Data Bootcamp Final Project

Presented By: Jok Aleu and Zach Xue

Background:

The year of 2020 was extremely tumultuous. Not just in the social and political aspect
but also economically as well. Because of the coronavirus, many corporations within t
States and all around the world experienced financial significant setbacks and change
This led us to wonder what the exactly caused the stock market to increase or decrea
that was affected by the coronavirus. We wanted to see what kind of impact US GDP
the stock market, and what exactly happened to the stock market in 2020 as a result
coronavirus.

Project Description:

Overall, we want to find the fundamental factors that cause stock market growth. We
comparing GDP data, company dividends, and estimated earnings data to stock mark
is clustered by industry sectors. The largest discernable differences between the pote
and stock market data show us the fundamental patterns/factors for stock market gro

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import statsmodels.formula.api as smf
%matplotlib inline
```

1. Reading in and Cleaning the Datasets

In [133]:

```
stock_prices = pd.read_csv('C:/Users/zachx/Documents/Data_Bootcam
stock_prices
#This dataset contains a list of all stock prices in the past 20
#It contains values such as the opening price, closing price, sto
```

Out[133]:

	symbol	date	open	high	low	close	close_adjusted	vol
0	MSFT	5/16/2016	50.80	51.96	50.75	51.83	49.7013	2003
1	MSFT	1/16/2002	68.85	69.84	67.85	67.87	22.5902	3097
2	MSFT	9/18/2001	53.41	55.00	53.17	54.32	18.0802	4159
3	MSFT	10/26/2007	36.01	36.03	34.56	35.03	27.2232	28812
4	MSFT	6/27/2014	41.61	42.29	41.51	42.25	38.6773	7464
...	...	...	...	...	...	...	...	
1048570	BOKF	10/4/2010	45.09	45.41	44.83	45.00	36.9596	9
1048571	BOKF	1/19/2007	52.51	52.66	52.33	52.63	40.2866	3
1048572	BOKF	10/6/2011	48.11	48.81	47.16	48.74	40.8965	13
1048573	BOKF	4/13/2011	52.19	52.39	51.13	51.13	42.4157	3
1048574	BOKF	1/27/2000	17.72	17.72	16.63	16.63	11.0822	3

1048575 rows × 9 columns

In [134]:
```python
sp500 = pd.read_csv('sp500.csv')
sp500
#This dataset, taken from Yahoo Finance, indicates the daily open
```

Out[134]:

	Date	Open	High	Low	Close	Adj Close
0	2015-12-11	2047.270020	2047.270020	2008.800049	2012.369995	2012.369995
1	2015-12-14	2013.369995	2022.920044	1993.260010	2021.939941	2021.939941
2	2015-12-15	2025.550049	2053.870117	2025.550049	2043.410034	2043.410034
3	2015-12-16	2046.500000	2076.719971	2042.430054	2073.070068	2073.070068
4	2015-12-17	2073.760010	2076.370117	2041.660034	2041.890015	2041.890015
...	...	...	...	...	...	...
1254	2020-12-04	3670.939941	3699.199951	3670.939941	3699.120117	3699.120117
1255	2020-12-07	3694.729980	3697.409912	3678.879883	3691.959961	3691.959961
1256	2020-12-08	3683.050049	3708.449951	3678.830078	3702.250000	3702.250000
1257	2020-12-09	3705.979980	3712.389893	3660.540039	3672.820068	3672.820068
1258	2020-12-10	3659.129883	3678.489990	3645.179932	3668.100098	3668.100098

1259 rows × 7 columns

In [135]:

```python
nasdaq = pd.read_csv('nasdaq.csv')
nasdaq
#This dataset, taken from Yahoo Finance, indicates the daily open
```

Out[135]:

	Date	Open	High	Low	Close	Adj Cl
0	2015-12-11	4979.770020	4996.189941	4928.669922	4933.470215	4933.470
1	2015-12-14	4932.609863	4953.600098	4871.589844	4952.229980	4952.229
2	2015-12-15	4991.209961	5026.540039	4986.990234	4995.359863	4995.359
3	2015-12-16	5033.479980	5078.990234	4992.629883	5071.129883	5071.129
4	2015-12-17	5087.169922	5088.580078	5002.549805	5002.549805	5002.549
...	...	...	...	...	...	
1254	2020-12-04	12399.320313	12464.230469	12376.440430	12464.230469	12464.230
1255	2020-12-07	12461.000000	12536.230469	12460.549805	12519.950195	12519.950
1256	2020-12-08	12503.169922	12594.540039	12453.209961	12582.769531	12582.769
1257	2020-12-09	12591.690430	12607.139648	12290.780273	12338.950195	12338.950
1258	2020-12-10	12247.549805	12431.559570	12214.740234	12405.809570	12405.809

1259 rows × 7 columns

In [136]:

```python
dowjones = pd.read_csv('dowjones.csv')
dowjones
#This dataset, taken from Yahoo Finance, indicates the daily open
```

Out[136]:

	Date	Open	High	Low	Close	Adj Cl
0	2015-12-11	17574.750000	17574.750000	17230.500000	17265.210938	17265.210
1	2015-12-14	17277.109375	17378.019531	17138.470703	17368.500000	17368.500
2	2015-12-15	17374.779297	17627.630859	17341.179688	17524.910156	17524.910
3	2015-12-16	17530.849609	17784.359375	17483.679688	17749.089844	17749.089
4	2015-12-17	17756.539063	17796.759766	17493.500000	17495.839844	17495.839
...	...	...	...	...	...	
1254	2020-12-04	29989.560547	30218.259766	29989.560547	30218.259766	30218.259
1255	2020-12-07	30233.029297	30233.029297	29967.220703	30069.789063	30069.789
1256	2020-12-08	29997.949219	30246.220703	29972.070313	30173.880859	30173.880
1257	2020-12-09	30229.810547	30319.699219	29951.849609	30068.810547	30068.810
1258	2020-12-10	30032.550781	30063.869141	29876.820313	29999.259766	29999.259

1259 rows × 7 columns

(Zach)

I created a dataset that calculated the percent change between the quarters for 5 yea Dow Jones Industrial Average. For each quarter, as shown below, there is a correspo column observation that indicates the percent change in closing prices between the p quarter and this quarter.

In [132]:

```python
dowjones_returns = pd.read_excel('dowjones_returns.xlsx')
dowjones_returns.columns = dowjones_returns.columns.str.lower()
dowjones_returns
```

Out[132]:

	quarter	return
0	2016 Q1	0.0794
1	2016 Q2	0.0371
2	2016 Q3	-0.0157
3	2016 Q4	0.0949
4	2017 Q1	0.0542
5	2017 Q2	0.0454
6	2017 Q3	0.0679
7	2017 Q4	0.1186
8	2018 Q1	-0.0760
9	2018 Q2	0.0518
10	2018 Q3	-0.0118
11	2018 Q4	-0.0046
12	2019 Q1	0.0637
13	2019 Q2	0.0102
14	2019 Q3	0.0068
15	2019 Q4	0.0447
16	2020 Q1	-0.1384
17	2020 Q2	0.0855
18	2020 Q3	0.0028

In [137]:

```python
earnings = pd.read_csv('earnings_latest.csv')
earnings = earnings.dropna()
earnings
#This dataset contains the earnings data for all stocks in the pa
```

Out[137]:

	symbol	date	qtr	eps_est	eps	release_time
14	A	2012-11-19	10/2012	0.800	0.84	post
15	A	2013-02-14	01/2013	0.660	0.63	post
16	A	2013-05-14	04/2013	0.670	0.77	post
17	A	2013-08-14	07/2013	0.620	0.68	post
18	A	2013-11-14	10/2013	0.760	0.81	post
...	...	...	...	...	...	...
160655	ZYXI	2019-10-29	Q3	0.057	0.06	post
160656	ZYXI	2020-02-27	Q4	0.077	0.09	post
160657	ZYXI	2020-04-28	Q1	0.063	0.09	post
160658	ZYXI	2020-07-28	Q2	0.086	0.09	post
160659	ZYXI	2020-10-27	Q3	0.053	0.04	post

77282 rows × 6 columns

In [138]:

```python
dividends = pd.read_csv('dividends_latest.csv')
dividends = dividends.dropna()
dividends
#This dataset contains the dividends data for all stocks in the p
```

Out[138]:

	symbol	date	dividend
0	MSFT	2016-11-15	0.3900
1	MSFT	2011-05-17	0.1600
2	MSFT	2008-05-13	0.1100
3	MSFT	2011-02-15	0.1600
4	MSFT	2012-02-14	0.2000
...	...	...	...
250144	EBTC	2020-11-09	0.1750
250145	EFAS	2020-11-04	0.0525
250146	EFBI	2020-11-12	0.0500
250147	CDL	2020-11-12	0.1111
250148	FANG	2020-11-10	0.3750

250149 rows × 3 columns

(Zach)

- I created a dataset that calculated the percent change in US GDP, based on the c gross_output dataset. Here, for the periods of 2018-2020, I found the percent cha GDP for each individual industry sector.
- To clean this dataset, I manually added in the column labels for the quarters. I als the dataset so that the quarter values would be able to merge successfully onto c values. I dropped one of the obsolete 'Quarter' columns from the dataset, and res so that the quarter column wasn't an index variable.

In [131]:

```
gross_output_returns = pd.read_excel('gross_output_returns.xlsx')
gross_output_returns.columns = ['Quarter', '2018 Q2', '2018 Q3',
gross_output_returns  = gross_output_returns.T
gross_output_returns.columns = gross_output_returns.iloc[0]
gross_output_returns = gross_output_returns.drop('Quarter')
gross_output_returns = gross_output_returns.reset_index()
gross_output_returns = gross_output_returns.rename(columns={'inde
gross_output_returns
```

Out[131]:

Quarter	quarter	All industries	Private industries	Agriculture, forestry, fishing, and hunting	Farms	Forestry, fishing, and related activities
0	2018 Q2	0.0171435	0.0176794	0.00831328	0.00972326	-0.00178571
1	2018 Q3	0.0127205	0.0128463	-0.0238663	-0.022716	-0.0322004
2	2018 Q4	0.0078176	0.00773506	0.0108913	0.0128853	-0.00554529
3	2019 Q1	0.00463912	0.00463614	-0.0171504	-0.0214517	0.0148699
4	2019 Q2	0.00886085	0.00847986	0.00313199	0.00458833	-0.00732601
5	2019 Q3	0.0074783	0.00757008	0.0102587	0.0126871	-0.00553506
6	2019 Q4	0.0052328	0.00488388	0.0183223	0.0170383	0.0278293
7	2020 Q1	-0.00950775	-0.0107804	0.0138738	0.0177384	-0.0144404
8	2020 Q2	-0.0942929	-0.103021	-0.109258	-0.109417	-0.108059

To clean the gross_output dataset, I skipped the first 5 rows and the last 6 rows, while
index to be the industries column. Then, I manually added in column labels.

In [139]:

```
gross_output = pd.read_csv('gross_output.csv', skiprows=5, skipfo
gross_output = gross_output.dropna()
gross_output = gross_output.drop('Line', axis=1)
gross_output.columns = ['2018 Q1', '2018 Q2', '2018 Q3', '2018 Q4
gross_output
```

```
<ipython-input-139-0c16db8932ff>:1: ParserWarning: Falling back t
hon' engine because the 'c' engine does not support skipfooter; y
id this warning by specifying engine='python'.
  gross_output = pd.read_csv('gross_output.csv', skiprows=5, skip
index_col=1)
```

Out[139]:

	2018 Q1	2018 Q2	2018 Q3	2018 Q4	2019 Q1	2019 Q2	2019 Q3	
All industries	35838.6	36453.0	36916.7	37205.3	37377.9	37709.1	37991.1	38
Private industries	31958.1	32523.1	32940.9	33195.7	33349.6	33632.4	33887.0	34(
Agriculture, forestry, fishing, and hunting	457.1	460.9	449.9	454.8	447.0	448.4	453.0	4
Farms	401.1	405.0	395.8	400.9	392.3	394.1	399.1	4
Forestry, fishing, and related activities	56.0	55.9	54.1	53.8	54.6	54.2	53.9	
...	...	...	...	...	...	...	...	
State and local	2714.9	2746.1	2777.9	2801.7	2807.4	2834.8	2852.6	28
General government	2360.1	2388.0	2417.1	2437.2	2440.6	2463.4	2479.0	24
Government enterprises	354.8	358.2	360.9	364.5	366.8	371.4	373.5	3
Private goods-producing industries1	8721.7	8892.3	9029.4	9031.1	8989.1	9010.3	8966.9	89
Private services-producing industries2	23236.4	23630.8	23911.5	24164.6	24360.5	24622.1	24920.1	25(

99 rows × 10 columns

In [140]:

```
gross_output['2019'] = gross_output['2019 Q1'] + gross_output['20
gross_output['2020'] = gross_output['2020 Q1'] + gross_output['20
#Here, I simplified the quarterly data by adding up all the quart
```

In [142]:
```
gross_output_2019 = gross_output.groupby(gross_output.index)['201
gross_output_2020 = gross_output.groupby(gross_output.index)['202
#I created two new dataframes that take in the index of the gross
#Then, I grouped the index by the new yearly columns that I just
#I only want to look at the Top 30 industries in the US GDP datas
```

2. Time Series Graphs of Stock Market Data (By Popular Mark Indices)

- From the S&P 500, NASDAQ, and Dow Jones datasets, I created dataframes dif year (2019 vs 2020) that compared the indices closing prices for each day.
- Following this, using matplotlib, I created line plots with each line representing a index for 2019 and 2020 data

In [14]:
```
sp500_2019 = sp500.loc[sp500['Date'].str.contains('2019')].groupb
sp500_2019 = sp500_2019.set_index('Date')

sp500_2020 = sp500.loc[sp500['Date'].str.contains('2020')].groupb
sp500_2020 = sp500_2020.set_index('Date')
```

In [15]:
```
nasdaq_2019 = nasdaq.loc[nasdaq['Date'].str.contains('2019')].gro
nasdaq_2019 = nasdaq_2019.set_index('Date')

nasdaq_2020 = nasdaq.loc[nasdaq['Date'].str.contains('2020')].gro
nasdaq_2020 = nasdaq_2020.set_index('Date')
```

In [16]:
```
dowjones_2019 = dowjones.loc[dowjones['Date'].str.contains('2019'
dowjones_2019 = dowjones_2019.set_index('Date')

dowjones_2020 = dowjones.loc[dowjones['Date'].str.contains('2020'
dowjones_2020 = dowjones_2020.set_index('Date')
```

In [148]:
```
plt.style.use('fivethirtyeight')
```

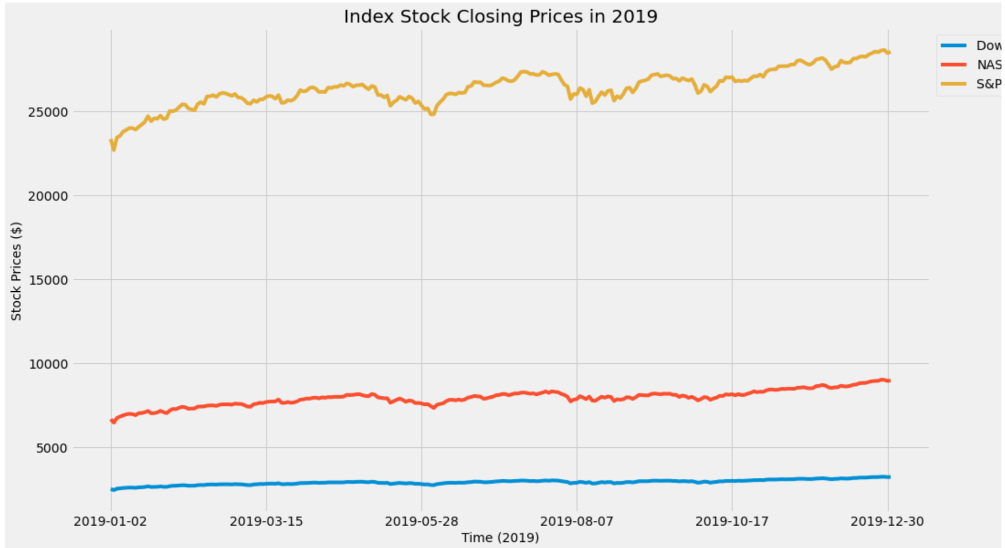
In [150]:

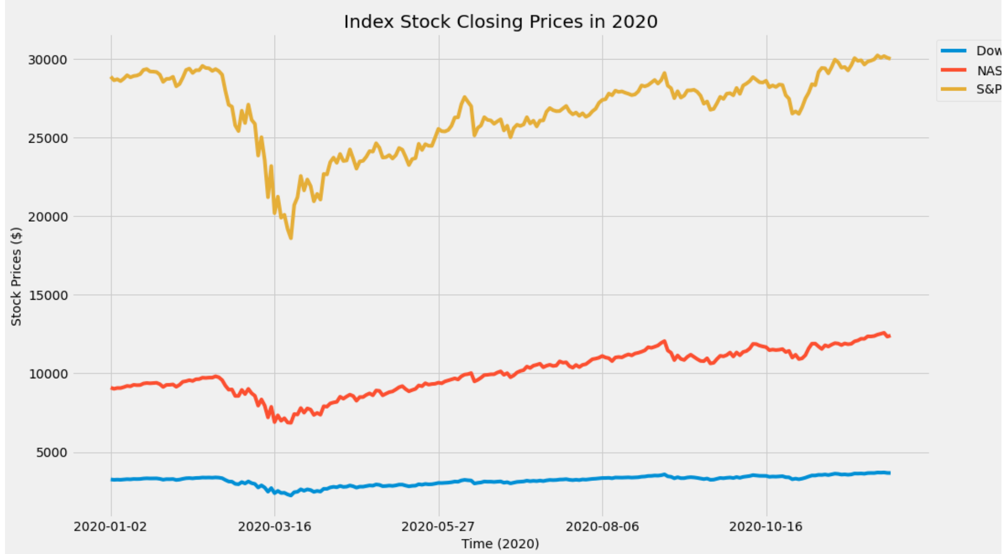
```
fig, ax = plt.subplots(nrows=2, ncols=1)

