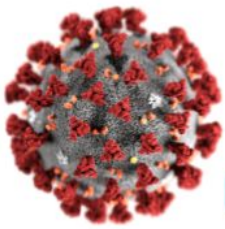


#Covid19

Investigating the relationship between social media activity and the coronavirus

Group 6: Andrew Cummings,
Oliver Posewitz, Ian Ustanik

Final Project Report: #Covid19
IST652 - Scripting for Data Analysis
Group 6: Andrew Cummings, Oliver Posewitz, Ian Ustanik

Data Acquisition

Twitter Data

To investigate the relationship between social media activity and the coronavirus, our group required the use of two separate data sources. The first of these two data sources was coronavirus-related tweets. We utilized the Tweepy package as a wrapper to make a GET request to the Twitter API. To see if there was a difference in results depending on specific tweet content, we search for tweets with keywords #covid19, #maskup, and #travel specifically. We did quickly run into a few constraints while using the Twitter API in that there were significant requests limitations for the basic, non-premium developers package. In order to work around this, we utilized a few pagination methods in order to iterate through the calls in a way that would be under our rate limits. We specifically retrieved tweet ID, tweet text, tweet location, and the time for each tweet. We were able to retrieve 1,000 tweets for each hashtag over a 7-day span, totalling 21,000 tweets in the end. The TwitterAPI jupyter notebook provides the script used to collect all of these tweets.

```
def tweet_search(hashtag):
    query = hashtag
    count = 1000
    searched_tweets = [status for status in tweepy.Cursor(api.search, q=query).items(count)]

    tweet_list = list(searched_tweets)
    tweet_json = [tweet._json for tweet in tweet_list]
    return tweet_json

def tweets_to_df(tweet_json):
    ID = []
    tweets = []
    location = []
    time = []

    for tweet in tweet_json:
        ID.append(tweet['id'])
        tweets.append(tweet['text'])
        location.append(tweet['user']['location'])
        time.append(tweet['created_at'])

    data_tuples = list(zip(ID, tweets,location,time))
    tweets_df = pd.DataFrame(data_tuples, columns=['ID','tweets','location','time'])
    return tweets_df
```

After importing the necessary packages (tweepy, json, pandas), we set up the Twitter API access configuration. We then defined a function for calling the Twitter API, taking a specific hashtag as input and returning the queried tweets as a json object. Next, we defined a function for converting that json object to a dataframe with the desired column attributes. Finally we then used a loop to iterate through hashtag searches until the user typed QUIT. Each of these

dataframes were then output to .csv file. Below is an example of a hashtag search using our defined functions and the resulting dataframe:

```
# Loop iterating through hashtag searches for the user
while True:
    hashtag = input("Please Enter A Hashtag Or Type 'QUIT': ")
    if hashtag == 'QUIT':
        break
    Day = tweets_to_df(tweet_search(hashtag))
```

```
Please Enter A Hashtag Or Type 'QUIT': #maskup until:2020-11-16 since:2020-11-15
Please Enter A Hashtag Or Type 'QUIT': QUIT
```

	ID	tweets	location	time
0	1328125627812114433	RT @ArTalks: GodSpeed @SpaceX ...	Los Angeles Universe Omniverse	Sun Nov 15 23:59:59 +0000 2020
1	1328125567271538688	As if we needed evidence that the Pandemic is ...	#notinabubble	Sun Nov 15 23:59:45 +0000 2020
2	1328125516222754818	RT @nonsequiteuse: Hey @HiltonHouston & @H...	Houston, Texas	Sun Nov 15 23:59:33 +0000 2020
3	1328125492545798144	RT @cartoonnetwork: Rep your fav #CartoonNetwo...	someone's mind	Sun Nov 15 23:59:27 +0000 2020
4	1328125490255835136	Waiting to take a pandemic seriously until you...	Norman, OK	Sun Nov 15 23:59:26 +0000 2020
...	...	...	...	...
995	1328063679926038531	RT @Alt_ReddTruq: #MaskUp	Florida, USA	Sun Nov 15 19:53:50 +0000 2020
996	1328063582974734339	Do you think racist folks will finally get #BL...	Texas, USA	Sun Nov 15 19:53:27 +0000 2020
997	1328063420739031043	RT @cartoonnetwork: Rep your fav #CartoonNetwo...	Naahhhh, have a cawffee -w-	Sun Nov 15 19:52:48 +0000 2020
998	1328063393438322688	For those of you that need a reminder. #maskup...	New York, NY	Sun Nov 15 19:52:41 +0000 2020
999	1328063376891801600	RT @cartoonnetwork: Rep your fav #CartoonNetwo...		Sun Nov 15 19:52:37 +0000 2020

Covid-19 Data

For our second data source, we required Covid data that included new case counts for each day by state. We found the numbers we needed in the Cases and Deaths by State portion of the CDC website. The COVID data was downloaded in CSV format and turned into a dataframe using the read_csv method from the pandas. Our initial data consisted of 18120 rows and 15 columns.

	submission_date	state	tot_cases	conf_cases	prob_cases	new_case	pnew_case	tot_death	conf_death	prob_death	new_death	pnew_death	created_at	consent_cases	consent_deaths
0	01/22/2020	CO	0	NaN	NaN	0	NaN	0	NaN	NaN	0	NaN	03/26/2020 04:22:39 PM	Agree	Agree
1	01/23/2020	CO	0	NaN	NaN	0	NaN	0	NaN	NaN	0	NaN	03/26/2020 04:22:39 PM	Agree	Agree
2	01/24/2020	CO	0	NaN	NaN	0	NaN	0	NaN	NaN	0	NaN	03/26/2020 04:22:39 PM	Agree	Agree
3	01/25/2020	CO	0	NaN	NaN	0	NaN	0	NaN	NaN	0	NaN	03/26/2020 04:22:39 PM	Agree	Agree
4	01/26/2020	CO	0	NaN	NaN	0	NaN	0	NaN	NaN	0	NaN	03/26/2020 04:22:39 PM	Agree	Agree
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
18115	11/14/2020	PW	0	NaN	NaN	0	0.0	0	NaN	NaN	0	0.0	11/15/2020 03:12:13 PM	NaN	NaN
18116	11/15/2020	PW	0	NaN	NaN	0	0.0	0	NaN	NaN	0	0.0	11/16/2020 06:40:02 PM	NaN	NaN
18117	11/16/2020	PW	0	NaN	NaN	0	0.0	0	NaN	NaN	0	0.0	11/17/2020 02:52:47 PM	NaN	NaN
18118	11/17/2020	PW	0	NaN	NaN	0	0.0	0	NaN	NaN	0	0.0	11/18/2020 02:57:46 PM	NaN	NaN
18119	11/18/2020	PW	0	NaN	NaN	0	0.0	0	NaN	NaN	0	0.0	11/19/2020 03:01:36 PM	NaN	NaN

