GROUP 4, PROJECT 1

Team Members

Luis Bejaran

Mike Dunlap

Maria Sierra Lizarazo

Rainer Perry

Erin Urban

Agenda

- Overview - Maria
- Analysis review
  - *Erin*
  - *Rainer*
  - *Maria*
  - *Mike*
  - *Luis*

- Post-Mortem
- Q&A

Note: Each speaker will draw conclusions on their analysis

Motivation and Summary

- Hypotheses
  - *Population mobility varies in the US according to the state and the distance*
  - *Travel in the US declined during Covid.*
- Questions
  - *Comparison of daily travel distances by state*
  - *Correlation between distance and number of trips*
  - *Travel data during COVID-19 analysis*
  - *Comparison of people staying at home vs. people travel in the state of NY*
  - *Comparison between weekdays and weekends*

Data Exploration *and* Cleanup

- Data source
  - *US Department of Transportation, Bureau of Transportation Statistics*
    - https://data.bts.gov/
    - Dataframe shape: (32028, 17)
    - Date range: Jan 1, 2019- Sept 19, 2020
- Cleaning
  - *Converted format to float for all numeric columns*
  - *Converted 'date' column to usable data by 'apply(pd.to_datetime)"*
    - "2020-09-19T00:00:00.000" to "2020-09-19"
- Important concepts
  - *Trip: movement that include staying longer than 10 minutes away from home.*
  - *Multiple stays away from home are considered multiple trips*
  - *Transportation methods: driving, rail, transit, and air*
  - *Experimental dataset*

Data Exploration *and* Cleanup

API request for getting the information

```
client = Socrata("data.bts.gov", None)

# Example authenticated client (needed for non-public datasets):
client = Socrata("data.bts.gov",
                 app_token,
                 username= username ,
                 password=password)

# First 2000 results, returned as JSON from API / converted to Python list of
# dictionaries by sodapy.
results = client.get("w96p-f2qv", level= "State", limit=33000)

#JSON data
print(json.dumps(results,indent = 4, sort_keys=True))
```

	level	date	state_fips	state_code	pop_stay_at_home	pop_not_stay_at_home
0	State	2019-01-01T00:00:00.000	01	AL	1028578	3844356

trips	trips_1	trips_1_3	trips_3_5	trips_5_10	trips_10_25	trips_25_50	trips_50_100	trips_100_250	trips_250_500	trips_500
11968328	2792091	3151108	1604875	1879113	1629105	534229	232409	108187	27164	10047

Comparison of Travel Distance by State

- Calculated “avg trip” as total distance traveled divided by number of trips for each day.
- Pulled Summary Statistics
  - Mean = 9.14
  - SEM = 0.01
  - Upper bound: 12.68; Lower bound: 5.36

In [88]:

```
# Calculate summary stats for daily "avg_trip" distance
max_avg_trip = max(analysis_df["avg_trip"])
min_avg_trip = min(analysis_df["avg_trip"])
mean_avg_trip = analysis_df["avg_trip"].mean ()
median_avg_trip = analysis_df["avg_trip"].median ()
var_avg_trip = analysis_df["avg_trip"].var ()
stdev_avg_trip = analysis_df["avg_trip"].std ()
sem_avg_trip = analysis_df["avg_trip"].sem ()

#f"{value:,.2f}"
print(f"The max avg_trip is: {max_avg_trip:,.2f}")
print(f"The min avg_trip is: {min_avg_trip:,.2f}")
print(f"The mean avg_trip is: {mean_avg_trip:,.2f}")
print(f"The median avg_trip is: {median_avg_trip:,.2f}")
print(f"The var avg_trip is: {var_avg_trip:,.2f}")
print(f"The std avg_trip is: {stdev_avg_trip:,.2f}")
print(f"The sem avg_trip is: {sem_avg_trip:,.2f}")


```

```
The max avg_trip is: 28.19
The min avg_trip is: 4.24
The mean avg_trip is: 9.14
The median avg_trip is: 8.94
The var avg_trip is: 2.78
The std avg_trip is: 1.67
The sem avg_trip is: 0.01
```

In [90]:

```
# Calculate quartiles and IQR for daily avg_trip distance
quartiles = analysis_df["avg_trip"].quantile([.25,.5,.75])
lowerq = quartiles[0.25]
upperq = quartiles[0.75]
iqr = upperq-lowerq


print(f"The lower quartile of daily average trip is: {lowerq:,.2f}")
print(f"The upper quartile of daily average trip is: {upperq:,.2f}")
print(f"The interquartile range of daily average trip is: {iqr:,.2f}")
print(f"The the median of daily average trip is: {quartiles[0.5]:,.2f} ")

lower_bound = lowerq - (1.5*iqr)
upper_bound = upperq + (1.5*iqr)
print(f"Values below {lower_bound:,.2f} could be outliers.")
print(f"Values above {upper_bound:,.2f} could be outliers.")
```

```
The lower quartile of daily average trip is: 8.11
The upper quartile of daily average trip is: 9.94
The interquartile range of daily average trip is: 1.83
The the median of daily average trip is: 8.94
Values below 5.36 could be outliers.
Values above 12.68 could be outliers.
```

