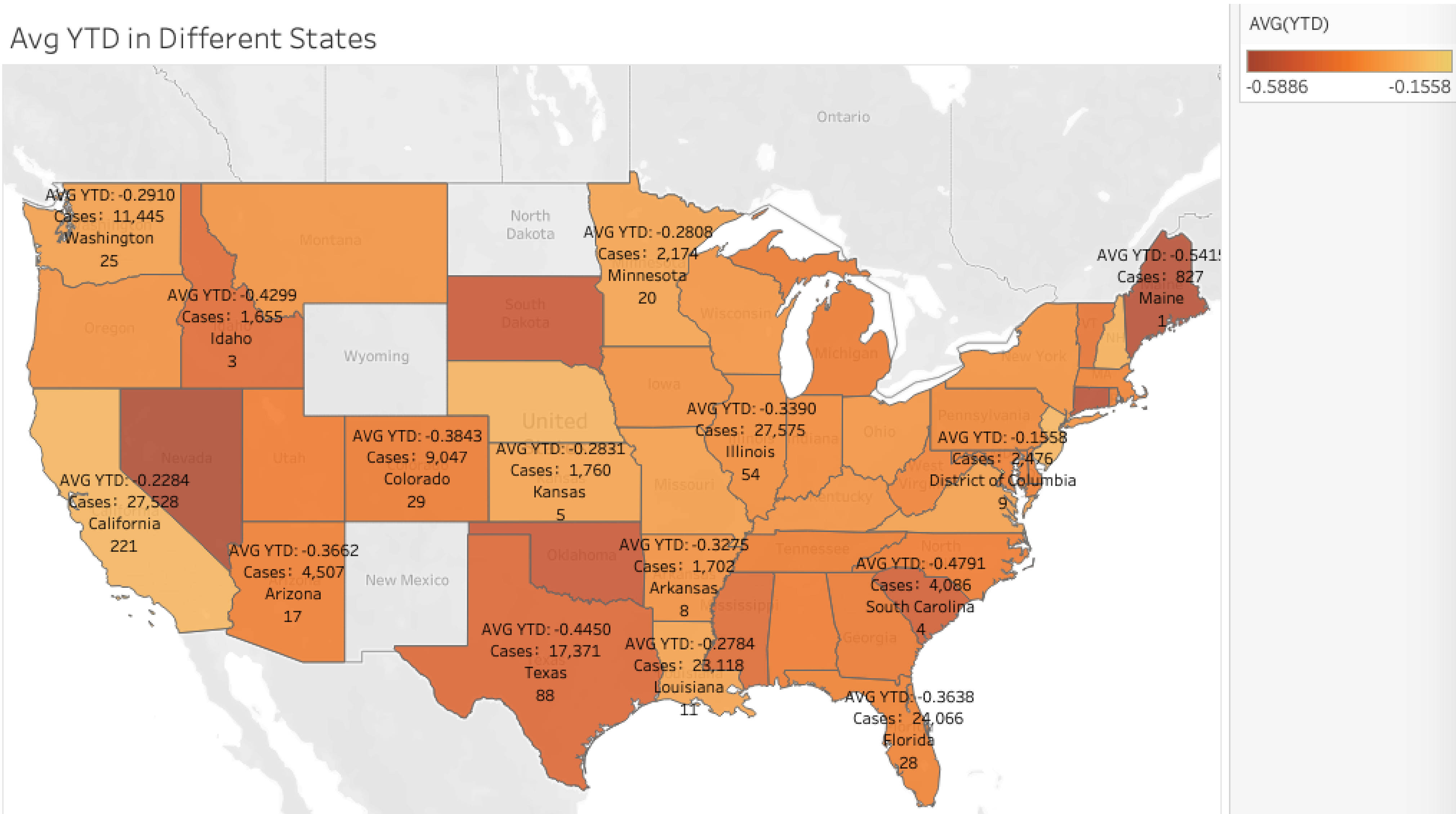
```
#geo.to_csv("geo.csv",index=False,header=True)
#could uncommened above to save the data
```

Average YTD in Different States



From this chart above, we can tell that Nevada is the state that has been affect the most, and the COVID-19 has large negative affect also for states such as CT, ME, SD, OK, ID. Califonia is the state was been affect least, we think maybe there are more software companies and IT firms which also work well when everything transfer to work online.