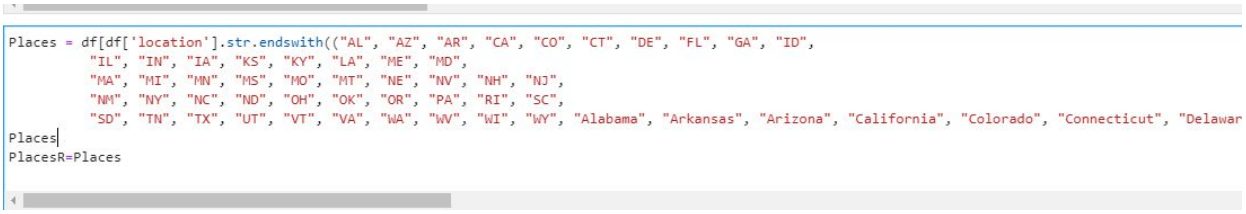
18120 rows × 15 columns

The raw data contained various descriptions of Covid cases and deaths for each day since the 22nd of January 2020 until the 19th of November 2020 (dataset updated everyday to reflect the day before accessing). These included attributes such as probable cases, confirmed deaths, new cases, state, and date. The data included information for all 50 states as well as the U.S territories and DC. From this set we were primarily interested in the new cases column.

Data Exploration and Data Cleaning

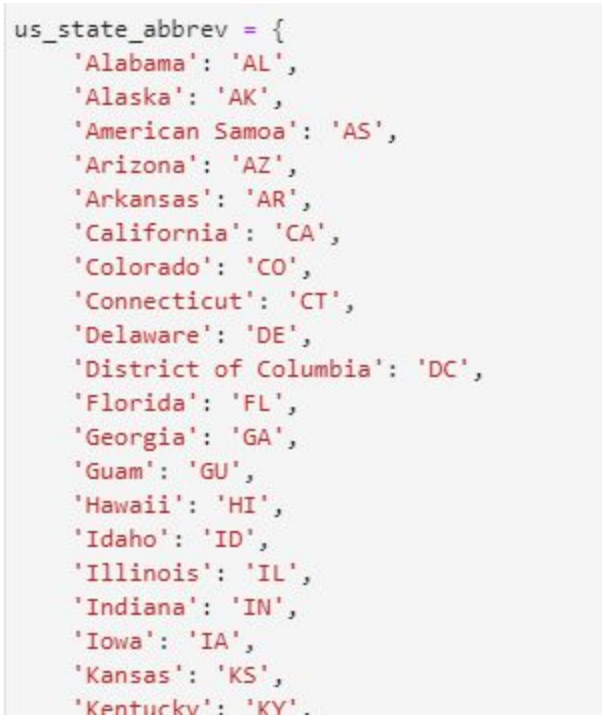
Cleansing and Purging

After collecting both sets of data, it was apparent that cleansing and purging were necessary in both cases. In the case of the tweets dataset, we had many tweets containing no location data to match them to the Covid dataset. The ones that contained location data lacked standardization, for example, we saw tweets with location " Northern Arkansas" , "Pittsburgh, PA", or simply "Pennsylvania". To start, we remove tweets that contain none of the state abbreviations or state names.



```
Places = df[df['location'].str.endswith(("AL", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA", "ID",
          "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
          "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ",
          "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC",
          "SD", "TN", "TX", "UT", "VT", "VA", "WA", "WV", "WI", "WY", "Alabama", "Arkansas", "Arizona", "California", "Colorado", "Connecticut", "Delawar
Places
PlacesR=Places
```

From there, we now have 4456 unique tweets from the 48 continental United States (we chose the 48 for mapping ease). From there, we standardized the tweet locations that we had, removing the words before the state name and turning the state name into the state abbreviation using a dictionary.



```
us_state_abbrev = {
    'Alabama': 'AL',
    'Alaska': 'AK',
    'American Samoa': 'AS',
    'Arizona': 'AZ',
    'Arkansas': 'AR',
    'California': 'CA',
    'Colorado': 'CO',
    'Connecticut': 'CT',
    'Delaware': 'DE',
    'District of Columbia': 'DC',
    'Florida': 'FL',
    'Georgia': 'GA',
    'Guam': 'GU',
    'Hawaii': 'HI',
    'Idaho': 'ID',
    'Illinois': 'IL',
    'Indiana': 'IN',
    'Iowa': 'IA',
    'Kansas': 'KS',
    'Kentucky': 'KY',
```

```
for location in PlacesR['location']:
    #print(location)

    for state in us_state_abbrev:
        if location.endswith(state):
            PlacesR['location'] = PlacesR['location'].replace([location],us_state_abbrev[state])


PlacesR
```

From there, we converted the time of the tweet into date format, to match our Covid dataset's daily numbers. With that, our tweet set was cleansed and transformed for analysis.

For the covid dataset, we only wanted the new case counts in the last seven days to match the twitter dataset, so any dates before the 12th of november were removed. We only wanted information from the 48 continental states, so the data from the other states and territories were removed. We also formatted the dates to dataTime using the Pandas dataframe functions. We went from 18120 rows to 288 rows using these metrics. At this point, the datasets can be combined and analyzed.

```
dfTime=df[df['submission_date']>'11/12/2020']
### define condition as cond_
cond=(dfTime["state"].isin(states)) #set our condition to sort through the dataset, removing unwanted states and territories
dfTimeReduced=dfTime.loc[cond,:]#Locate condition and include values that meet condition
```

```
dfTimeReduced
```