Comparison of Travel Distance by State

- Identified count of outliers above upper band: 1,104
- Identified count of outliers below lower band: 194
- Removed outliers and created a clean dataframe

In [65]:
```
# Identify outliers above upper bound
analysis_df[analysis_df["avg_trip"] > upper_bound].count()
```

Out[65]:
```
state_code         1104
tot_pop            1104
trips              1104
daily_distance     1104
trips_per_pop      1104
dist_per_pop       1104
avg_trip           1104
date               1104
dtype: int64
```

In [66]:
```
# Identify outliers below lower bound
analysis_df[analysis_df["avg_trip"] < lower_bound].count()
```

Out[66]:
```
state_code         194
tot_pop            194
trips              194
daily_distance     194
trips_per_pop      194
dist_per_pop       194
avg_trip           194
date               194
dtype: int64
```

In [67]:
```
# Create a df to eliminate outliers
clean_analysis_df = analysis_df
```

In [68]:
```
# Drop outliers above upper bound
clean_analysis_df = clean_analysis_df[analysis_df["avg_trip"] > lower_bound]
clean_analysis_df.dropna()
```

Out[68]:

	state_code	tot_pop	trips	daily_distance	trips_per_pop	dist_per_pop	avg_trip	date
15709	AK	735182.0	3714276.0	27174836.0	5.052186	36.963413	7.316321	2019-08-21
3141	AK	737438.0	1627288.0	17696513.0	2.206678	23.997289	10.874850	2020-07-19
6631	AK	735182.0	2300248.0	19676042.5	3.128814	26.763499	8.553879	2019-02-24
31570	AK	737438.0	2125612.0	18146743.0	2.882428	24.607822	8.537185	2020-06-27
4948	AK	735182.0	2733503.0	22585225.5	3.718131	30.720591	8.262375	2019-01-22
...	...	...	...	...	...	...	...	...
20807	WY	575970.0	2523476.0	25309725.0	4.381263	43.942783	10.029707	2019-11-28
20756	WY	575970.0	2869675.0	28977103.0	4.982334	50.310091	10.097695	2019-11-27
20705	WY	575970.0	2901268.0	24883650.5	5.037186	43.203032	8.576819	2019-11-26
21062	WY	575970.0	3118757.0	27950144.0	5.414791	48.527083	8.961950	2019-12-03
32027	WY	577737.0	1748958.0	23318694.5	3.027256	40.362128	13.332907	2020-07-05

31834 rows × 8 columns

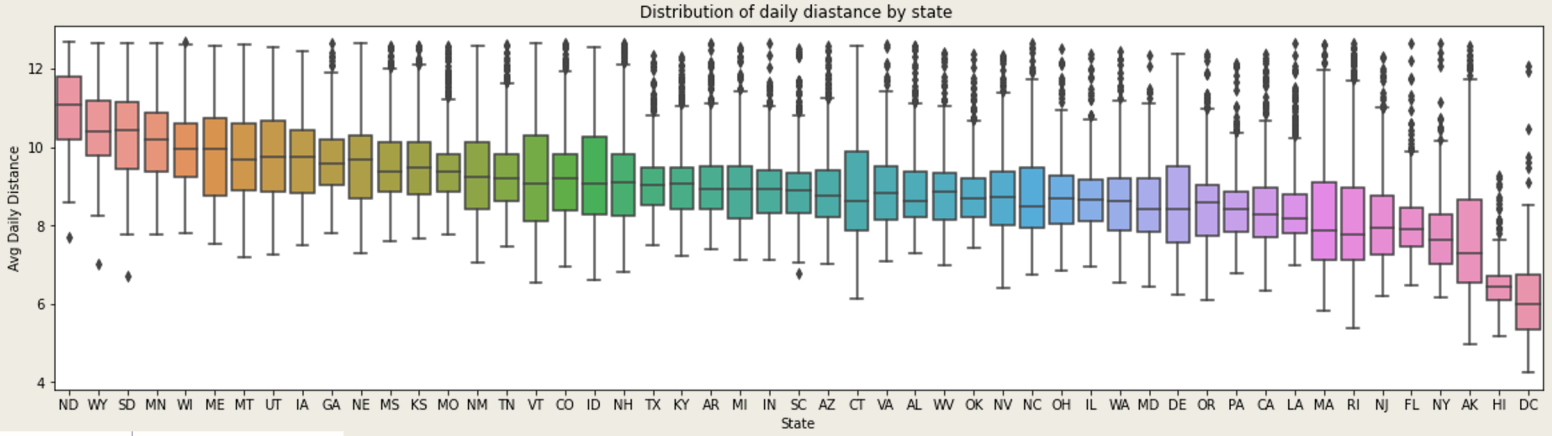
In [69]:
```
# Drop outliers below lower bound
clean_analysis_df = analysis_df[analysis_df["avg_trip"] < upper_bound]
clean_analysis_df.dropna()
```

Comparison of Travel Distance by State

Displayed box plots of average daily trip distance grouped by state.
Longest 5 states: ND, WY, SD, MN WI Shortest 5 states: DC, HI, AK, NY, FL

In [99]:
```
#Diplay box plots of daily average trip distance by state on cleaned data
plt.figure(figsize=(20,5))
ranks = clean_analysis_df.groupby("state_code")["avg_trip"].mean().fillna(0).sort_values()[::-1].index
sns.boxplot(y='avg_trip', x='state_code',
                 data=clean_analysis_df,
                order = ranks)

plt.xlabel("State")
plt.ylabel("Avg Daily Distance")
plt.title("Distribution of daily diastance by state ")
```

Out[99]: Text(0.5, 1.0, 'Distribution of daily diastance by state ')



Data Analysis –Distance vs # of Trips

Many people take trips daily - but how does distance affect the number of trips taken?