sp500_2019.plot(ax=ax[0], figsize=(15,20))
nasdaq_2019.plot(ax=ax[0])
dowjones_2019.plot(ax=ax[0])
#Plots all three lines on the same plot

sp500_2020.plot(ax=ax[1])
nasdaq_2020.plot(ax=ax[1])
dowjones_2020.plot(ax=ax[1])
#Plots all three lines on the next plot

ax[0].set_title('Index Stock Closing Prices in 2019', fontsize=20
ax[0].set_ylabel('Stock Prices ($)', fontsize=14)
ax[0].set_xlabel('Time (2019)', fontsize=14)

ax[1].set_title('Index Stock Closing Prices in 2020', fontsize=20
ax[1].set_ylabel('Stock Prices ($)', fontsize=14)
ax[1].set_xlabel('Time (2020)', fontsize=14)

ax[0].legend({'S&P 500', 'NASDAQ', 'Dow Jones Industrial Average'
ax[1].legend({'S&P 500', 'NASDAQ', 'Dow Jones Industrial Average'
```

```
C:\Users\zachx\anaconda3\lib\site-packages\pandas\plotting\_matpl
e.py:1235: UserWarning: FixedFormatter should only be used togeth
xedLocator
  ax.set_xticklabels(xticklabels)
C:\Users\zachx\anaconda3\lib\site-packages\pandas\plotting\_matpl
e.py:1235: UserWarning: FixedFormatter should only be used togeth
xedLocator
  ax.set_xticklabels(xticklabels)
```

Out[150]:

```
<matplotlib.legend.Legend at 0x2c445630cd0>
```





Explanation and Analysis:

- In 2019, all of the popular indices looks to be relatively stable throughout most of S&P 500 looks to fluctuate more than either the NASDAQ or the Dow. But, overa performance seems to be consistently rising throughout 2019. The S&P's fluctuat be because many of the Top 500 companies listed on that index are technology c These companies tend to have more price fluctation than others because of cons innovations within the industry. New developments and product announcements c drastically affect the price of the stock on a daily, weekly, or monthly basis. Lookir NASDAQ and the Dow, the NASDAQ tends to vary more than the Dow. This migl because it simply contains more companies in its index when compared to the Do companies.

- In 2020, there was a significant drop in stock prices for all three indices during the and the beginning of Q2 2020. This likely was because of the introduction of coro the worldwide pandemic as a whole. In addition, the pandemic also ushered in th mandatory quarantines and stay-at-home orders, which looks to have drastically stock prices. Particually, the S&P 500 looks to have fallen the most during this pe this could be due to the large amount of technology companies present with the S

impact of coronavirus cannot be understated. Many companies in the S&P that s
outdoors entertainment, like carnivals or cruises, lost significant amount of marke
during the beginning of Q2 2020. Interestingly enough, the Dow and NASDAQ to
significantly smaller hit than the S&P. This might be because of the Federal reser
influx of monetary stimulus, allowing the stock market to rebound after the initial e
coronavirus. However, the fluctuations in the data still remain to this day, starkly c
the relative stability present in the 2019 data.

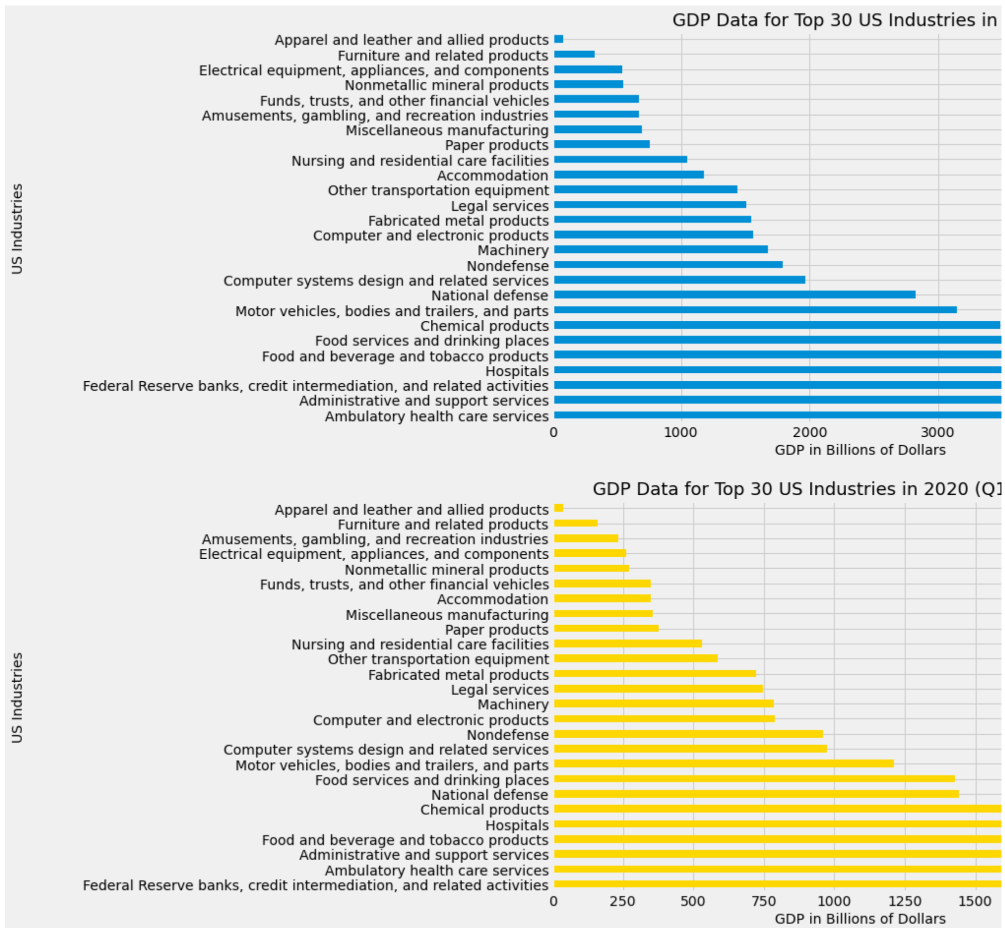
- This data shows us that coronavirus had a fundamental impact on the stock mark when compared 2019 vs. 2020 data. If stock market data fluctuated and changec between these specific time periods, what does that mean for GDP data? Are spe industries now far different than they were in the previous years?

3. Comparison of GDP Data By Sector (2019 vs. 2020)

- For the gross_output datasets, I removed the rows: Private industries, all industri they do not provide adequate or relevant information about the specific industries to investigate.

In [151]:
```
fig, ax = plt.subplots(nrows=2, ncols=1)

gross_output_2019.sort_values(ascending=False).iloc[4:].plot.barh
gross_output_2020.sort_values(ascending=False).iloc[4:].plot.barh

ax[0].set_title('GDP Data for Top 30 US Industries in 2019', font
ax[0].set_xlabel('GDP in Billions of Dollars', fontsize=14)
ax[0].set_ylabel('US Industries', fontsize=14)

