Effect of Covid-19 on New York City Transportation (March 2020 – June 2020)

By: Michael Reichard, Yeonjae Riyoon, Joshua Sohan, Anumala Thapa

Table of Contents

Ridership

Taxi & For-Hire Vehicles

Citibike

Subway

Questions

Questions

- Did Citibike usage increase/decrease due to the several phases in NYC?
- What was the transportation usage between March 2019 to June 2019?
- What was the transportation usage between March 2020 to June 2020?
- What means of transportation was used the most and the least between March 2019 to June 2019?
- What means of transportation was used the most and the least between March 2020 to June 2020?
- Which borough had the most and the least usage of transportation during March 2020 to June 2020?

Taxi & For-Hire Vehicles Usage

Taxi and For-hire Vehicles:

Data Sources:

- https://github.com/nychealth/coronavirus-data
- https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

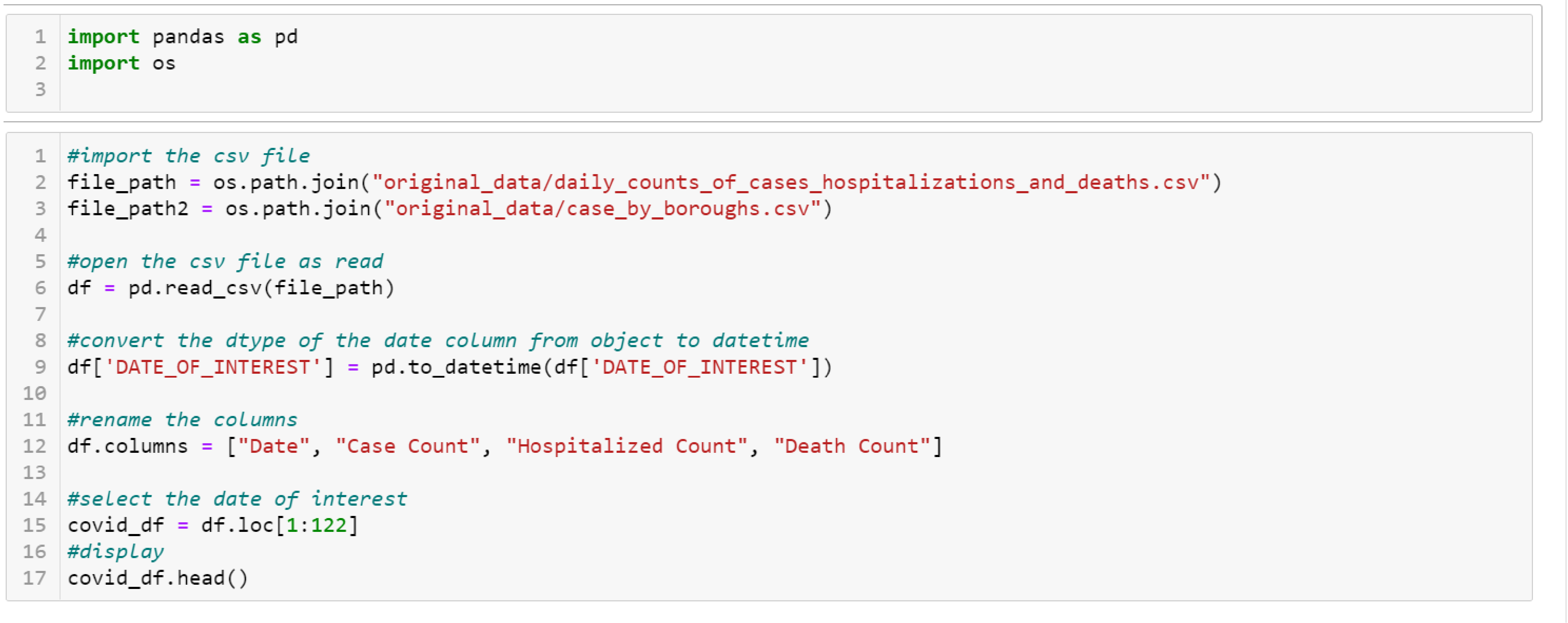
Focal Point:

- Case count from coronavirus data
- Pick-up datetime from trip record data

Consideration:

- Yellow Taxi, Green Taxi and For-hire Vehicles(i.e community based liveries, black cars and luxury limousines)