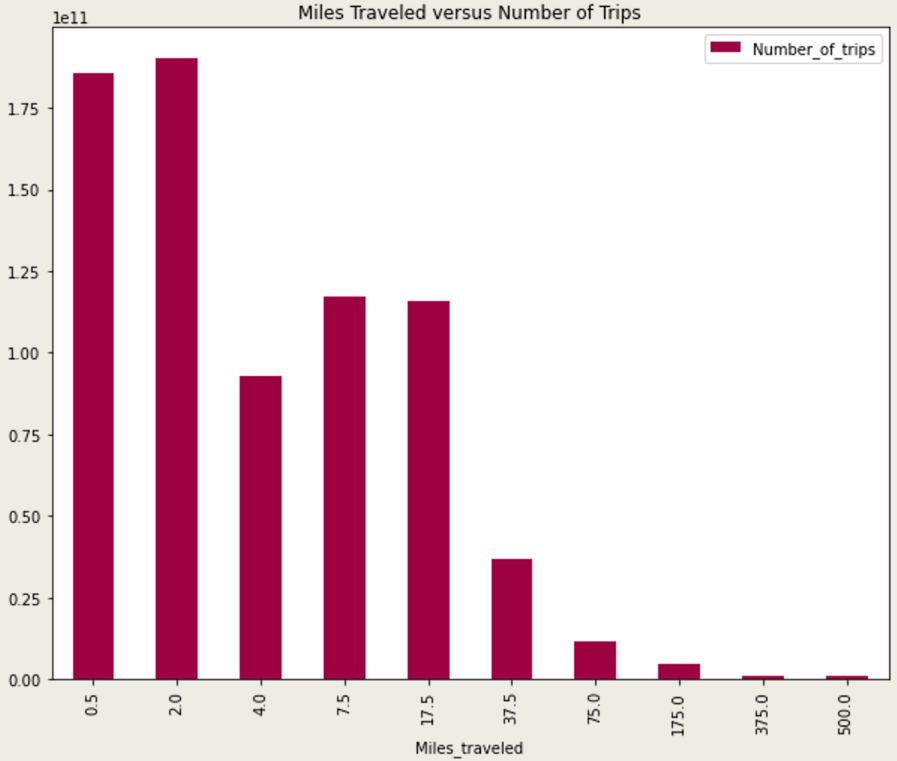
```
# Get sums for each column
sums = []
for column in trips_only.columns:
    sums.append(trips_only[column].sum())
print(sums)
```

```
[763830810713.0, 187295665612.0, 191861510451.0, 93639109511.0, 118371486095.0, 116688453115.0, 37323955281.0, 116788
45468.0, 4792113413.0, 1119380357.0, 1060291410.0, 205096143442.0, 120418.54316018664, 93647832806.0, 383723020902.0,
374556438044.0, 887786145712.5, 2042047929512.5, 1399648323037.5, 875913410100.0, 838619847275.0, 419767633875.0, 795
218557500.0, 6711280815727.0, 1088672.624236992, 292647.9737116295]
```

```
# trips_1 = 0.5
# trips_1_3  = 2
# trips_3_5 = 4
# trips_5_10 = 7.5
# trips_10_25 = 17.5
# trips_25_50  = 37.5
# trips_50_100  = 75
# trips_100_250 = 175
# trips_250_500 = 375
# trips_500 = 500

corr_data = pd.DataFrame({"Miles_traveled": [0.5, 2, 4, 7.5, 17.5, 37.5, 75, 175, 375, 500],
                          "Number_of_trips": [185754409397.0, 190217502447.0, 92827307781.0,
                                              117332036135.0, 115638845508.0, 36940290862.0, 11531199023.0,
                                              4723303135.0, 1103282749.0, 1049803846.0]})

corr_data = corr_data.sort_values(['Miles_traveled'], ascending = True)
# Plotting

#corr_data.plot(kind='bar',x='Number_of_trips',y='Miles_traveled',title='None',legend=True)
corr_data.plot(kind='bar',x='Miles_traveled',y='Number_of_trips',
               title='Miles Traveled versus Number of Trips',legend=True, figsize=(10,8),
              style='dict', colormap ='Spectral')
plt.savefig('Images/correlation_trips_distance.png')
corr_data
```

Data Analysis –Distance vs # of Trips

	Miles_traveled	Number_of_trips
0	0.5	1.857544e+11
1	2.0	1.902175e+11
2	4.0	9.282731e+10
3	7.5	1.173320e+11
4	17.5	1.156388e+11
5	37.5	3.694029e+10
6	75.0	1.153120e+10
7	175.0	4.723303e+09
8	375.0	1.103283e+09
9	500.0	1.049804e+09