ax[1].set_title('GDP Data for Top 30 US Industries in 2020 (Q1 an
ax[1].set_xlabel('GDP in Billions of Dollars', fontsize=14)
ax[1].set_ylabel('US Industries', fontsize=14)
```

Out[151]: Text(0, 0.5, 'US Industries')



Explanation and Analysis

- Because the 2020 GDP data only includes values for Q1 and Q2 2020, it is rougł value of 2019 GDP data

- The 2019 GDP data lists ambulatory health services as the highest gross output dollars for all quarters of 2019. This is closely followed by administrative and sup and Federal Reserve banks, and hospitals. Because of the United States health (none of this is particularly surprising. However, the 2020 data suggests a differen The Federal Reserve banks, credit intermediation and related activities actually h highest gross output so far This is likely because of the Federal Reserve's stimulı

that helped the stock market and the overall economy (in the form of GDP) rebou
initial outbreak of coronavirus. In addition, sectors such as motor vehicles are rel
in 2020. This might be because of the nationwide lockdowns and quarantines tha
people from moving about too much, especially by motor vehicles, for the time be

- By looking at this comparison of GDP data, we can see that specific industries/se federal reserve and motor vehicles, differ in their GDP values. Compared to the e downturn of the stock market, particularly for the federal reserve sector, it's unde how the two metrics (GDP data and stock market data) would be correlated, or a an observable effect on each other.

4. Regression Analysis of Potential Stock Market Growth Fact against Stock Market Growth

- For this section, we decided to examine the impact of the estimated earnings per the dividends on the stock closing price for 2020.

In [143]:
```
from sklearn.linear_model import LinearRegression as linreg
#Importing the linear regression package from scikit learn
```

In [144]:
```
earnings_stocks = pd.merge(earnings, stock_prices, on='symbol', h
#Merging the two datasets: earnings data and stock prices data in
```

In [145]:

```
earnings_stocks_2020 = earnings_stocks[earnings_stocks['date_x'].
del earnings_stocks_2020['date_y']
earnings_stocks_2020 = earnings_stocks_2020.rename({'date_x': 'da
earnings_stocks_2020

#Forming a new dataframe that only takes in values for 2020 stock
#Dropping the redundant date_y column, as there already exists a
```

Out[145]:

	symbol	date	qtr	eps_est	eps	release_time	open	high	lc
159132	AAOI	2020-02-27	Q4	-0.234	-0.18	post	25.85	25.85	24.85
159133	AAOI	2020-02-27	Q4	-0.234	-0.18	post	18.41	18.46	17.43
159134	AAOI	2020-02-27	Q4	-0.234	-0.18	post	22.10	22.42	21.69
159135	AAOI	2020-02-27	Q4	-0.234	-0.18	post	24.49	24.96	24.23
159136	AAOI	2020-02-27	Q4	-0.234	-0.18	post	12.93	13.13	12.56
...	...	...	...	...	...	...	...	...	
15187200	BMRA	2020-10-15	Q1	-0.080	-0.14	post	0.70	0.70	0.70
15187201	BMRA	2020-10-15	Q1	-0.080	-0.14	post	3.75	3.75	3.28
15187202	BMRA	2020-10-15	Q1	-0.080	-0.14	post	0.43	0.43	0.43
15187203	BMRA	2020-10-15	Q1	-0.080	-0.14	post	0.45	0.45	0.45
15187204	BMRA	2020-10-15	Q1	-0.080	-0.14	post	1.10	1.10	1.10

720507 rows × 13 columns

4a. Running the Regression

In [153]:

```
reg = linreg().fit(earnings_stocks_2020[['eps_est']], earnings_st
#Creating the regression variable to fit the estimated earnings p
```

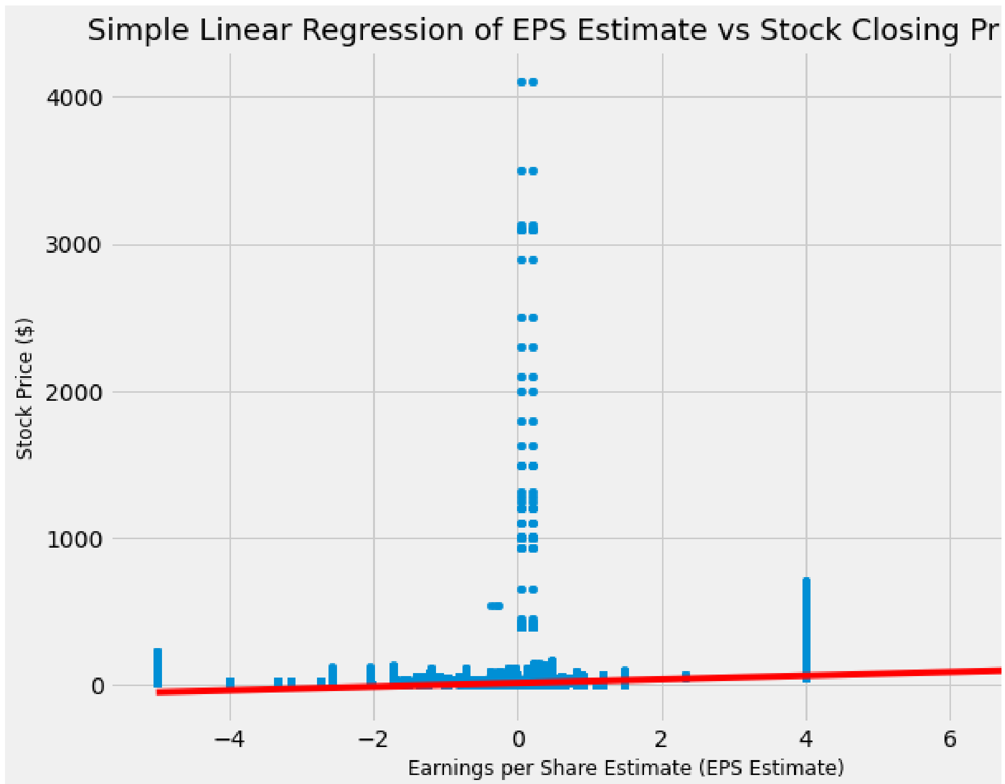
In [154]:

```
earnings_stocks_2020['yhat'] = reg.predict(earnings_stocks_2020[[
#Creating the yhat regression line that will be plotted along wit
```

In [156]:

```
fig, ax = plt.subplots()

earnings_stocks_2020.plot.scatter(x='eps_est', y='close', ax=ax)
earnings_stocks_2020.sort_values('eps_est').set_index('eps_est')[
#Creating the scatterplot for the data, with the index set to est

ax.set_title('Simple Linear Regression of EPS Estimate vs Stock C
ax.set_ylabel('Stock Price ($)', fontsize=12)
ax.set_xlabel('Earnings per Share Estimate (EPS Estimate)', fonts
```

Out[156]: Text(0.5, 0, 'Earnings per Share Estimate (EPS Estimate)')



In [370]:

```
earnings_stocks_2020[['eps_est', 'close']].corr()
#Correlation coefficient between estimated earnings per share and
```

Out[370]:

	eps_est	close
eps_est	1.000000	0.395993
close	0.395993	1.000000

In [146]:

```
reg1 = smf.ols('eps_est ~ close', data=earnings_stocks_2020).fit(
print(reg1.summary())
#OLS Regression Results with summary statistics
```

```
                            OLS Regression Results
=================================================================
===
Dep. Variable:                eps_est   R-squared:
157
Model:                            OLS   Adj. R-squared:
157
Method:                 Least Squares   F-statistic:
+05
Date:                Sat, 12 Dec 2020   Prob (F-statistic):
0.00
Time:                        15:03:41   Log-Likelihood:
+06
No. Observations:              720507   AIC:
+06
Df Residuals:                  720505   BIC:
+06
Df Model:                           1
Covariance Type:            nonrobust
=================================================================
===
                 coef    std err          t      P>|t|      [0.02
75]
-----------------------------------------------------------------
---
Intercept     -0.0815      0.002    -49.632      0.000      -0.08
078
close          0.0126   3.44e-05    366.053      0.000       0.01
013
=================================================================
===
Omnibus:                   309446.607   Durbin-Watson:
254
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
378
Skew:                           0.698   Prob(JB):
0.00
Kurtosis:                      75.435   Cond. No.
0.5
=================================================================
===