	submission_date	state	tot_cases	conf_cases	prob_cases	new_case	pnew_case	tot_death	conf_death	prob_death	new_death	pnew_death	created_at	consent_cases	consent_deaths
296	2020-11-13	CO	154038	146023.0	8015.0	6439	157.0	2504	2096.0	408.0	36	0.0	11/14/2020 02:44:40 PM	Agree	Agree
297	2020-11-14	CO	159234	151115.0	8119.0	5196	104.0	2525	2116.0	409.0	21	1.0	11/15/2020 03:12:13 PM	Agree	Agree
298	2020-11-15	CO	163417	155258.0	8159.0	4183	40.0	2546	2137.0	409.0	21	0.0	11/16/2020 06:40:02 PM	Agree	Agree
299	2020-11-16	CO	167713	158864.0	8849.0	4296	690.0	2578	2153.0	425.0	32	16.0	11/17/2020 02:52:47 PM	Agree	Agree
300	2020-11-17	CO	172044	163076.0	8968.0	4331	119.0	2608	2177.0	431.0	30	6.0	11/18/2020 02:57:46 PM	Agree	Agree
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
16907	2020-11-14	VT	2843	NaN	NaN	100	0.0	59	NaN	NaN	0	0.0	11/15/2020 03:12:13 PM	Not agree	Not agree
16908	2020-11-15	VT	2889	NaN	NaN	46	0.0	59	NaN	NaN	0	0.0	11/16/2020 06:40:02 PM	Not agree	Not agree
16909	2020-11-16	VT	3008	NaN	NaN	119	0.0	59	NaN	NaN	0	0.0	11/17/2020 02:52:47 PM	Not agree	Not agree
16910	2020-11-17	VT	3104	NaN	NaN	96	0.0	60	NaN	NaN	1	0.0	11/18/2020 02:57:46 PM	Not agree	Not agree
16911	2020-11-18	VT	3161	NaN	NaN	57	0.0	60	NaN	NaN	0	0.0	11/19/2020 03:01:36 PM	Not agree	Not agree

288 rows × 15 columns

Combining Different Data Sets

For each request sent through the Tweepy API, data was returned and then stored as a comma separated file. Due to data request limitations, pagination was introduced as a work around. This resulted in the data being horizontally scaled. There were separate csv files for each bin, or category, and within each bin there were separate files for each day of the five day period we analyzed. In order to consolidate the data, a for loop was iterated over each category. Once all days of a single category were combined into one dataframe, a new column labeled "bin" was attached and the string containing the category descriptor was inserted into each cell for that column. This process was done for each of the three bins we collected tweet data from, resulting in three main dataframes representing each of the three categories. Then the

dataframes were vertically concatenated with each other, resulting in one main dataframe that's easy to analyze.

```
#stitch csv's together for each bin

#name the directory folder. Different categories are in different folders.
directory = 'travel'

li = []

#get list of filenames and read in to append to list
for csv_file in os.listdir(directory):
    mask_df = pd.read_csv(directory+'/'+csv_file, index_col=None, header = 0)
    li.append(mask_df)

#convert to df
travel_df = pd.concat(li, axis=0, ignore_index=True)

#export combined bin to csv
travel_df.to_csv(directory+'.csv')
```

Aggregation and Summarization

There are many ways to summarize this type of natural language data. As a group, we decided to use the Natural Language Toolkit as it contained modules that specifically handles punctuation, capitalization, degree modifiers, conjunctions, and various other free form text challenges. We decided to use the Vader module for sentiment analysis as it returns various yet condensed fields of sentiment calculations.

A sentiment scoring function was created that handled the tweet text data as an argument and returned the Vader score output. This function was applied for each row and saved the Vader output into the newly formed scores column. Since the Vader module returns a dictionary, a dictionary of scores were stored in each cell of the scores column of the dataframe. Then each key of the dictionaries were expanded into their own column labelled pos, neg, neu, compound. The corresponding sentiment values for each row in each column were inserted into their respective index location.

```
#run analysis of all tweets

sid = SentimentIntensityAnalyzer()

all_df = pd.read_csv('all.csv')
all_df = all_df.drop(columns=['Unnamed: 0'])


def scorer(x):
    scores = sid.polarity_scores(x)
    return scores

all_df['scores']=all_df.tweets.apply(func=scorer)
```

```
#Make each score value it's own column
all_df = pd.concat([all_df.drop(['scores'],
                                axis=1), all_df['scores'].apply(pd.Series)],
                   axis=1)
```

This process resulted in the final dataframe containing sentiment fields for each tweet that would enable summarization of sentiment across multiple different grouping methods.

	bin	location	time	tweets	neg	neu	pos	compound
0	mask	NaN	Fri Nov 13 23:59:50 +0000 2020	Exactly! Jersey Stand up! #maskon #maskup #COV...	0.000	1.000	0.000	0.0000
1	mask	NaN	Fri Nov 13 23:59:41 +0000 2020	RT @stromulus: Republicans are right about one...	0.159	0.841	0.000	-0.5574
2	mask	NaN	Fri Nov 13 23:58:42 +0000 2020	RT @StevensonRec: The @CDCgov officially ackno...	0.000	0.874	0.126	0.3818
3	mask	Miami, FL	Fri Nov 13 23:58:31 +0000 2020	Pennsylvania nurses plan to go on strike as co...	0.070	0.798	0.131	0.3182
4	mask	NaN	Fri Nov 13 23:58:30 +0000 2020	#MaskUp	0.000	1.000	0.000	0.0000

Methods of Analysis

Question 1: What is the current state of Covid-19?

Fields Used – Covid Cases (States), New Cases (States), Case timeline by state and Total new case count nationwide

Computation – To answer this question, we used the matplotlib and pandas libraries to get a picture of what trends in new covid cases looked like by state and overall count over the 7 days we collected data from. We wanted to see which states had the most new cases per capita. We started by removing unwanted columns, such as the death metrics and the data certifications. From there we used the pandas groupby and nlargest methods to look at the top 3 states in terms of total new case count. We then use the plot function in matplotlib with the sort_values index method to show the new cases by state over the last 7 days. For our third graphic we

want to look at the new cases totals over the last seven days. Graphics are attached below. Finally, we use the folium package to create a map of the covid data in terms of increase in cases per capita.