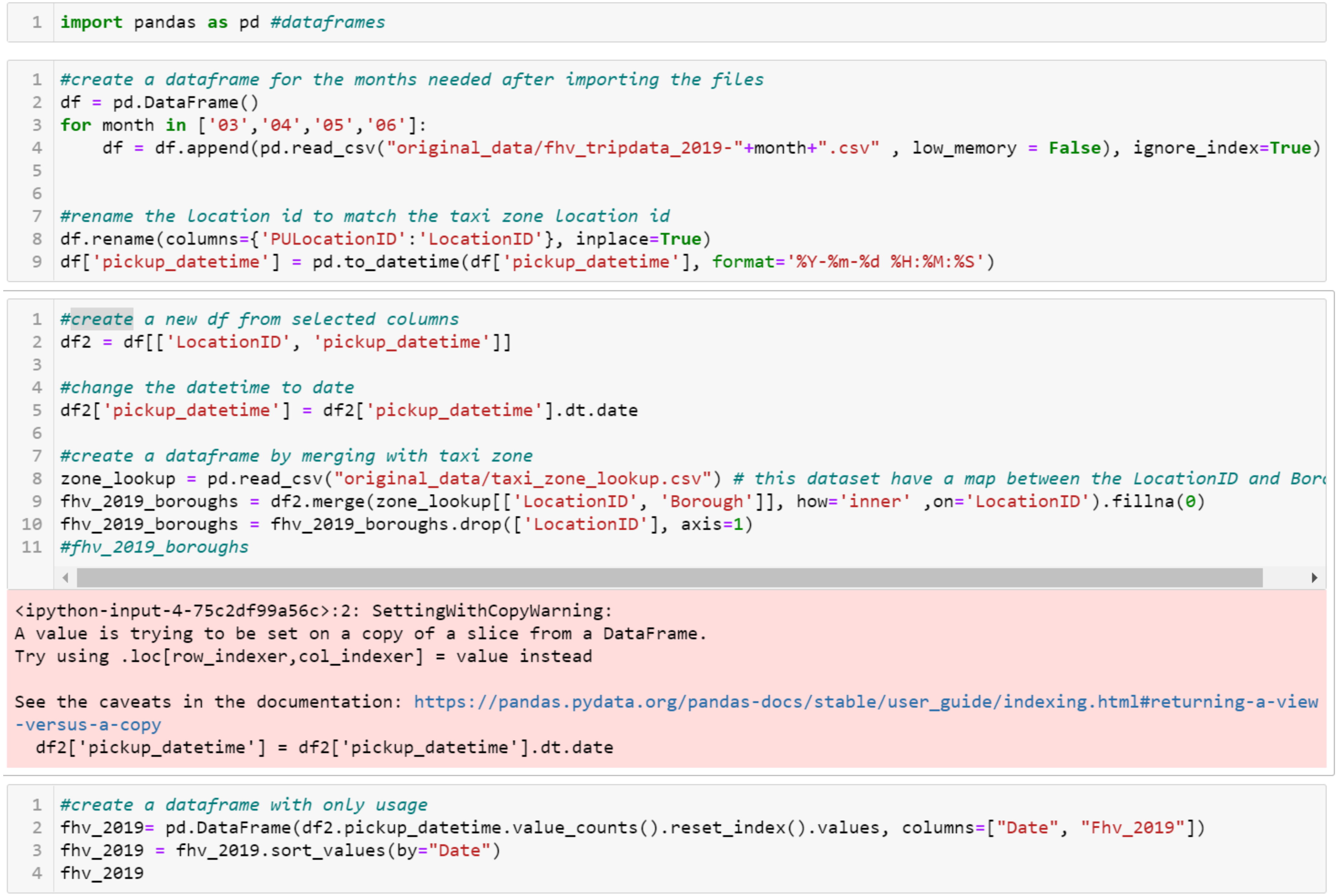
Covid-19 Data



```
import pandas as pd
import os

```

```
#import the csv file
file_path = os.path.join("original_data/daily_counts_of_cases_hospitalizations_and_deaths.csv")
file_path2 = os.path.join("original_data/case_by_boroughs.csv")

#open the csv file as read
df = pd.read_csv(file_path)

#convert the dtype of the date column from object to datetime
df['DATE_OF_INTEREST'] = pd.to_datetime(df['DATE_OF_INTEREST'])

#rename the columns
df.columns = ["Date", "Case Count", "Hospitalized Count", "Death Count"]

#select the date of interest
covid_df = df.loc[1:122]
#display
covid_df.head()
```