Travel data during COVID-19 analysis

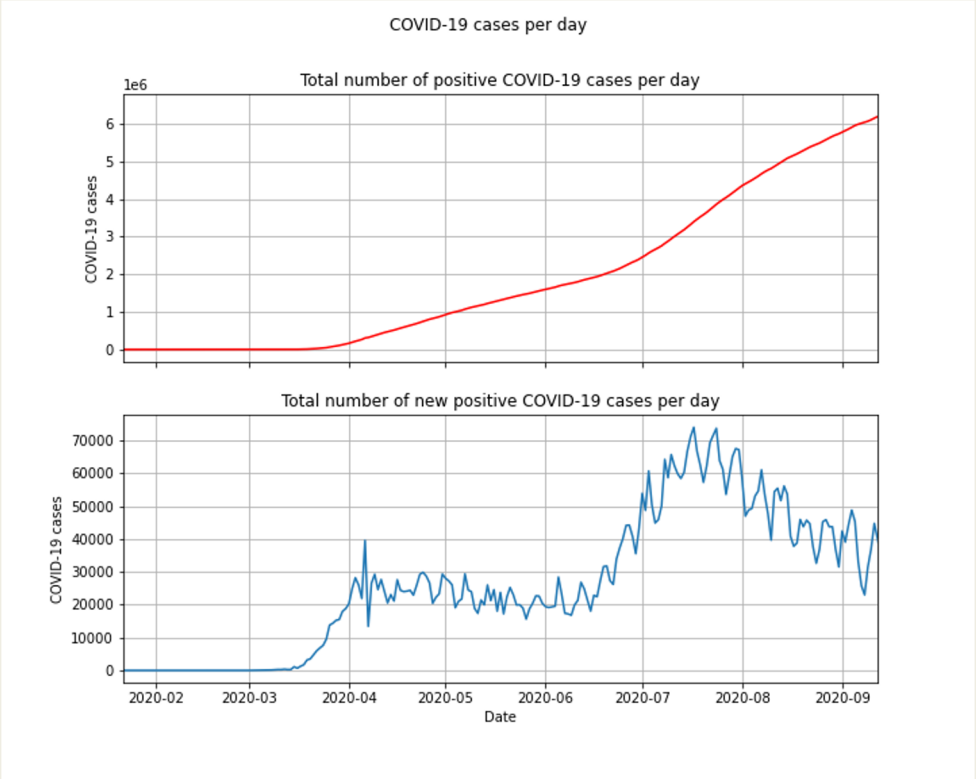
Data source: Centers for Disease Control and Prevention (CDC) - US Department of Health and Human Services.

- https://healthdata.gov/
- Dataframe shape: (14880, 15) - CSV reading file

Cleaning:

- Converted format to float for all numeric columns
- Converted 'date' column to usable data by "apply(pd.to_datetime)"

	date	state_code	tot_cases	new_case
0	2020-01-22	CO	0	0
1	2020-01-23	CO	0	0
2	2020-01-24	CO	0	0



Travel data during COVID-19 analysis

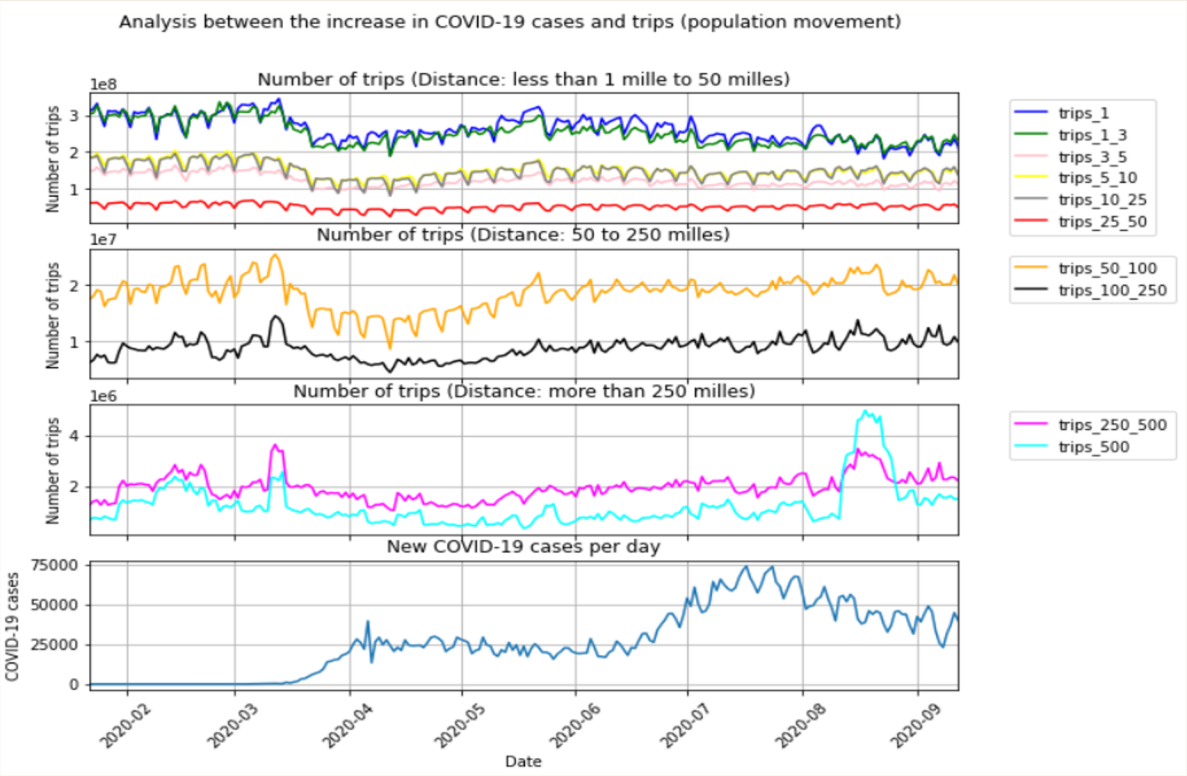
Has the amount of trips decreased during the pandemic?

```
# Merge the COVID-19 data with the data related with trips
results_copy_df = pd.merge(results_copy_df,covid_df, on=['date','state_code'])
```

```
pos_cases_trips = results_copy_df.groupby(['date'],as_index=False).agg({'trips':'sum','trips_1':'sum',
                                                                        'trips_1_3':'sum','trips_3_5':'sum',
                                                                        'trips_5_10':'sum','trips_10_25':'sum',
                                                                        'trips_50_100':'sum','trips_100_250':'sum',
                                                                        'trips_250_500':'sum','trips_500':'sum',
                                                                        'new_case':'sum', 'trips_25_50':'sum' })
```