Results – See below graphics for further detail:

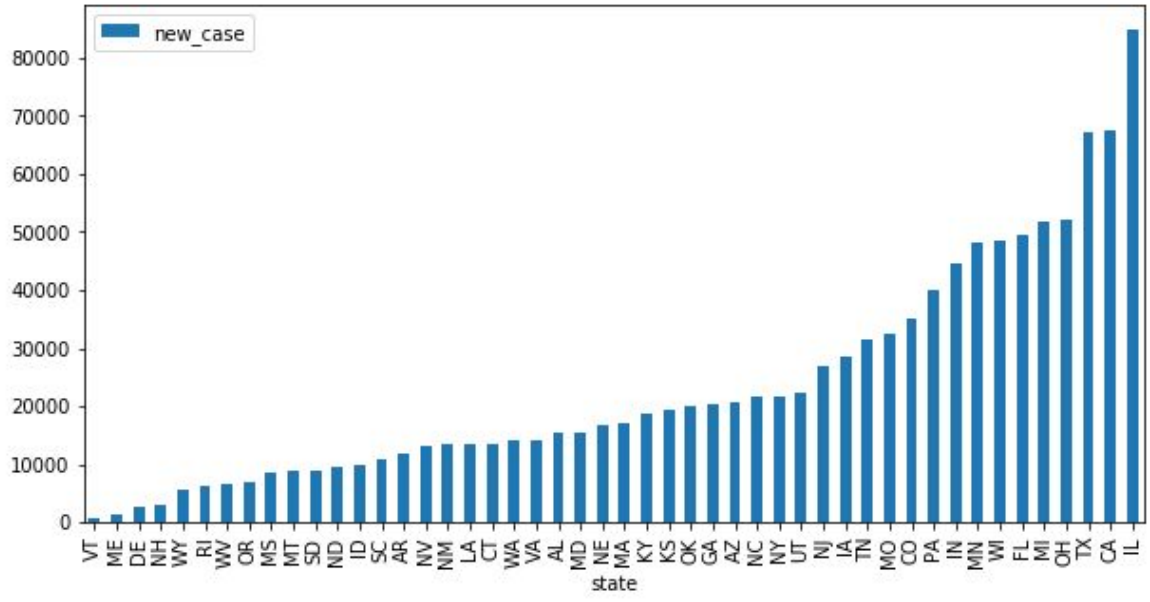
state	tot_cases	conf_cases	new_case	tot_death	conf_death	prob_death	new_death
IL	4099809	4099809.0	84841	78778	75762.0	3016.0	802
CA	7200493	0.0	67658	127996	0.0	0.0	358
TX	7216532	0.0	67042	137620	0.0	0.0	966

From our first computation, we see that Illinois had the largest new case count with 84841 over the 7 day period, with California and Texas coming in second and third.

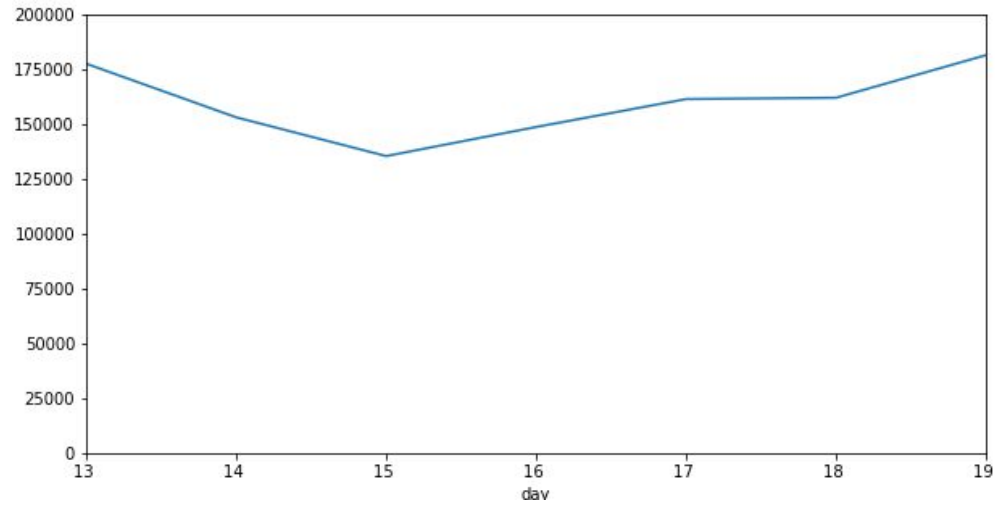
```
#plot new cases by state
s_plot = by_state_covid.reset_index().sort_values(by='new_case')

fs = (10, 5)
s_plot.plot(x='state',y='new_case', kind='bar', figsize=fs)
```

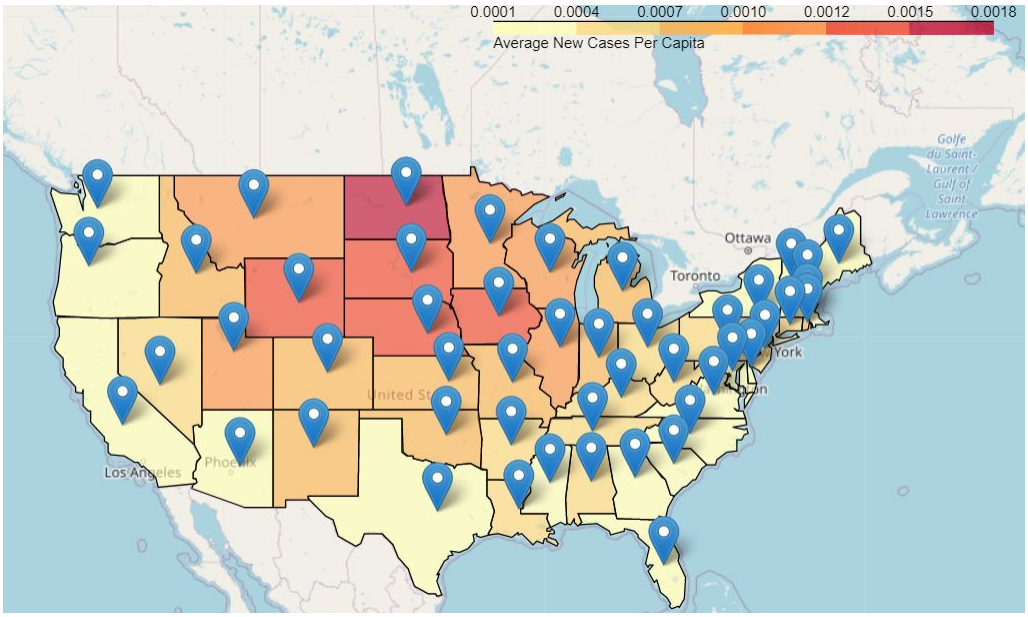
```
<matplotlib.axes._subplots.AxesSubplot at 0x1d8565b9c50>
```



We can see above that Vermont and Maine had the lowest increase over the last 7 days, with a somewhat steady trend upward as we go through states.