Warnings:
[1] Standard Errors assume that the covariance matrix of the erro
ectly specified.
```

Explanation and Analysis

- Looking at the data on the scatter plot, there seems to be quite the number of out an x value near zero earnings per share. Earnings per share is supposed to be re of a company's profitability, so it is a little bit unusual to see such high stock closi comparison to an almost zero earnings per share value. Thus, we can conclude t values are outliers towards the regression analysis, and we cannot factor that int conclusions

- The general understanding of the regression analysis seems to be that when ear share increases, so does the company's stock price. From the definition of earnir this is understandable. However, the R^2 value of 0.157, as stated in the statistic data above, means that the model can only explain 15.7% of the variation in y. Tl coefficient is also only 39.59%. The p value is also at 0.0, meaning that these val likely did not occur just due to chance. This allows us to conclude that earnings p not a fundamental factor for predicting stock market growth.

- This is not to say that earnings per share isn't a factor for predicting stock market entirely. Earnings per share is a widely used metric for showing how much mone makes for each share of its stock, showcasing corporate value. While companies low EPS, their stock price can be high because stock price by itself isn't the only company's value. The nature of the variables lend variability towards this analysis

In [107]:

```
dividends_stocks = pd.merge(dividends, stock_prices, on='symbol',
#Reading in a merged dataframe of the dividends dataset and the s
```

In [108]:

```python
dividends_stocks_2020 = dividends_stocks[dividends_stocks['date_x
del dividends_stocks_2020['date_y']
dividends_stocks_2020 = dividends_stocks_2020.rename({'date_x': '
dividends_stocks_2020

#Creating a new dataframe for dividends only in 2020
#Removing the excess date_y column from the new merged dataframe
```

Out[108]:

	symbol	date	dividend	open	high	low	close	close_adjusted
290816	MSFT	2020-02-19	0.51	50.80	51.96	50.75	51.83	49.7013
290817	MSFT	2020-02-19	0.51	68.85	69.84	67.85	67.87	22.5902
290818	MSFT	2020-02-19	0.51	53.41	55.00	53.17	54.32	18.0802
290819	MSFT	2020-02-19	0.51	36.01	36.03	34.56	35.03	27.2232
290820	MSFT	2020-02-19	0.51	41.61	42.29	41.51	42.25	38.6773
...	...	...	...	...	...	...	...	...
23970812	AOBC	2020-09-16	0.05	0.78	0.81	0.69	0.69	0.6900
23970813	AOBC	2020-09-16	0.05	1.75	1.78	1.70	1.78	1.7800
23970814	AOBC	2020-09-16	0.05	1.45	1.45	1.35	1.42	1.4200
23970815	AOBC	2020-09-16	0.05	3.88	3.93	3.87	3.92	3.9200
23970816	AOBC	2020-09-16	0.05	1.50	1.53	1.44	1.44	1.4400

914632 rows × 10 columns

In [157]:

```python
reg2 = linreg().fit(dividends_stocks_2020[['dividend']], dividend
#Creating the regression variable to fit the dividends with the s
```

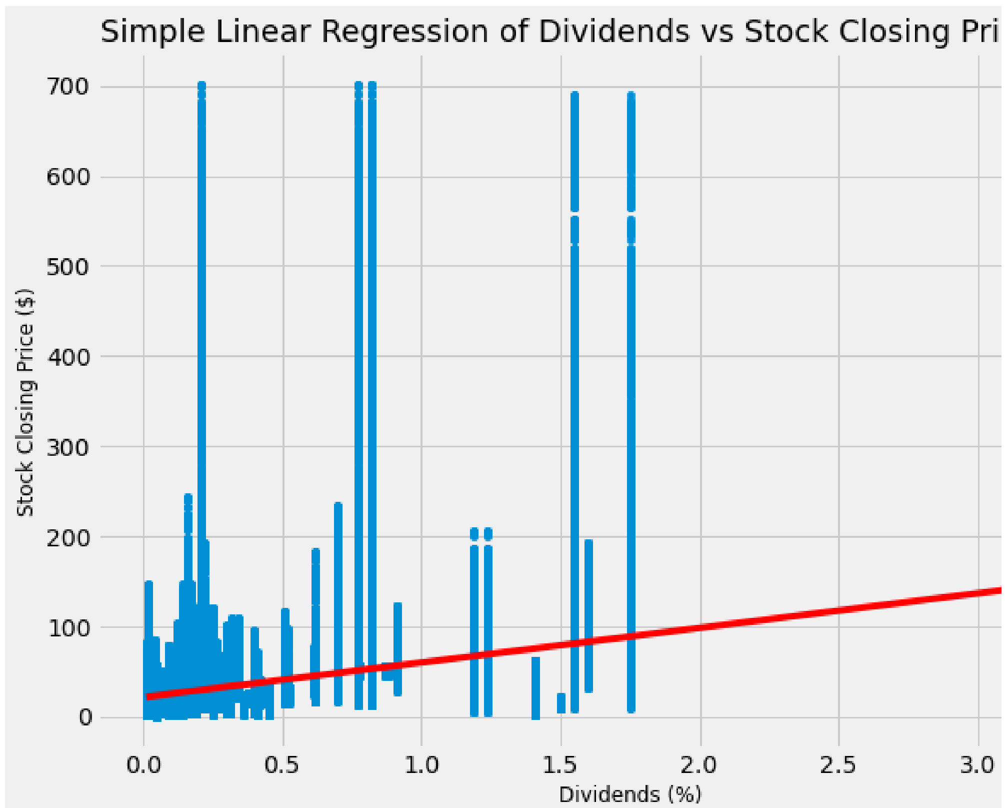
In [158]:

```python
dividends_stocks_2020['yhat'] = reg2.predict(dividends_stocks_202
#Creating the yhat regression line that will be fit to the scatte
```

In [196]:

```
fig, ax = plt.subplots()

dividends_stocks_2020.plot.scatter(x='dividend', y='close', ax=ax
dividends_stocks_2020.sort_values('dividend').set_index('dividend

ax.set_title('Simple Linear Regression of Dividends vs Stock Clos
ax.set_ylabel('Stock Closing Price ($)', fontsize=12)
ax.set_xlabel('Dividends (%)', fontsize=12)
```

Out[196]: Text(0.5, 0, 'Dividends (%)')



In [384]:

```
dividends_stocks_2020[['dividend', 'close']].corr()
#Correlation coefficient between dividends and stock closing pric
```

Out[384]:

	dividend	close
dividend	1.000000	0.380043
close	0.380043	1.000000

In [147]:

```python
reg3 = smf.ols('dividend ~ close', data=dividends_stocks_2020).fi
print(reg3.summary())
#OLS Regression summary
```

```
                            OLS Regression Results
=================================================================
===
Dep. Variable:               dividend   R-squared:
144
Model:                            OLS   Adj. R-squared:
144
Method:                 Least Squares   F-statistic:
+05
Date:                Sat, 12 Dec 2020   Prob (F-statistic):
0.00
Time:                        15:41:30   Log-Likelihood:
+05
No. Observations:              914632   AIC:
+06
Df Residuals:                  914630   BIC:
+06
Df Model:                           1
Covariance Type:            nonrobust
=================================================================
===
                 coef    std err          t      P>|t|      [0.02
75]
-----------------------------------------------------------------
---
Intercept      0.1998      0.001    363.661      0.000       0.19
201
close          0.0038   9.56e-06    392.942      0.000       0.00
004
=================================================================
===
Omnibus:                   755068.463   Durbin-Watson:
183
Prob(Omnibus):                  0.000   Jarque-Bera (JB):
718
Skew:                           3.881   Prob(JB):
0.00
Kurtosis:                      26.028   Cond. No.
1.2
=================================================================
===