Taxi & FHV: Cleaning the Data



```
import pandas as pd #dataframes
```

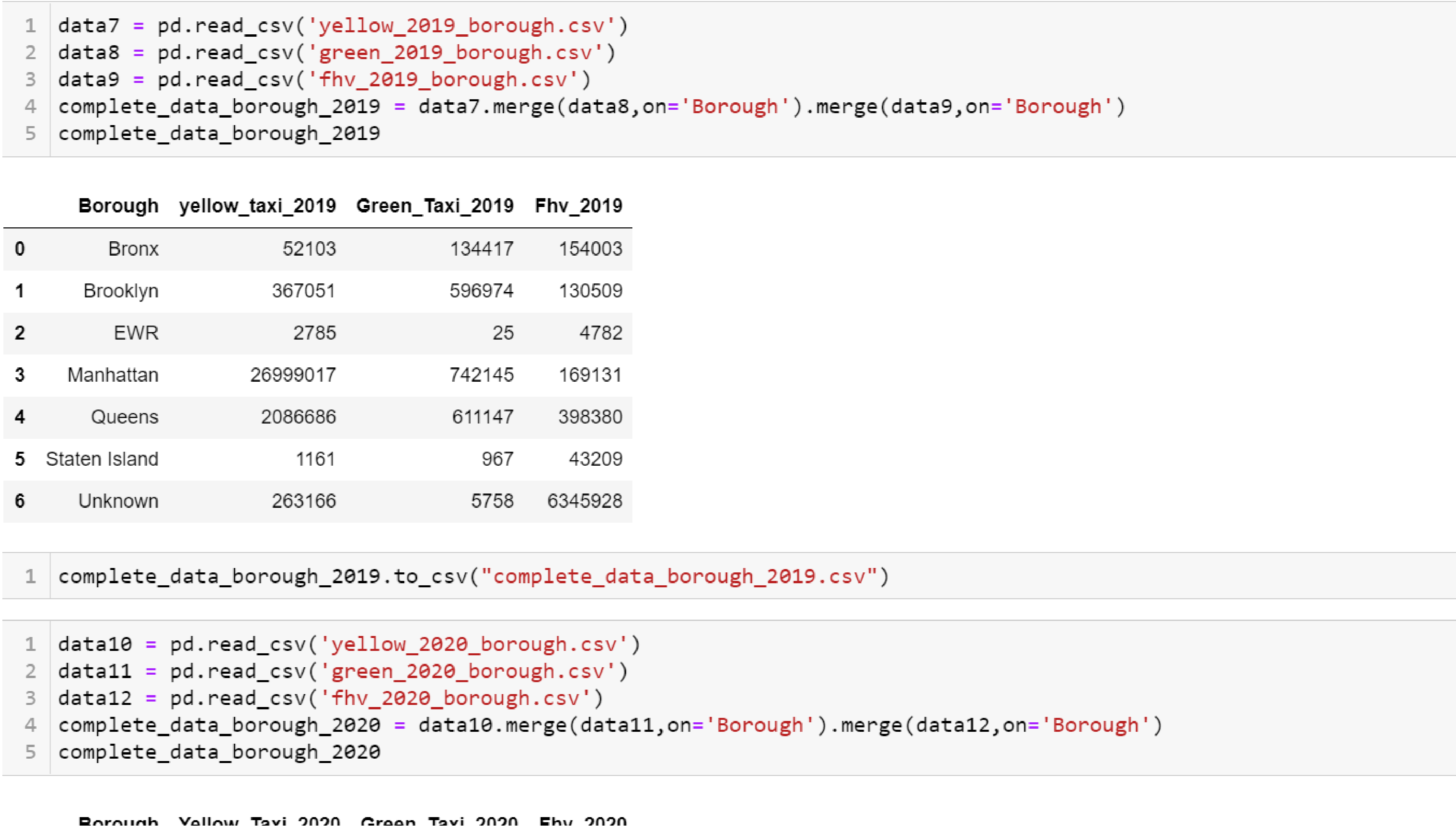
```
#create a dataframe for the months needed after importing the files
df = pd.DataFrame()
for month in ['03','04','05','06']:
    df = df.append(pd.read_csv("original_data/fhv_tripdata_2019-"+month+".csv" , low_memory = False), ignore_index=True)


#rename the location id to match the taxi zone location id
df.rename(columns={'PULocationID':'LocationID'}, inplace=True)
df['pickup_datetime'] = pd.to_datetime(df['pickup_datetime'], format='%Y-%m-%d %H:%M:%S')
```

```
#create a new df from selected columns
df2 = df[['LocationID', 'pickup_datetime']]

#change the datetime to date
df2['pickup_datetime'] = df2['pickup_datetime'].dt.date

#create a dataframe by merging with taxi zone
zone_lookup = pd.read_csv("original_data/taxi_zone_lookup.csv") # this dataset have a map between the LocationID and Bor
fhv_2019_boroughs = df2.merge(zone_lookup[['LocationID', 'Borough']], how='inner' ,on='LocationID').fillna(0)
fhv_2019_boroughs = fhv_2019_boroughs.drop(['LocationID'], axis=1)
#fhv_2019_boroughs
```

```
<ipython-input-4-75c2df99a56c>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view
-versus-a-copy
  df2['pickup_datetime'] = df2['pickup_datetime'].dt.date
```

```
#create a dataframe with only usage
fhv_2019= pd.DataFrame(df2.pickup_datetime.value_counts().reset_index().values, columns=["Date", "Fhv_2019"])
fhv_2019 = fhv_2019.sort_values(by="Date")
fhv_2019
```