We see that, over the last 7 days there was not a constantly increasing change of new cases by day.

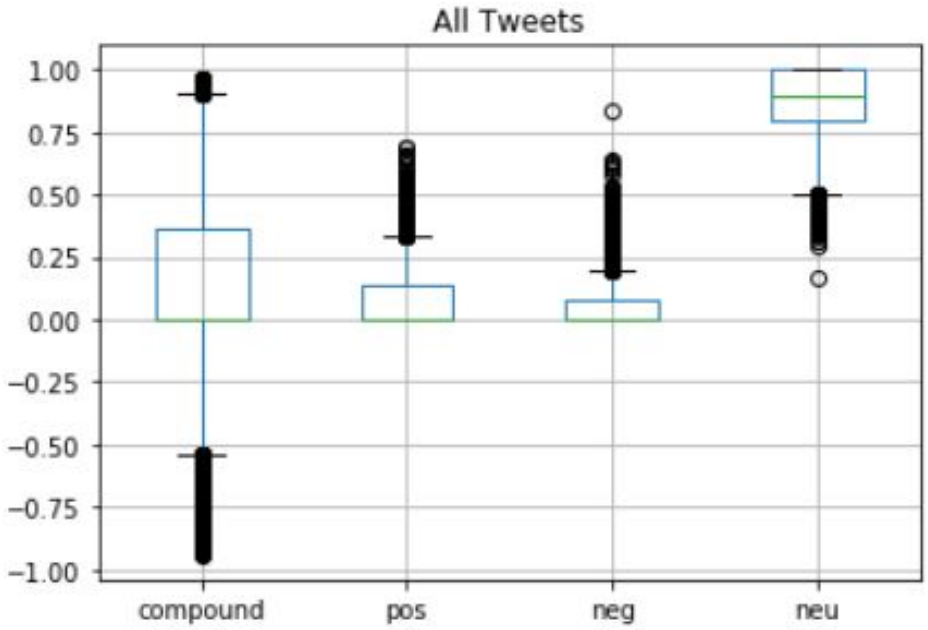


From this graph we can see that although Texas and Illinois had the largest case increase, per capita the midwestern states of the Dakotas and Minnesota had a scaled larger increase in cases over the 7 days.

Question 2: What is the sentiment of coronavirus-related tweets?

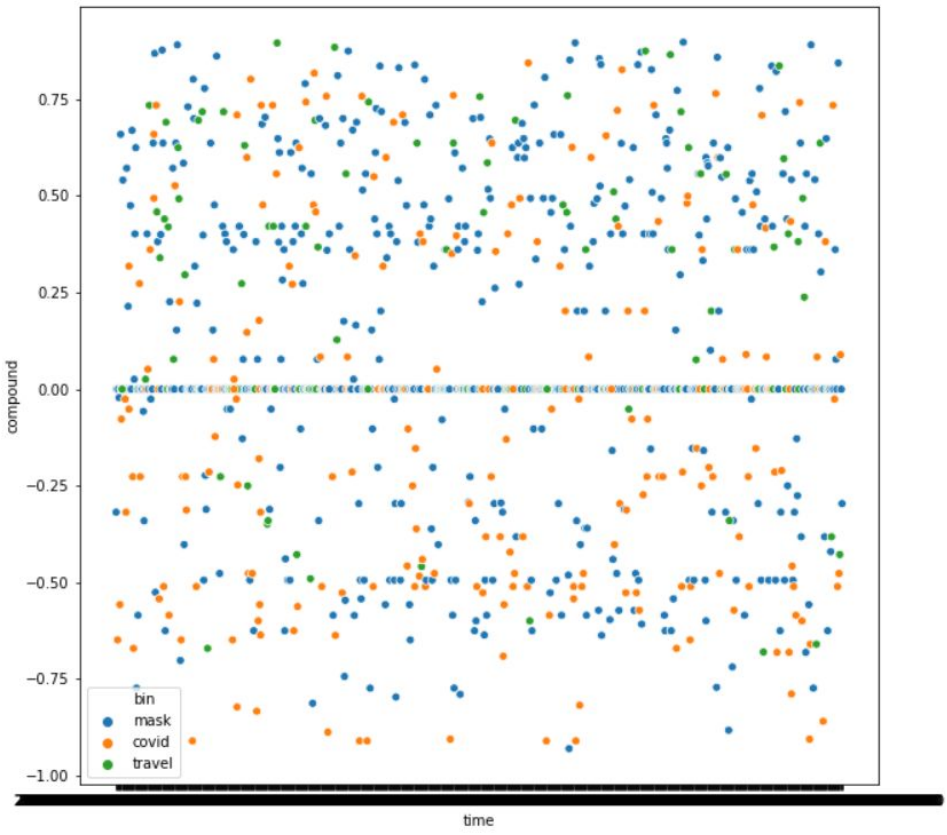
Fields Used – Tweet Text, Tweet Location, Bin (Hashtag) → Negative, Neutral, Positive, and Compound scores

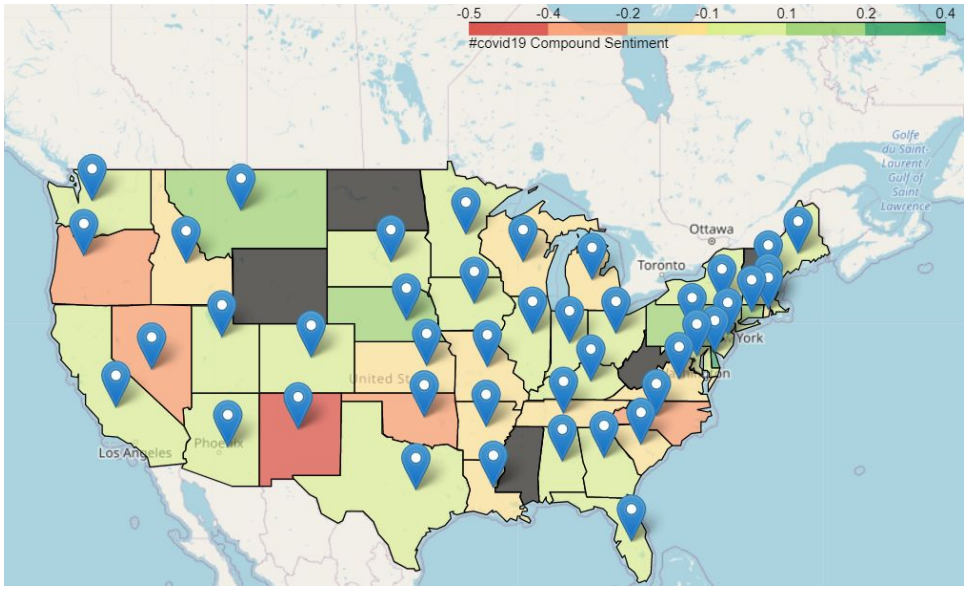
Computation – To answer this question and carry out the specific method of analysis, our group used the VADER model as part of the Natural Language Toolkit python package. VADER stands for Valence Aware Dictionary for Sentiment Reasoning and is the model that we used to perform a sentiment analysis for each tweet. The text sentiment analysis is sensitive to both polarity and intensity of emotion. We used a simple function to iterate through each tweet for all hashtags and for all locations. This was accomplished by creating a SentimentIntensityAnalyzer() object and calling the polarity_scores() method, applying the function across all tweets. Through the analysis, we were able to produce negative, neutral, positive, and compound sentiment scores for each tweet and then compare sentiment across locations (states) and bins (hashtags).