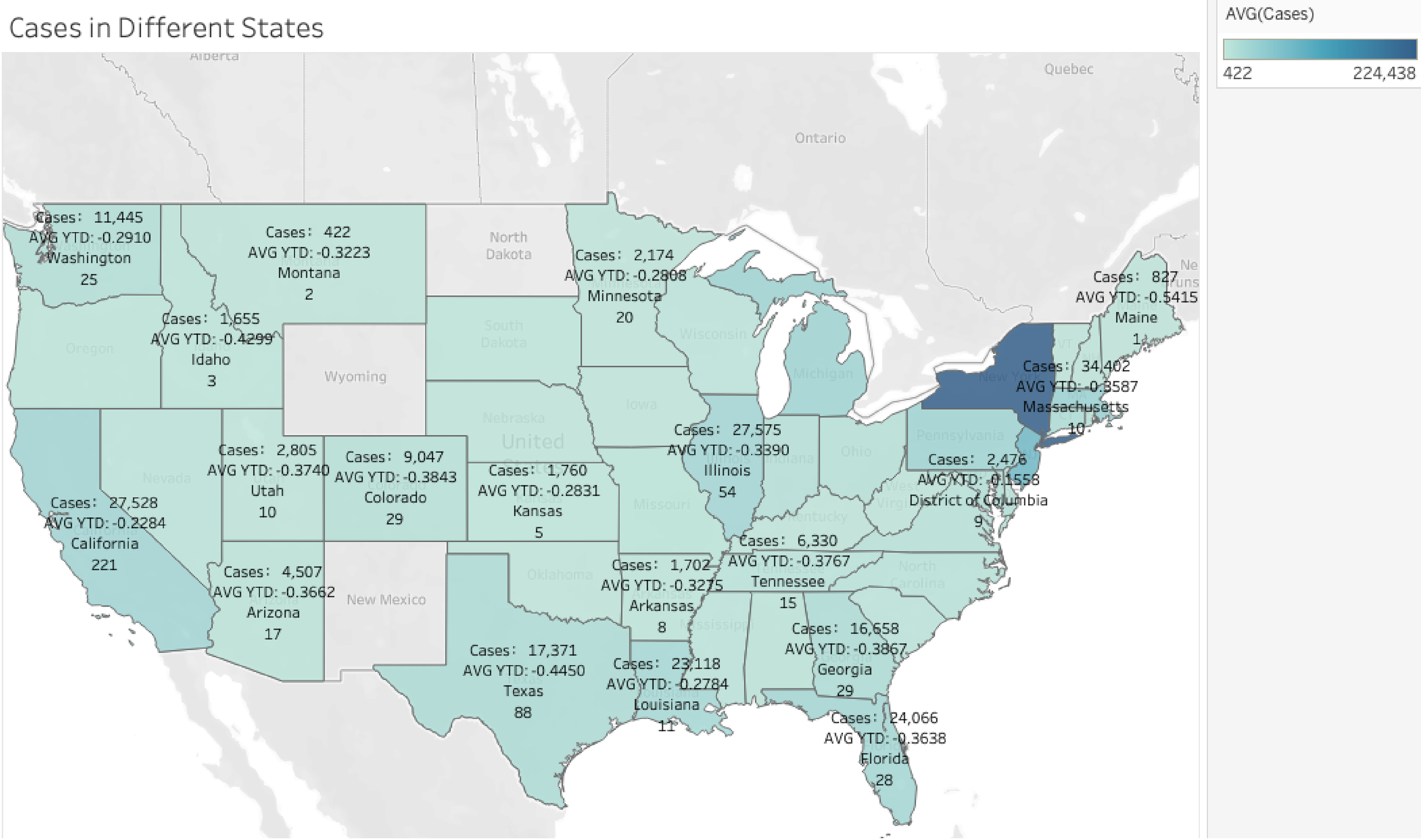
Average YTD in Different States with Cases and Ticker Count in that State



More deeply to see, since every state may have different companies, this may have a large effect on the sum of YTD, so we use the average YTD rather than the sum of YTD. We can see even California is the state that has the smallest effect by COV-19, it has the highest companies(tickers).

Cases in Different States with Avg YTD and Ticker Count in that State

Cases in Different States



From this chart above, we could see most COVID-19 cases are appearing in New York, Massachusetts, Illinois, and California, that may because some large cities are in those states, and there is a larger population. However, based on the previous chart, those states are not where has the largest negative COVID-19 effect on the stock market. Thus, we don't think there is a positive correlation between COVID-19 cases and stock returns.

OLS Regression

We want to run a regression to check our findings.

```
geo.head()
```

	Ticker	YTD	Group	State	Range	Cases
0	HES	-0.4983	1	New York	10001 or more	224438
1	APA	-0.7898	1	Texas	10001 or more	17371
2	HAL	-0.6890	1	Texas	10001 or more	17371
3	HP	-0.6597	1	Oklahoma	1001 to 5000	2465
4	MRO	-0.7312	1	Texas	10001 or more	17371

```
import statsmodels.api as sm
geo['Cases'] = pd.to_numeric(geo.Cases,errors='coerce')
X = geo['Cases']
y = geo['YTD']
X = sm.add_constant(X) # adding a constant
model = sm.OLS(y,X).fit()
model.summary()
```

OLS Regression Results

Dep. Variable:	YTD	**R-squared:**	0.001
Model:	OLS	**Adj. R-squared:**	-0.000
Method:	Least Squares	**F-statistic:**	0.8427
Date:	Mon, 20 Apr 2020	**Prob (F-statistic):**	0.359
Time:	00:54:26	**Log-Likelihood:**	193.49
No. Observations:	999	**AIC:**	-383.0
Df Residuals:	997	**BIC:**	-373.2
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-0.3320	0.007	-45.038	0.000	-0.346	-0.318
Cases	9.073e-08	9.88e-08	0.918	0.359	-1.03e-07	2.85e-07

Omnibus:	20.593	**Durbin-Watson:**	1.808
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	13.237
Skew:	0.138	**Prob(JB):**	0.00134
Kurtosis:	2.508	**Cond. No.**	8.71e+04

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.71e+04. This might indicate that there are
strong multicollinearity or other numerical problems.

Conclusion

Thus, we can conclude that Geographic Location does not have to affect the results of COV-19 shock. If we look at the regression on `YTD` & `Cases`, it seems that the number of cases doesn't affect the stock returns since the coefficient for Cases is close to 0, and the p-value is 0.359 which is larger than 0.05, and also it's R-square is only 0.001. Those firms in larger cities may have less negative stock returns in 2020 after COVID-19 shock because of their special industry. Thus, the results of COVID-19 shock is not determined by only the case numbers in those different locations.