Taxi & FHV: Cleaning the Data



```python
data7 = pd.read_csv('yellow_2019_borough.csv')
data8 = pd.read_csv('green_2019_borough.csv')
data9 = pd.read_csv('fhv_2019_borough.csv')
complete_data_borough_2019 = data7.merge(data8,on='Borough').merge(data9,on='Borough')
complete_data_borough_2019
```

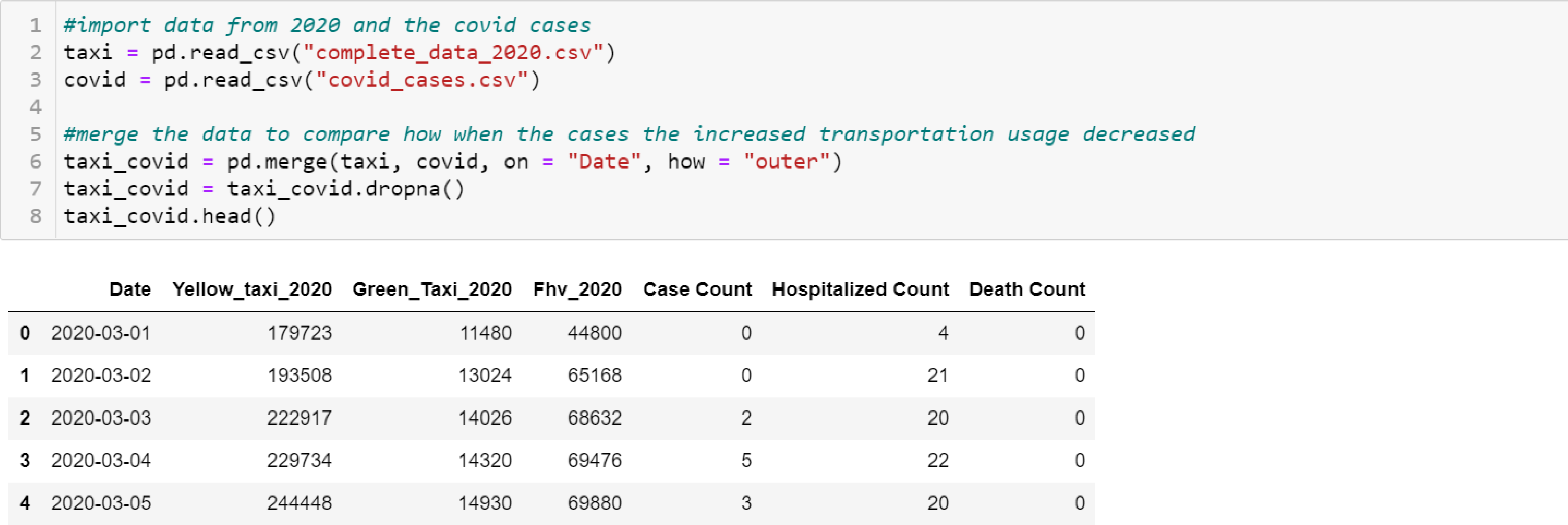
	Borough	yellow_taxi_2019	Green_Taxi_2019	Fhv_2019
0	Bronx	52103	134417	154003
1	Brooklyn	367051	596974	130509
2	EWR	2785	25	4782
3	Manhattan	26999017	742145	169131
4	Queens	2086686	611147	398380
5	Staten Island	1161	967	43209
6	Unknown	263166	5758	6345928

```python
complete_data_borough_2019.to_csv("complete_data_borough_2019.csv")
```

```python
data10 = pd.read_csv('yellow_2020_borough.csv')
data11 = pd.read_csv('green_2020_borough.csv')
data12 = pd.read_csv('fhv_2020_borough.csv')
complete_data_borough_2020 = data10.merge(data11,on='Borough').merge(data12,on='Borough')
complete_data_borough_2020
```