Results – From our sentiment analysis we concluded that while #travel had the highest compound score, #covid19 had the highest negative score, and #maskup had the highest positive score. States like New Mexico and North Carolina experienced a high percent change of new cases over the 7-day period, while they received high negative scores for #covid19 during the same time period. In the end, sentiment scores were somewhat consistent over the 7-day period and a possible correlation between social media sentiment and Covid-19 in the United States warranted further investigation.

	bin	neg	neu	pos	compound
2	travel	0.020455	0.893188	0.086358	0.170609
1	mask	0.049105	0.854347	0.096553	0.105901
0	covid	0.081858	0.845110	0.073033	-0.029543





Question 3: Does a correlation exist between social media sentiment and Covid-19 in the United States?

Fields Used – New Cases (States) → Negative, Neutral, Positive, and Compound scores

Computation – At this point our group had prepared Covid-19 data and prepared sentiment scores for the tweets that we collected. We could now further investigate the relationship between social media sentiment and Covid-19 in the United States. Our group chose to perform both regression analysis and location analysis to investigate the correlation. For the regression analysis we utilized python's built-in scikit learn package. More specifically, we used the LinearRegression() method to fit multiple models of various attributes.

```
#set intital regression model, with time and new case count pr
X = data[['time','new_case']] # compound~time+new_case
Y = data['compound']

# with sklearn
regr = linear_model.LinearRegression()#main call for using lin
regr.fit(X, Y)#import in our dependent and independent variabl

print('Intercept: \n', regr.intercept_)
print('Coefficients: \n', regr.coef_)
#with statsmodels
X = sm.add_constant(X) # adding a constant

model = sm.OLS(Y, X).fit()
predictions = model.predict(X)
```

OLS Regression Results

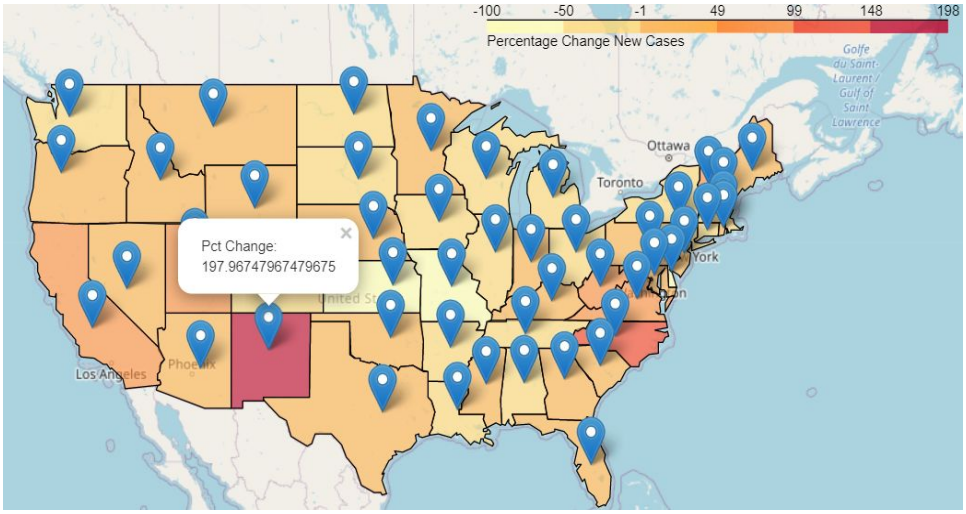
Dep. Variable:	compound	R-squared:	0.003
Model:	OLS	Adj. R-squared:	0.003
Method:	Least Squares	F-statistic:	957.7
Date:	Mon, 23 Nov 2020	Prob (F-statistic):	6.04e-210
Time:	14:15:08	Log-Likelihood:	-1.6245e+05
No. Observations:	303000	AIC:	3.249e+05
Df Residuals:	302998	BIC:	3.249e+05
Df Model:	1		
Covariance Type:	nonrobust		

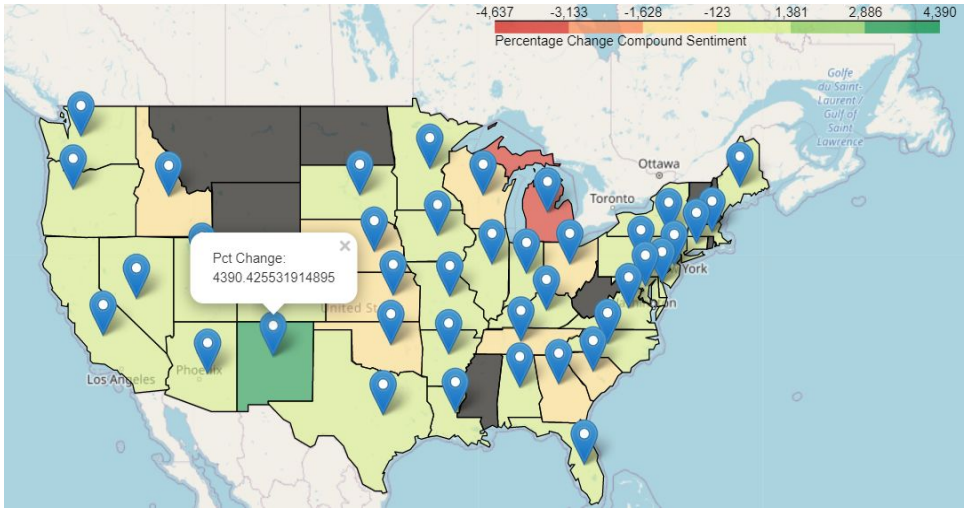
	coef	std err	t	P>|t|	[0.025	0.975]
const	-8408.2516	271.705	-30.946	0.000	-8940.786	-7875.718
time	0.0114	0.000	30.946	0.000	0.011	0.012