Travel data during COVID-19 analysis

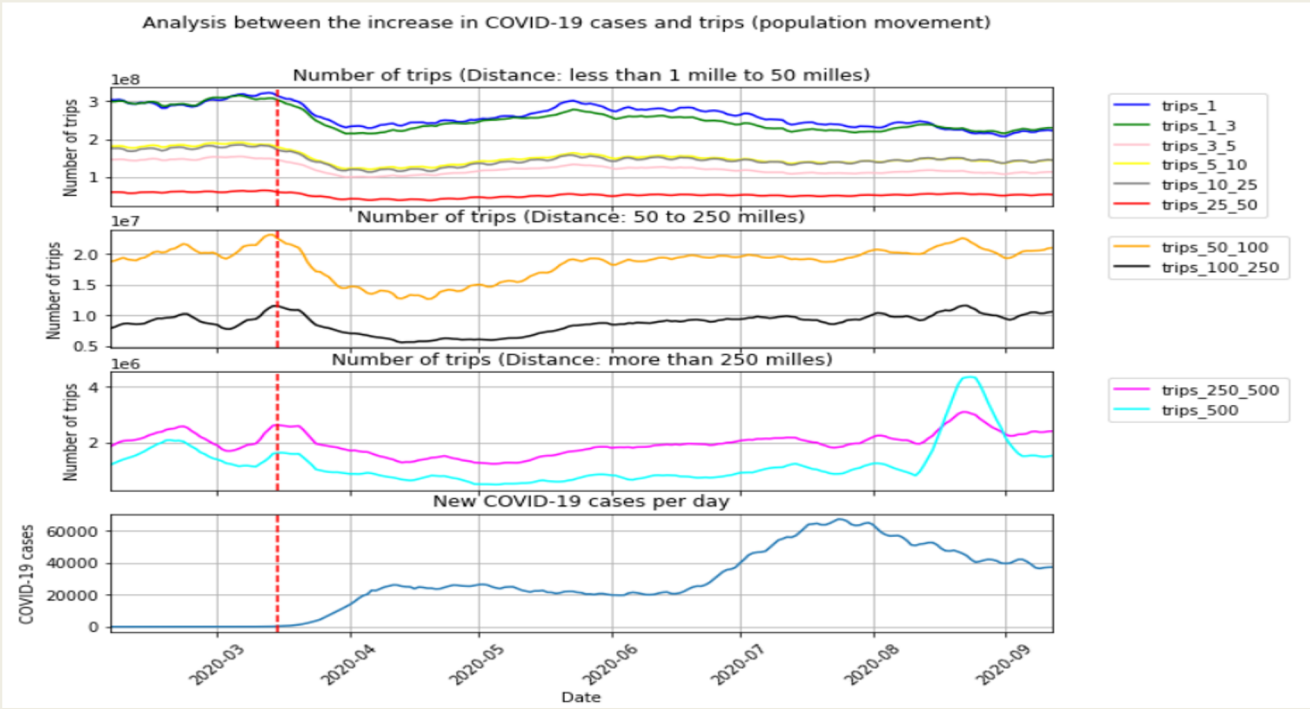
Has the amount of trips decrease during the pandemic?