Borough Yellow_Taxi_2020 Green_Taxi_2020 Fhv_2020

Taxi & FHV: Cleaning the Data



```python
#import data from 2020 and the covid cases
taxi = pd.read_csv("complete_data_2020.csv")
covid = pd.read_csv("covid_cases.csv")

#merge the data to compare how when the cases the increased transportation usage decreased
taxi_covid = pd.merge(taxi, covid, on = "Date", how = "outer")
taxi_covid = taxi_covid.dropna()
taxi_covid.head()
```

	Date	Yellow_taxi_2020	Green_Taxi_2020	Fhv_2020	Case Count	Hospitalized Count	Death Count
0	2020-03-01	179723	11480	44800	0	4	0
1	2020-03-02	193508	13024	65168	0	21	0
2	2020-03-03	222917	14026	68632	2	20	0
3	2020-03-04	229734	14320	69476	5	22	0
4	2020-03-05	244448	14930	69880	3	20	0

Taxi vs Covid:

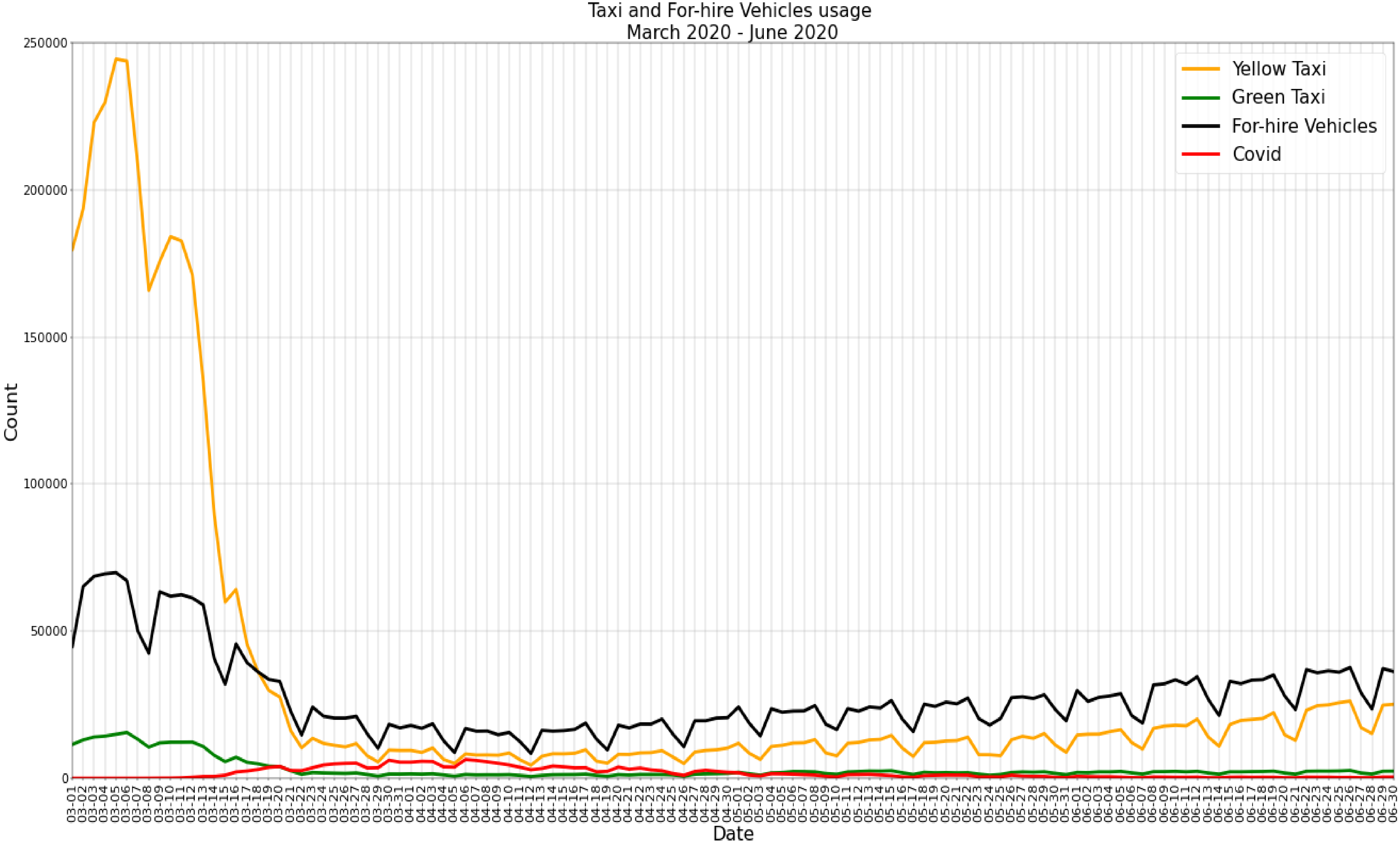
- March 20:

  Lockdown

  Case count: 4007
- June 8:

  Phase 1

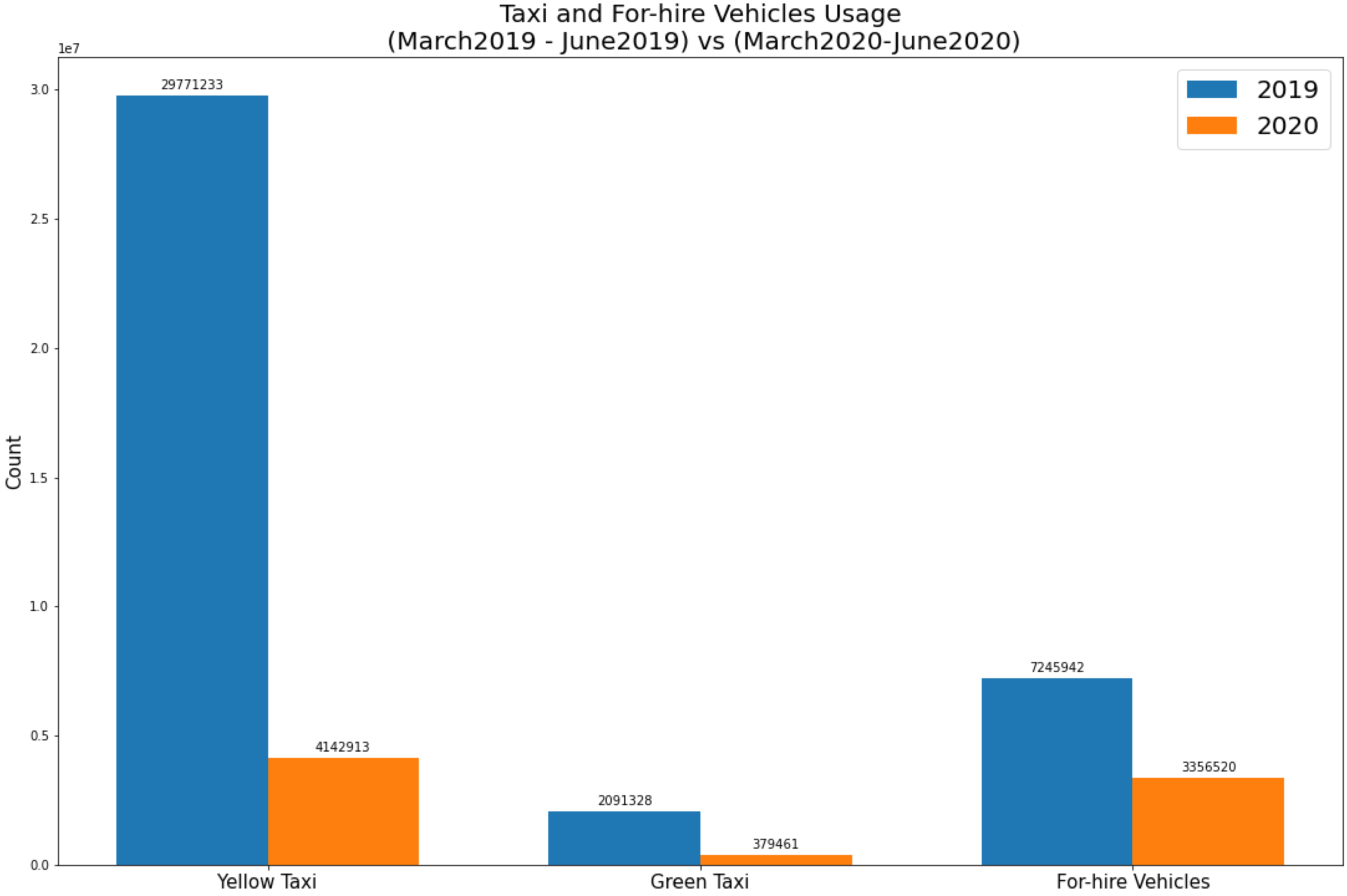
  Case count: 451
- Usage went down during lockdown and started picking up slowly
- Yellow cabs took the hit the most