Omnibus:	7583.644	Durbin-Watson:	0.006
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3814.192
Skew:	-0.005	Prob(JB):	0.00
Kurtosis:	2.450	Cond. No.	2.67e+11

For the location analysis we generated multiple interactive maps to visually compare and contrast the relationship between tweet sentiment and new cases across the lower 48 states. Our group used the folium package to produce these maps for aggregated datasets in order to view percent change in new cases and tweet sentiment over the 7-day period specifically.

Results – Our linear regression models failed to produce significant results. This was due to the fact that we were limited by time frame and volume of publicly accessible tweets. The data was thoroughly tested for any correlation and this same process projected onto a larger scale would undoubtedly yield more statistically significant results. Our location analysis did provide credible results and it appeared that some states with high percent changes of new cases also experienced high percent changes of compound sentiment score:





Program Description

Explanation of workflow and order of script execution. The following list contains the script names with their corresponding input/outputs:

1. Get Tweets ("1_Twitter_API.ipynb")
    a. Input:

        i. Query parameters

        ii. File name and save location

    b. Output:

        i. Tweepy API response to csv

        ii. Csv for each day (five days) with a query for each category or bin (3 bins)

        iii. Example: "covid1913.csv", "mask14.csv", "travel19.csv", …

2. Combine Tweets ("2_final_project_script")

    a. Input:

        i. Directory name containing all csv files output from the "1_Twitter_API.ipynb" script

    b. Output:

        i. One csv file containing all twitter data with sentiment scores ("all_scores.csv")

3. Get Covid Data ("3_CovidData.ipynb")

    a. Input:

i. csv from CDC website (“US_COVID_FINAL.csv”)

b. Output:

i. “covidData.csv”

4. Extract locations from tweets then insert (“4_StatesLocationFull.ipynb”)

a. Input:

i. “all_scores.csv”

b. Output:

i. “all_states.csv”

5. Map out data for analysis (“5_Maps.ipynb”)

a. Input:

i. “covidData.csv”, “center.csv”, “allStatesR.csv”, “2019Pops.csv”, “WMC3-us-states.json”

b. Output:

i. Map plots

6. Regression Analysis (“6_AnalysisFinal.ipynb”)

c. Input:

i. “allStatesR.csv”, “2019Pops.csv”

d. Output:

i. Various regression models

Output Description

The following list corresponds with each element of the previous list, further explaining the output of each script file.

1. Get Tweets
    a. The files that were returned from the Tweepy requests. Each file represents the return of one single day for one specific category.
2. Combine All Tweets

    a. This csv file represents all the Twitter data collected, along with their corresponding sentiment analysis computations.
3. Get Covid Data
    a. This file represents a cleaned, Pandas friendly version of the data collected from the CDC website.
4. Extract Locations
    a. This file is the same shape as the output of “combine all Tweets” output, but with more accurate location information inserted into their respective locations.
5. Map Data Analysis
    a. These outputs are plots that display the processed data geographically.
6. Regression Analysis
    a. These outputs are various regression models fit to investigate the direct correlation between social media activity and the coronavirus.

Group Tasks/Roles

In order to complete a project of this scope and magnitude effectively, we as a group chose to assign tasks to each group member who each fulfilled a particular role. These roles and responsibilities were as follows:

Andrew Cummings - Data Acquisition (Covid-19), Data Preprocessing (Cleansing), Methods of Analysis (Regression)

Oliver Posewitz - Data Preprocessing (Compilation & Aggregation), Methods of Analysis (Statistical, NLP - Sentiment)

Ian Ustanik - Data Acquisition (Twitter), Data Preprocessing (Cleansing), Methods of Analysis (Location)

Conclusion

Our Covid-19 analysis yielded relevant information on the spread of Covid in the last 7 days, showing which states were hit hardest by number and scaled for their population. We were surprised by the fact that the increase in new cases did not go up over the seven days linearly.

From our sentiment analysis, it was clear that #travel had a significantly more positive score than that of #maskup and #covid. This would imply that tweets using this hashtag are more positive overall, something that we pictured but had not proved before the project.

Bringing the datasets together for our final question, we failed to achieve statistically significant results that Covid-19 and the hashtags associated with the pandemic had correlation. Having said that, we have no doubts that the data was thoroughly tested for any correlation. Our largest limitation was the time frame of publicly accessible tweets. Our results were limited by the 7-day

window of data we collected; had we used a larger window there would have definitely been some more actionable insights.

Appendix Visualizations

Below you will find additional visualizations including further regression analysis and map plots:

```
Intercept:
 -8402.00634020658
Coefficients:
 [ 1.13888485e-02 -8.85595420e-07]
                            OLS Regression Results
==============================================================================
Dep. Variable:               compound   R-squared:                       0.003
Model:                            OLS   Adj. R-squared:                  0.003
Method:                 Least Squares   F-statistic:                     482.8
Date:                Tue, 01 Dec 2020   Prob (F-statistic):          4.46e-210
Time:                        09:42:07   Log-Likelihood:            -1.6245e+05
No. Observations:              303000   AIC:                         3.249e+05
Df Residuals:                  302997   BIC:                         3.249e+05
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -8402.0063    271.711    -30.923      0.000   -8934.552   -7869.461
time           0.0114      0.000     30.923      0.000       0.011       0.012
new_case   -8.856e-07   3.15e-07     -2.815      0.005    -1.5e-06   -2.69e-07
==============================================================================
Omnibus:                     7574.269   Durbin-Watson:                   0.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3810.962
Skew:                          -0.005   Prob(JB):                         0.00
Kurtosis:                       2.451   Cond. No.                     2.67e+11
==============================================================================
```