Travel data during COVID-19 analysis

Has the amount of trips decrease during the pandemic?

```
# Rolling window (10 datapoints) for getting the mean each 10 points in the data
pos_cases_trips2['trips_1'] = pos_cases_trips2['trips_1'].rolling(window=10).mean()
```



Weekend versus Weekday travel behavior

New Dataframe -

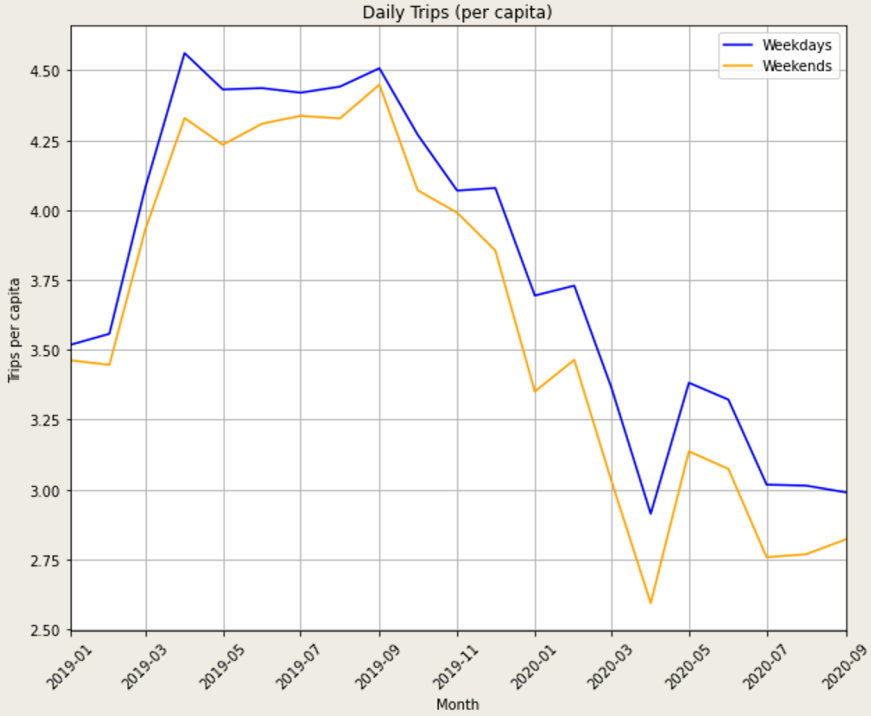
- Identifying “day of the week” based on date
- Labeling each entry “weekday” or “weekend”

```
# WEEKDAYS AND WEEKENDS COMPARISON

copy_2_df = pd.DataFrame()
copy_2_df = results_df

# Creating columns for 1) day of the week, 2) weekend/weekday indicator, 3)
copy_2_df['day'] = copy_2_df['date'].dt.day_name()
copy_2_df['weekend'] = ""
total_pop = []
trips_per_capita = []
for i, row in copy_2_df.iterrows():
    total_pop.append(row.pop_stay_at_home + row.pop_not_stay_at_home)
    day = row['day']
    if day in ["Saturday", "Sunday"]:
        copy_2_df.loc[i,'weekend'] = 'Weekend'
    else:
        copy_2_df.loc[i,'weekend'] = 'Weekday'
copy_2_df['total_pop'] = total_pop

for i, row in copy_2_df.iterrows():
    trips_per_capita.append(row.trips / row.total_pop)
copy_2_df['trips_per_capita'] = trips_per_capita

day = copy_2_df['day']
weekend = copy_2_df['weekend']

copy_2_df.drop(labels=['day'], axis=1,inplace = True)
copy_2_df.insert(2, 'day', day)
copy_2_df.drop(labels=['weekend'], axis=1,inplace = True)
copy_2_df.insert(3, 'weekend', weekend)
copy_2_df.drop(labels=['total_pop'], axis=1,inplace = True)
copy_2_df.insert(9, 'total_pop', total_pop)
copy_2_df.drop(labels=['trips_per_capita'], axis=1,inplace = True)
copy_2_df.insert(10, 'trips_per_capita', trips_per_capita)
```

Weekend versus Weekday travel behavior

Key Findings:

- Weekend trips are fewer than weekday trips
- COVID caused a sharp drop-off in travel and slightly expanded the gap between weekend and weekday trips taken in Summer 2020 vs. Summer 2019



Weekend versus Weekday travel behavior

New Dataframe -

- Identifying “day of the week” based on date
- Labeling each entry “weekday” or “weekend”