Taxi and For-hire Vehicles usage
March 2020 - June 2020
Yellow Taxi
Green Taxi
For-hire Vehicles
Covid
Count
250000
200000
150000
100000
50000
0
Date
03-01
03-02
03-03
03-04
03-05
03-06
03-07
03-08
03-09
03-10
03-11
03-12
03-13
03-14
03-15
03-16
03-17
03-18
03-19
03-20
03-21
03-22
03-23
03-24
03-25
03-26
03-27
03-28
03-29
03-30
03-31
04-01
04-02
04-03
04-04
04-05
04-06
04-07
04-08
04-09
04-10
04-11
04-12
04-13
04-14
04-15
04-16
04-17
04-18
04-19
04-20
04-21
04-22
04-23
04-24
04-25
04-26
04-27
04-28
04-29
04-30
05-01
05-02
05-03
05-04
05-05
05-06
05-07
05-08
05-09
05-10
05-11
05-12
05-13
05-14
05-15
05-16
05-17
05-18
05-19
05-20
05-21
05-22
05-23
05-24
05-25
05-26
05-27
05-28
05-29
05-30
05-31
06-01
06-02
06-03
06-04
06-05
06-06
06-07
06-08
06-09
06-10
06-11
06-12
06-13
06-14
06-15
06-16
06-17
06-18
06-19
06-20
06-21
06-22
06-23
06-24
06-25
06-26
06-27
06-28
06-29
06-30

2019 vs 2020:

- Compared to last year of March to June:
- Yellow cabs decreased by almost 86%
- Green cabs decreased by 81%
- For-hire vehicles by 54%

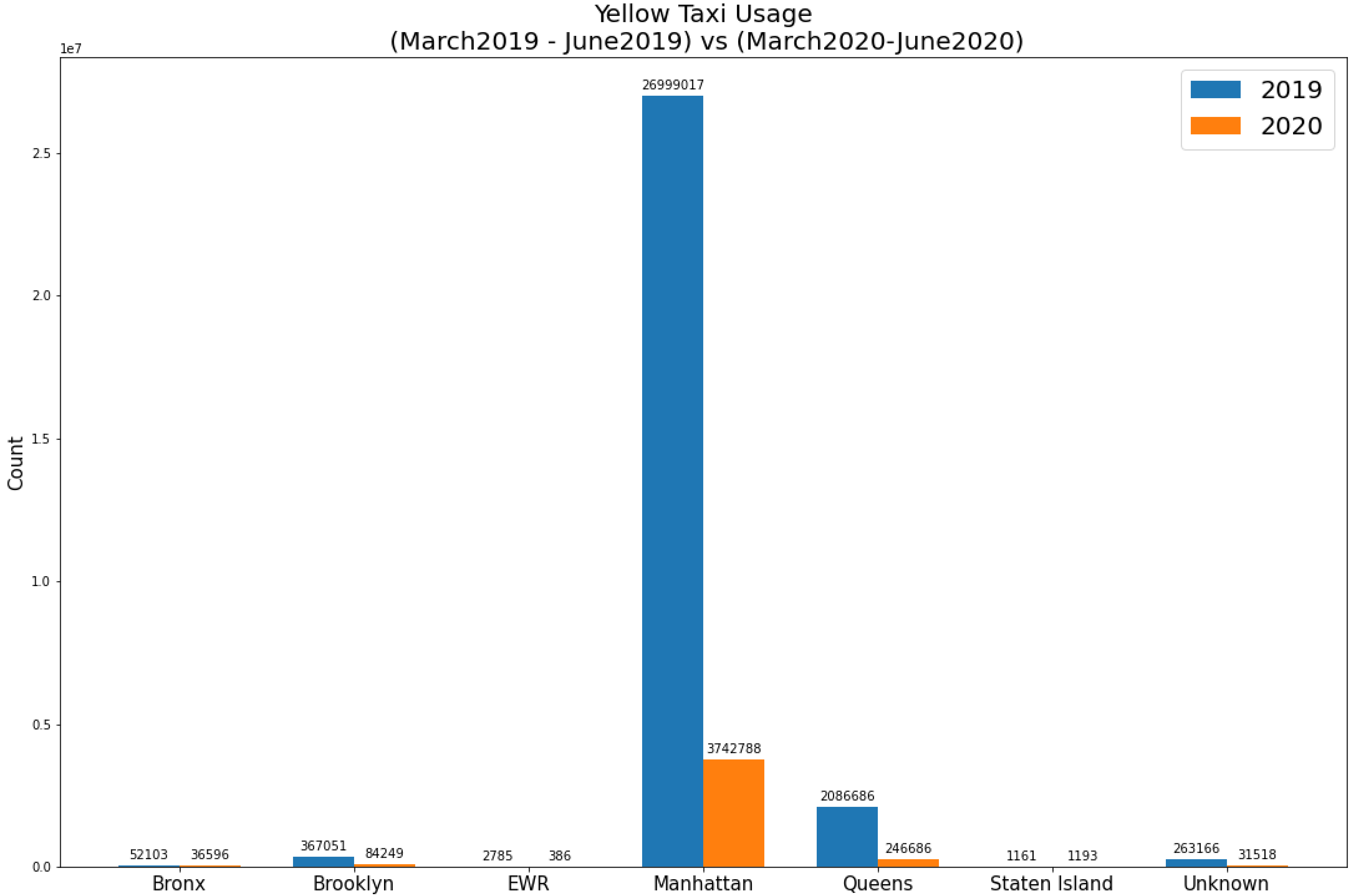
- (fyi: only 1 in 4 yellow cabs are in operation)