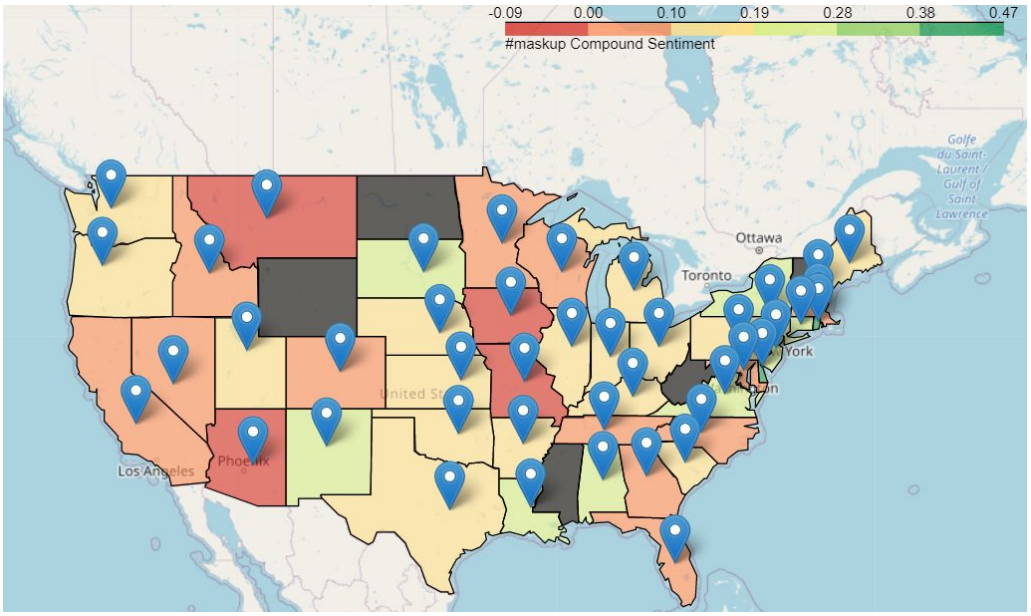
above regression:compound~time+new_case

```
Intercept:
 -8408.251607969973
Coefficients:
 [0.01139731]
                            OLS Regression Results
==============================================================================
Dep. Variable:               compound   R-squared:                       0.003
Model:                            OLS   Adj. R-squared:                  0.003
Method:                 Least Squares   F-statistic:                     957.7
Date:                Tue, 01 Dec 2020   Prob (F-statistic):          6.04e-210
Time:                        09:42:07   Log-Likelihood:            -1.6245e+05
No. Observations:              303000   AIC:                         3.249e+05
Df Residuals:                  302998   BIC:                         3.249e+05
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -8408.2516    271.705    -30.946      0.000   -8940.786   -7875.718
time           0.0114      0.000     30.946      0.000       0.011       0.012
==============================================================================
Omnibus:                     7583.644   Durbin-Watson:                   0.006
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3814.192
Skew:                          -0.005   Prob(JB):                         0.00
Kurtosis:                       2.450   Cond. No.                     2.67e+11
==============================================================================
```

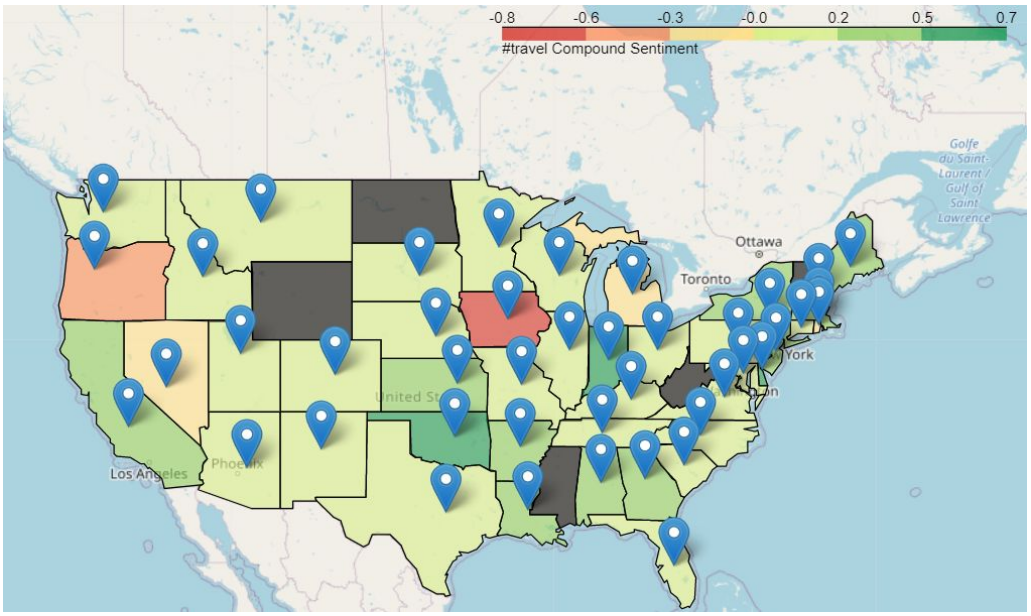
above regression:compound~time

```
Intercept:
 6803742.099321884
Coefficients:
 [ -9.22021894 -29.53207015]
                            OLS Regression Results
==============================================================================
Dep. Variable:               new_case   R-squared:                       0.000
Model:                            OLS   Adj. R-squared:                  0.000
Method:                 Least Squares   F-statistic:                     14.06
Date:                Tue, 01 Dec 2020   Prob (F-statistic):           7.85e-07
Time:                        09:42:07   Log-Likelihood:            -2.7868e+06
No. Observations:              303000   AIC:                         5.574e+06
Df Residuals:                  302997   BIC:                         5.574e+06
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const       6.804e+06   1.57e+06      4.330      0.000    3.72e+06    9.88e+06
time          -9.2202      2.130     -4.329      0.000     -13.395      -5.045
compound     -29.5321     10.491     -2.815      0.005     -50.094      -8.971
==============================================================================
Omnibus:                   170031.248   Durbin-Watson:                   0.300
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1665529.545
Skew:                           2.563   Prob(JB):                         0.00
Kurtosis:                      13.279   Cond. No.                     2.67e+11
==============================================================================
```

above regression:new_case~time+compound



above map:#maskup compound sentiment



above map:#travel compound sentiment

Citations

Beri, Aditya. “SENTIMENTAL ANALYSIS USING VADER.” *Medium*, Towards Data Science, 27 May 2020, towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664.

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.