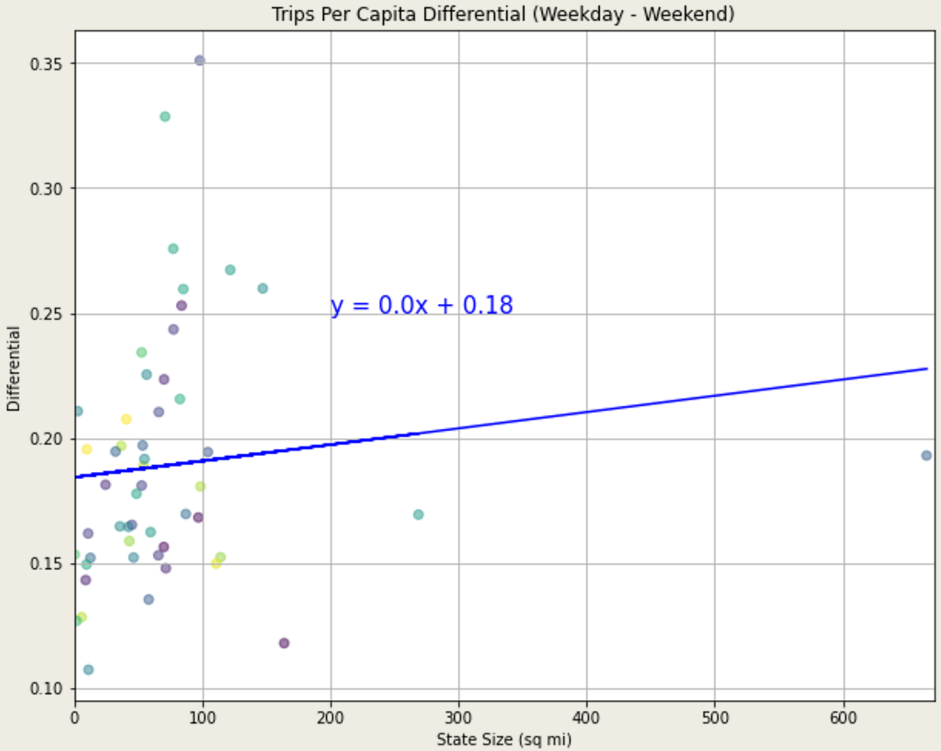
In [31]:

```
# Creating dataframe of state sizes and weekday/weekend per capita differentials
results_df_state = results_df.groupby(['state_code', 'weekend'])
per_capita_states = results_df_state['trips_per_capita'].mean()
states_per_capita = pd.DataFrame({'Trips Per Capita': per_capita_states})
states_per_capita = states_per_capita.reset_index()
states_per_capita_weekend = states_per_capita.loc[states_per_capita['weekend'] == 'Weekend']
weekend_per_cap = states_per_capita_weekend['Trips Per Capita']
weekend_per_cap = list(weekend_per_cap)
weekend_per_cap

states_per_capita_weekday = states_per_capita.loc[states_per_capita['weekend'] == 'Weekday']
weekday_per_cap = states_per_capita_weekday['Trips Per Capita']
weekday_per_cap = list(weekday_per_cap)
weekday_per_cap

list_states = states_per_capita_weekend['state_code']
list_states = list(list_states)

# State sizes (square miles, in thousands), alphabetical by state code, including "DC"
state_size = [665.4, 52.4, 53.2, 114, 163.7, 104.1, 5.5, 0.1, 2.5, 65.8, 59.4, 10.9, 56.3, 83.6,
              57.9, 36.4, 82.3, 40.4, 52.4, 10.6, 12.4, 35.4, 96.7, 86.9, 69.7, 48.4, 147, 53.8, 70.7,
              77.3,  9.3, 110.6, 121.6, 8.7, 54.6, 44.8, 69.9, 98.4,46.1, 1.5, 32, 77.1,
              42.1, 268.6, 84.9,  42.8, 9.6, 71.3, 65.5, 24.2, 97.8]


state_diffs = pd.DataFrame({"State": list_states,
                            "Size": state_size,
                            "Weekday": weekday_per_cap,
                            "Weekend": weekend_per_cap,
                            "Trips Differential": ""})


for i, row in state_diffs.iterrows():
    weekday = state_diffs.loc[i,'Weekday']
    weekend = state_diffs.loc[i,'Weekend']
    state_diffs.loc[i,'Trips Differential'] = (weekday - weekend)

state_diffs.sort_values(by="State",ascending=True)
```

Weekend versus Weekday travel behavior by State Size

Question: Does state size have an impact on differential in weekend vs. weekday trip totals?

- Finding: There is no significant correlation (r-squared: 0.015)



Comparison of NY State population that did not stay at home vs who did stay at home

Data Analysis -

```
In [ ]:
1  # Convert to pandas DataFrame
2  results_df = pd.DataFrame.from_records(results)
3  print(f'The data size is: {results_df.shape}')
4  results_df = results_df.sort_values(['date','state_fips'])
5  results_df = results_df.reset_index(drop=True)
6  results_df
```

```
The data size is: (32028, 17)
```

Out[3]:

	level	date	state_fips	state_code	pop_stay_at_home	pop_not_stay_at_home	trips	trips_1	trips_1_3	trips_3_5	trips_5_10	trips_
0	State	2019-01-01T00:00:00.000	01	AL	1028578	3844356	11968328	2792091	3151108	1604875	1879113	16
1	State	2019-01-01T00:00:00.000	02	AK	200344	534838	1942538	558829	529269	242666	264106	2
2	State	2019-01-01T00:00:00.000	04	AZ	1721026	5428700	18705619	5351386	4462153	2087404	2659479	27
3	State	2019-01-01T00:00:00.000	05	AR	642665	2361951	7488494	1715475	2086803	964511	1162837	9
4	State	2019-01-01T00:00:00.000	06	CA	9212440	30223696	111648618	33567702	28725797	12723636	14685031	133
...	...	...	...	...	...	...	...	...	...	...	...	
32023	State	2020-09-19T00:00:00.000	51	VA	2166787.0	6350898.0	27570481.0	6257753.0	6728144.0	3494711.0	4385998.0	4166
32024	State	2020-09-19T00:00:00.000	53	WA	2130975.0	5404616.0	20203302.0	4798678.0	5120927.0	2386824.0	3093663.0	3082
32025	State	2020-09-19T00:00:00.000	54	WV	472635.0	1333197.0	5481803.0	1155200.0	1360656.0	688792.0	864758.0	843
32026	State	2020-09-19T00:00:00.000	55	WI	1489727.0	4323841.0	18251715.0	4098483.0	4330159.0	2182163.0	2820040.0	2848
32027	State	2020-09-19T00:00:00.000	56	WY	148734.0	429003.0	1948852.0	507499.0	535028.0	239039.0	188756.0	180