Warnings:
[1] Standard Errors assume that the covariance matrix of the erro
ectly specified.
```

Explanation and Analysis

- Looking at the graph, there seems to be quite a large number of outliers in the da many stocks that have small dividend percentages and large stock prices. This c to the fact that not all companies pay dividends. Instead of divulging it to shareho choose to keep that money as retained earnings. As such, they have relatively sr dividend percentages. However, this can make the analysis a bit ambiguous.

- Currently, the regression analysis states that as dividend percentages increase, t stock closing price also increases. This is represented by the correlation coefficie correlation. However, the R^2 value for this relationship: 0.144, indicates that the only explain 14.4% of the variation in y. Although there is a slight positive relation these two variables, it is not significant enough to make company dividends a reli fundamental factor for stock market growth.

- However, dividend percentages are still a decent indicator for stock market growt fundamental factor. Just like the earnings per share estimate, there is some corre

5. Regression Analysis of GDP Growth vs Stock Market Growt

In [161]:
```
dj_gdp = pd.merge(gross_output_returns, dowjones_returns, on='qua
#Creating a new dataframe that merges the gross output percent ch
```

In [162]:
```
dj_gdp.columns = dj_gdp.columns.str.replace(' ', '')
#Deleting the excess spaces in the column names
```

In [163]:
```
gross_output_returns = gross_output_returns.apply(pd.to_numeric,
#Converting all accessible values to floats
```

In [164]:
```
dj_gdp['Return'] = [0.0518, -0.0118, -0.0046, 0.0637, 0.0102, 0.0
#The return column that was merged into the gross output returns
#manually create a new column with the same return data as the or
```

In [165]:
```
pd.set_option('display.max_columns', None)
#Makes it easier to see all columns in the dataset; not truncated
```

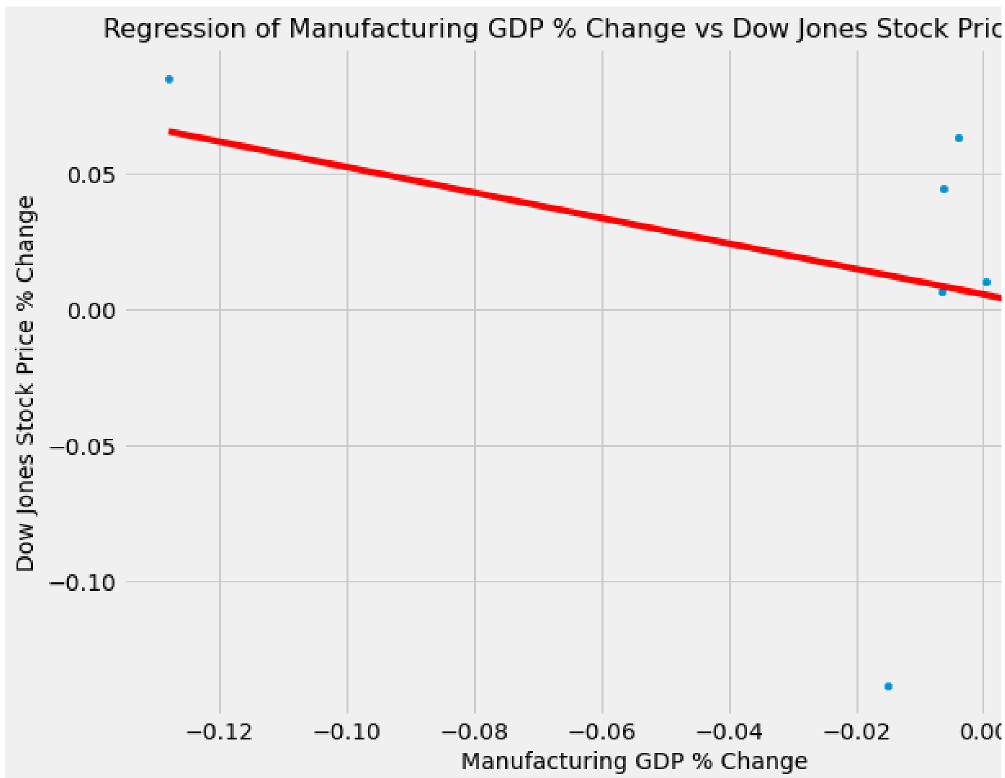
In [166]:
```
from sklearn.linear_model import LinearRegression as linreg
```

In [167]:
```
reg_gdp = linreg().fit(dj_gdp[['Manufacturing']], dj_gdp['Return'
#Creating the regression function for the Manufacturing sector of
```

In [168]:
```
dj_gdp['yhat_manu'] = reg_gdp.predict(dj_gdp[['Manufacturing']])
#Creating the yhat regression line that will be fit onto the scat
```

In [195]:
```
fig, ax = plt.subplots()

dj_gdp.plot.scatter(x='Manufacturing', y='Return', ax=ax)
dj_gdp.sort_values('Manufacturing').set_index('Manufacturing')['y

ax.set_title('Regression of Manufacturing GDP % Change vs Dow Jon
ax.set_ylabel('Dow Jones Stock Price % Change', fontsize=14)
ax.set_xlabel('Manufacturing GDP % Change', fontsize=14)
```

Out[195]: Text(0.5, 0, 'Manufacturing GDP % Change')



In [170]:
```
reg_gdp.score(X = dj_gdp[['Manufacturing']], y = dj_gdp['Return']
```

Out[170]: 0.10109117735834694

Explanation and Analysis:

- Here, the graph shows a slight negative relationship between the percent change manufacturing gdp and the percent change in the dow jones stock prices. This sh as manufacturing gdp increases, the dow jones gdp should decrease respectivel this data is not exactly significant because the R^2 value is only 0.101, which me 10.1% of the variation in y can be explained by the model.
- However, this data shows us that the manufacturing sector/industry exhibits an in relationship with the Dow Jones index's stock prices. This could be because the

manufacturing sector is not accuruately or well represented within the Dow, or tha
manufacturing sector is being overshadowed by all the technology/finance indust
consistently innovating and developing new products.

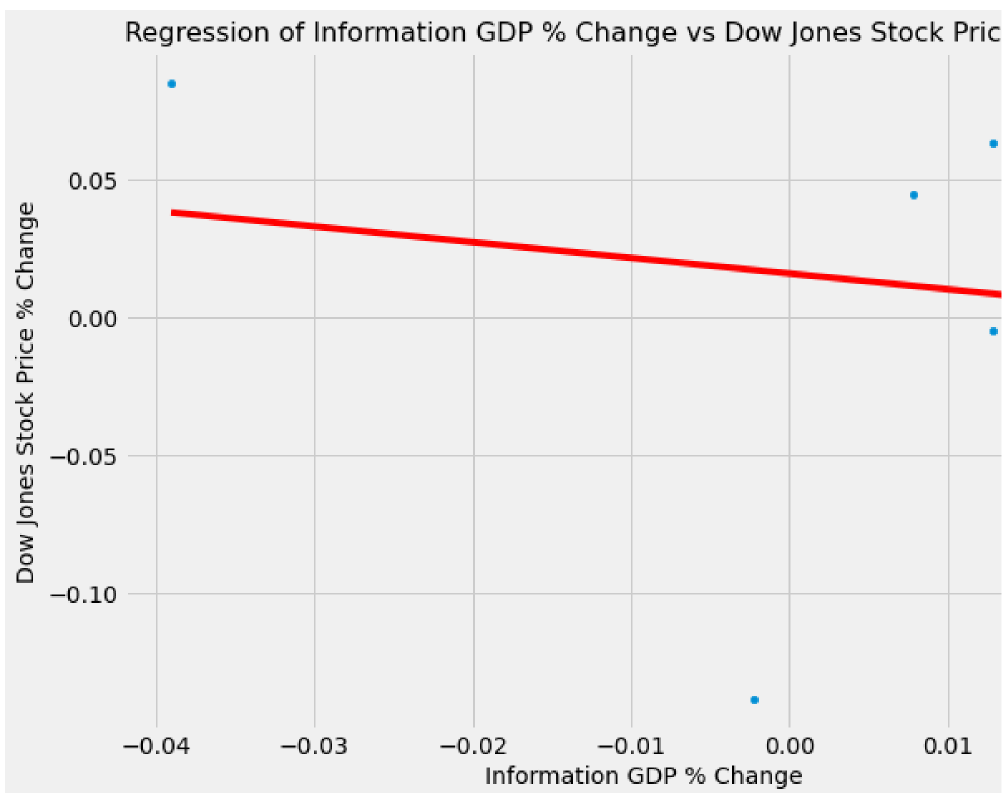
- More data points are required in order to reach a more conclusive assessment.

In [171]:
```
reg_gdp1 = linreg().fit(dj_gdp[['Information']], dj_gdp['Return']
#Creating the regression function for the dow jones gdp percent c
```

In [172]:
```
dj_gdp['yhat_info'] = reg_gdp1.predict(dj_gdp[['Information']])
#Creating the yhat regression line to be fit in a scatter plot
```

In [194]:
```
fig, ax = plt.subplots()

dj_gdp.plot.scatter(x='Information', y='Return', ax=ax)
dj_gdp.sort_values('Information').set_index('Information')['yhat_

ax.set_title('Regression of Information GDP % Change vs Dow Jones
ax.set_ylabel('Dow Jones Stock Price % Change', fontsize=14)
ax.set_xlabel('Information GDP % Change', fontsize=14)
```

Out[194]:
```
Text(0.5, 0, 'Information GDP % Change')
```



In [174]:

```python
reg_gdp1.score(X = dj_gdp[['Information']], y = dj_gdp['Return'])
```

Out[174]: 0.026478919735641093

Explanation and Analysis:

- Here, the graph shows a slight negative relationship between the Information sec percent change and the Dow's stock price percent change, from quarter to quarte analysis shows us that as the information sector's GDP rises, the Dow's stock pri decrease. This is very similar to the analysis from the manufacturing sector show
- Overall, because of the R^2 value of 0.0264, only 2.64% of the variaition in the y be explained by the model. This means that the information sector's GDP data is or fundamental indicator of stock market growth. Although, there is still a slight dc trend as present from the regression data.
- Again, this might be because the information sector is being overshadowed by ot the Dow, or that there simply are not enough data points in order to make a reliat conclusion.

In [197]:

```python
reg_gdp2 = linreg().fit(dj_gdp[['Machinery']], dj_gdp['Return'])
#Creating the regression function for the machinery sector's GDP
```

In [198]:

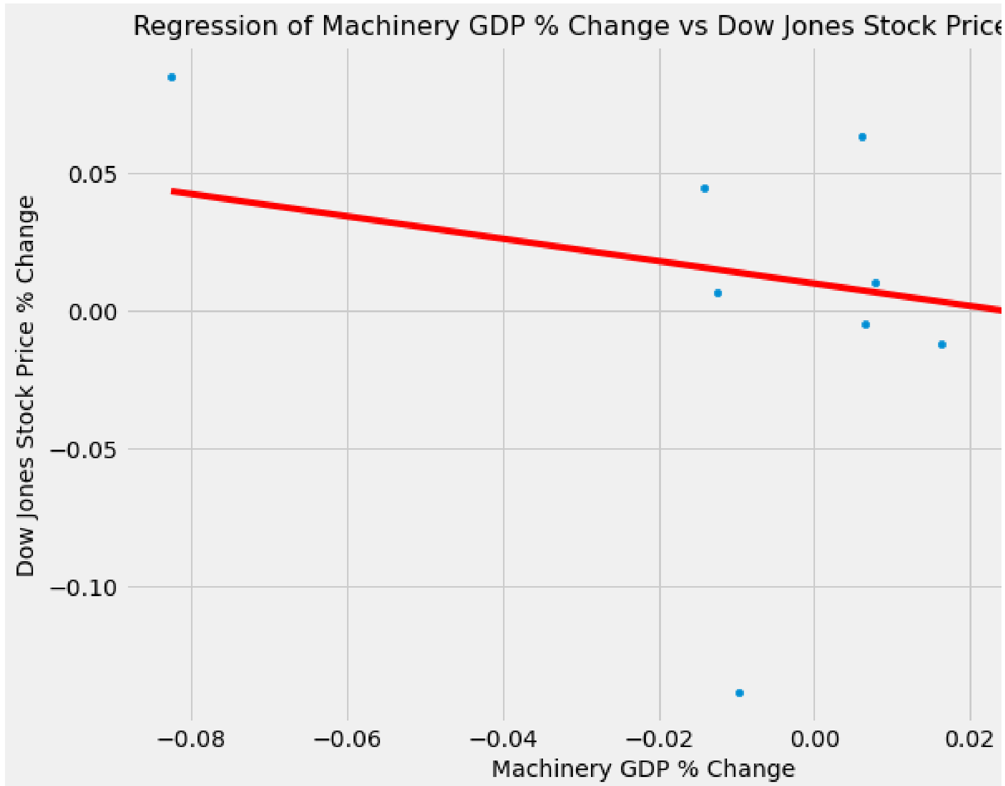
```python
dj_gdp['yhat_mach'] = reg_gdp2.predict(dj_gdp[['Machinery']])
#Creating the yhat line that will be fit to the scatter plot
```

In [193]:

```
fig, ax = plt.subplots()

dj_gdp.plot.scatter(x='Machinery', y='Return', ax=ax)
dj_gdp.sort_values('Machinery').set_index('Machinery')['yhat_mach

ax.set_title('Regression of Machinery GDP % Change vs Dow Jones S
ax.set_ylabel('Dow Jones Stock Price % Change', fontsize=14)
ax.set_xlabel('Machinery GDP % Change', fontsize=14)
```

Out[193]:

```
Text(0.5, 0, 'Machinery GDP % Change')
```



In [178]:
```
reg_gdp2.score(X = dj_gdp[['Machinery']], y = dj_gdp['Return'])
```
Out[178]: 0.043455584352390386

Explanation and Analysis:

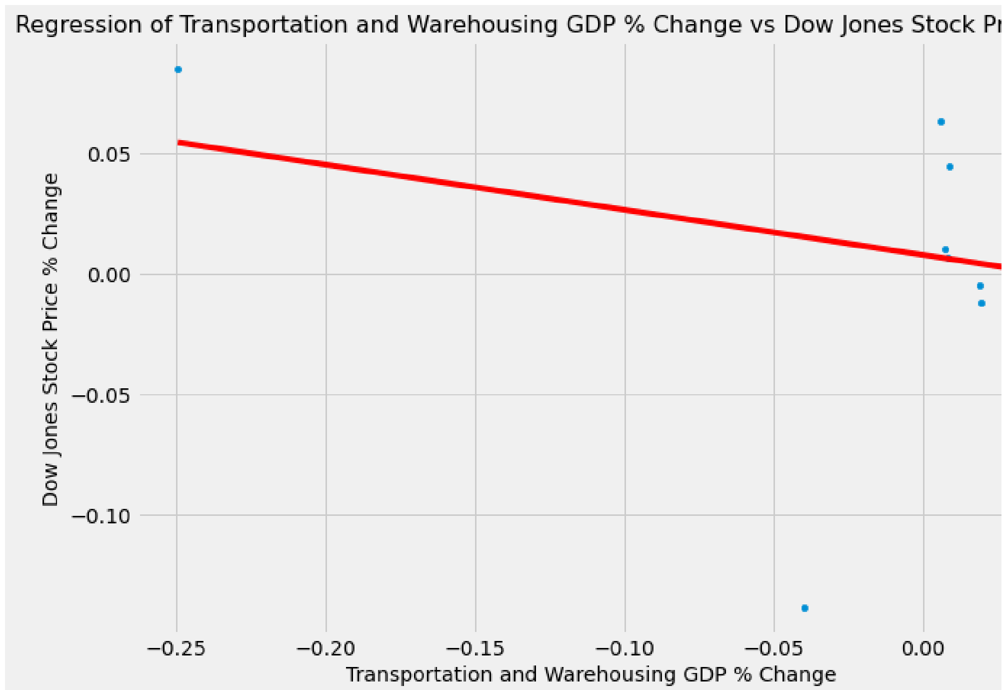
- Here, the regression graph shows us that there is a slight negative relationship b Machinery sector's GDP percent change and the Dow's stock price percent chan means that as the machinery sector's GDP increases, the Dow's stock prices dec
- The R^2 value is very similar to that of the last analysis. An R^2 value of 0.0434 only 4.34% of the variation in the y variable can be explained by the regression m meaning that the Machinery sector is not a reliable or fundamental indicator of st growth.

In [199]:
```
reg_gdp3 = linreg().fit(dj_gdp[['Transportationandwarehousing']],
#Creating the regression function for the Transportation and Ware
```
In [200]:
```
dj_gdp['yhat_trans'] = reg_gdp3.predict(dj_gdp[['Transportationan
#Creating the yhat line to be fit into the scatter plot
```

In [201]:

```
fig, ax = plt.subplots()

dj_gdp.plot.scatter(x='Transportationandwarehousing', y='Return',
dj_gdp.sort_values('Transportationandwarehousing').set_index('Tra

ax.set_title('Regression of Transportation and Warehousing GDP %
ax.set_ylabel('Dow Jones Stock Price % Change', fontsize=14)
ax.set_xlabel('Transportation and Warehousing GDP % Change', font
```

Out[201]: Text(0.5, 0, 'Transportation and Warehousing GDP % Change')



In [182]:

```
reg_gdp3.score(X = dj_gdp[['Transportationandwarehousing']], y =
```

Out[182]: 0.06308237534741967

Explanation and Analysis:

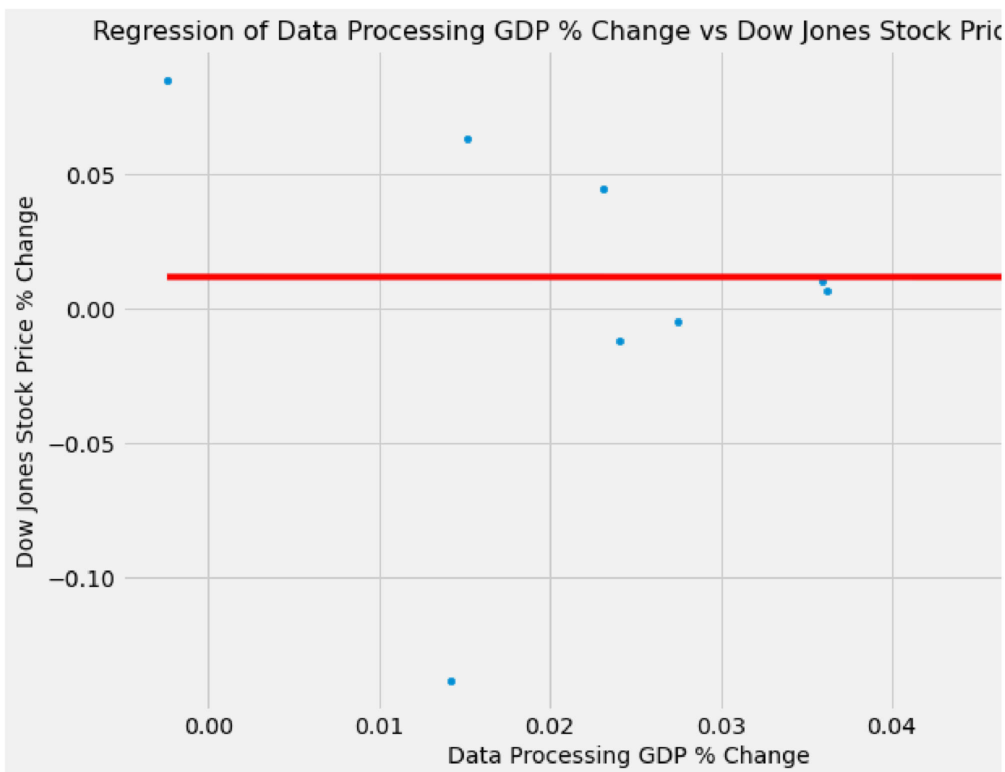
- Here, the regression graph shows us that there exists a slight negative relationsh the Transportation and Warehousing sector's GDP percent change and the Dow's percent change. This means that as the transportation and warehousing sector's increases, the Dow's stock prices would decrease.
- The R^2 value in this case is 0.063, which means that only 6.3% of the variation explained by the model. This is not a reliable or fundamental indicator for determi market growth.

In [203]:
```
reg_gdp4 = linreg().fit(dj_gdp[['Dataprocessing,internetpublishin
#Creating the regression function for the data processing sector'
```

In [204]:
```
dj_gdp['yhat_data'] = reg_gdp4.predict(dj_gdp[['Dataprocessing,in
#Creating the yhat line to be fit to the scatter plot
```

In [191]:
```
fig, ax = plt.subplots()

dj_gdp.plot.scatter(x='Dataprocessing,internetpublishing,andother
dj_gdp.sort_values('Dataprocessing,internetpublishing,andotherinf

ax.set_title('Regression of Data Processing GDP % Change vs Dow J
ax.set_ylabel('Dow Jones Stock Price % Change', fontsize=14)
ax.set_xlabel('Data Processing GDP % Change', fontsize=14)
```

Out[191]: Text(0.5, 0, 'Data Processing GDP % Change')



In [202]:
```
reg_gdp4.score(dj_gdp[['Dataprocessing,internetpublishing,andothe
```

Out[202]: 1.0209375234104812e-08

Explanation and Analysis:

- Here, the graph shows us a completely flat regression line, which indicates absol relationship between the data processing sector's GDP percent change and the D

price percent change. This means that there is no predicting the Dow's stock pric
using the data processing sector's GDP data to predict it.

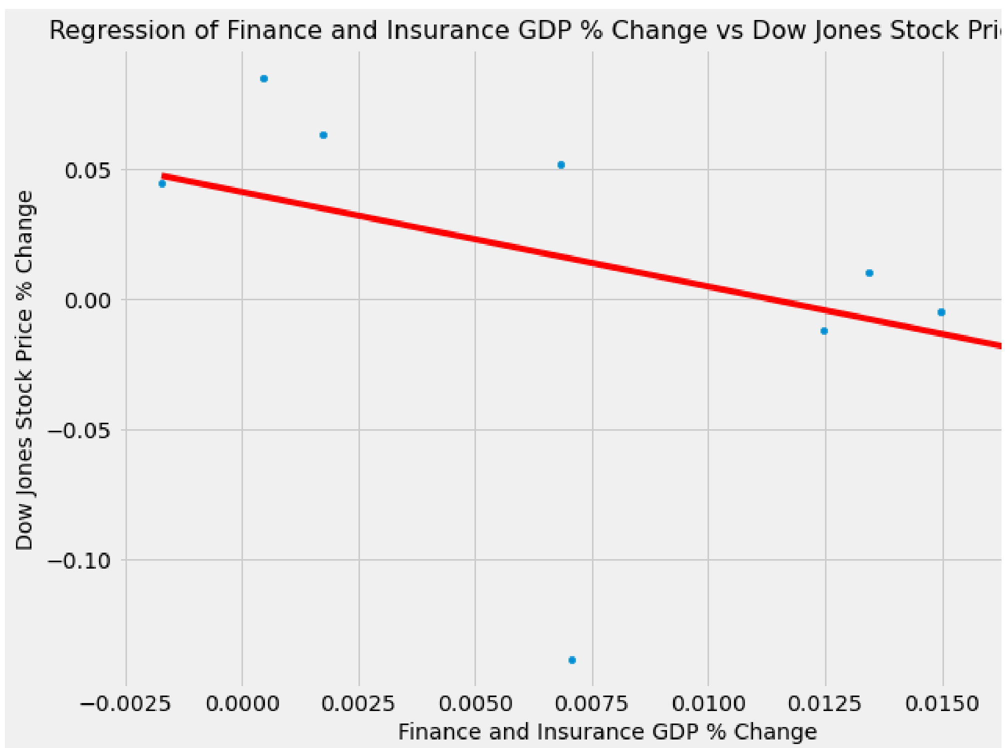
- The R^2 value is extremely small, thus proving that this variable is not a reliable o
fundamental indicator of stock market growth.

In [205]:
```
reg_gdp5 = linreg().fit(dj_gdp[['Financeandinsurance']], dj_gdp['
#Creating the regression function for the Finance and Insurance s
```

In [206]:
```
dj_gdp['yhat_finance'] = reg_gdp5.predict(dj_gdp[['Financeandinsu
#Creating the yhat line to be fit on the scatter plot
```

In [207]:
```
fig, ax = plt.subplots()

dj_gdp.plot.scatter(x='Financeandinsurance', y='Return', ax=ax)
dj_gdp.sort_values('Financeandinsurance').set_index('Financeandin

ax.set_title('Regression of Finance and Insurance GDP % Change vs
ax.set_ylabel('Dow Jones Stock Price % Change', fontsize=14)
ax.set_xlabel('Finance and Insurance GDP % Change', fontsize=14)
```

Out[207]:
```
Text(0.5, 0, 'Finance and Insurance GDP % Change')
```



In [208]:
```
reg_gdp5.score(X=dj_gdp[['Financeandinsurance']], y=dj_gdp['Retur
```
Out[208]:
```
0.1468030800959117
```

Explanation and Analysis:

- Here, the regression graph shows us a negative relationhip between the Finance Insurance sector's GDP percent change and the Dow's stock price percent chang means that as the Finance and Insurance sector's GDP rises, the Dow's stock pr
- The R^2 value is the highest that we've seen in the analyses so far, at 0.1468. Th that 14.68% of the variation in y can be explained by the regression model. This c because a good portion of the Dow's companies include ones from the finance ar sectors. Assuming that this downward, negative relationship cannot be rejected, t that economical GDP (at least for this particular sector) and stock prices are inve correlated. All of the other graphs also suggest this conclusion as well, but not to

In [ ]:
```

```