32028 rows × 17 columns

Comparison of NY State population that did not stay at home vs who did stay at home

Data Analysis -

```
# Create a new df and only call NY "state"
# date range from jan 1, 2019 to sept 9, 2020
ny_df = results_df.loc[results_df['state_code'] == "NY",["date", 'state_code',
                                                         "pop_not_stay_at_home" , "pop_stay_at_home"]]
ny_df.sort_values(['date'], ascending=True, inplace=True)
ny_df.set_index('date', inplace=True)
ny_df
```

date	state_code	pop_not_stay_at_home	pop_stay_at_home
2019-01-01	NY	13550603.0	5931873.0
2019-01-02	NY	14860900.0	4621576.0
2019-01-03	NY	14768458.0	4714018.0
2019-01-04	NY	14813149.0	4669327.0
2019-01-05	NY	14527397.0	4955079.0
...	...	...	...
2020-09-15	NY	13097591.0	6444618.0
2020-09-16	NY	13084394.0	6457815.0
2020-09-17	NY	13174722.0	6367487.0
2020-09-18	NY	13267226.0	6274983.0
2020-09-19	NY	13016840.0	6525369.0

628 rows × 3 columns

New Dataframe - ny_df

- **Columns**
  - *Date*
  - *State code*
  - *Pop not stay at home*
  - *Pop stay at home*

Comparison of NY State population that did not stay at home vs who did stay at home

Data Analysis - Line Graph showing the comparison of population movement

```
In [26]:
# Create a line graph to show the comparison in population
multi_plot = ny_df.plot(kind="line", figsize=(20,7))

plt.title("NYS Pop. of people who did not stay home vs. who did stay home")
plt.xlabel('Date')
plt.ylabel('NYS Population in the Millions')
plt.grid()
plt.savefig('Images/population_NY.png')

plt.show()
```

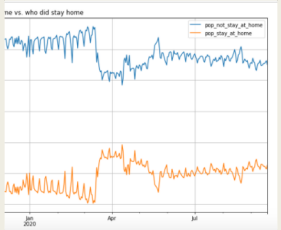
Blue line - pop. not stay at home
Orange line - pop. stay at home
X axis - Date
Y axis - NYS in the millions

Comparison of NY State population that did not stay at home vs who did stay at home

Data Analysis - **Conclusions**

- ➔ From data we can see that population movement was consistent from Jan 1, 2019 – March 2020.
- ➔ There is a sharp decline in the population not staying at home which correlates with the sharp increase of people staying at home in March 2020 (COVID quarantine begins in NY).

- ➔ The numbers then remain steady over the first month and then we see a peak in the population not staying home in late April-May and a second peak in late May/early June.



.

Comparison of NY State population that did not stay at home vs who did stay at home

Data Analysis - **Conclusions**

➔ These peaks coincide with the Black Lives Matter protests and phase 1 NYS re-opening.

➔ There has been a steady increase in the population not staying at home since June but the overall number of people that remains at home is steady. Likely due to work from home arrangements, flexible schedules, virtual schooling, social distancing regulations, etc.

Post Mortem

What difficulties did we face?

- Lack of experience with github
- Integrating our separate work within the repository
- Time, time, time . . . .we could all use more time.

Additional questions or opportunities

- We broke tasks up in the beginning and each worked on specific questions and collaborated to compile all our work in the end
  - *Now seeing the individual analyses - we could further enhance our individual work based on the work of others.*
  - *Example: Incorporating the 'state size" data into the travel distance by state analysis*

Questions?

Appendix

Appendix – Skills Applied

- Unit 6
  - *API key for data retrieval*
  - *Dataframe created and saved as .csv*
- Unit 5
  - *Summary statistics: mean, median, variance, standard deviation or SEM*
  - *Quartiles, IQR and Outliers*
  - *Plots created: scatterplot, line, bar, pie, box plot and saved as png*
  - *Linear regression and correlation*
- Unit 4

  Data manipulation using:
    - *Groupby*
    - *Merge dataframes*
    - *Drop duplicates*
    - *Identify unique*

Appendix – Skills Applied

- Python dependencies used:
  - `import matplotlib.pyplot as plt`
  - `import requests`
  - `from scipy import stats`
  - `from scipy.stats import linregress`
  - `import pandas as pd`
  - `import numpy as np`
  - `import json`
  - `import datetime as dt`
  - `import seaborn as sns`
  - `import matplotlib.dates as mdates`
  - `# - Install sodapy (pip install sodapy)`
  - `# - In the config.py file write your username, password, and app_token`
  - `from sodapy import Socrata`
  - `from config import app_token`
  - `from config import username`
  - `from config import password`