Taxi and For-hire Vehicles Usage
(March2019 - June2019) vs (March2020-June2020)
1e7
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Count
2019
2020
29771233
4142913
2091328
379461
7245942
3356520
Yellow Taxi
Green Taxi
For-hire Vehicles

Usage by Borough:

- Focal Point : Yellow Taxi
- Used the most in Manhattan
- 86% decrease in use in Manhattan
- 88% decrease in use in Queens

- (fyi: highest case counts in Queens)


Yellow Taxi Usage
(March2019 - June2019) vs (March2020-June2020)
1e7
2.5
2.0
1.5
1.0
0.5
0.0
Count
2019
2020
52103
36596
367051
84249
2785
386
26999017
3742788
2086686
246686
1161
1193
263166
31518
Bronx
Brooklyn
EWR
Manhattan
Queens
Staten Island
Unknown

Citibike Usage

Citibike Data

Data Source:

- https://s3.amazonaws.com/tripdata/index.html

Focal Point:

- Number of trips by month
  - (March – June 2019) vs (March – June 2020)

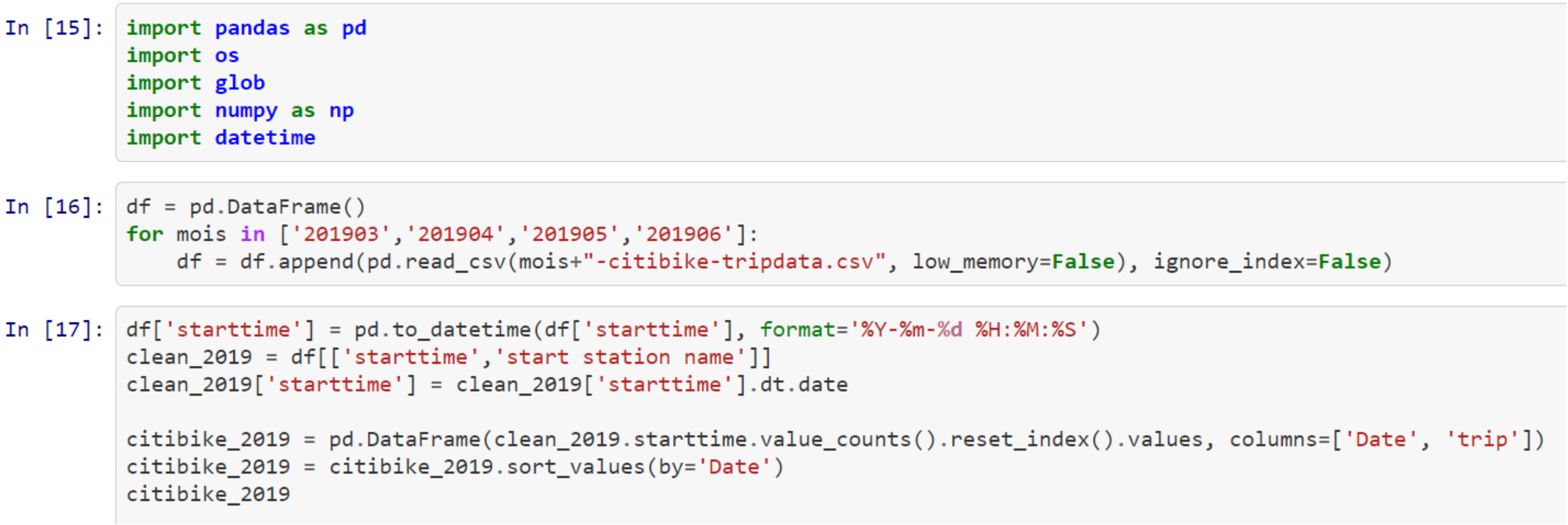
Citibike vs Covid

- We wanted to compare the impact Covid19 had on the usage of Citibikes in NYC
- We compared March – June 2020
- Our Questions for this segment are

  - How did Covid 19/Coronavirus affect the usage of Citibikes?

  -

Citibike Usage <March – June 2019>

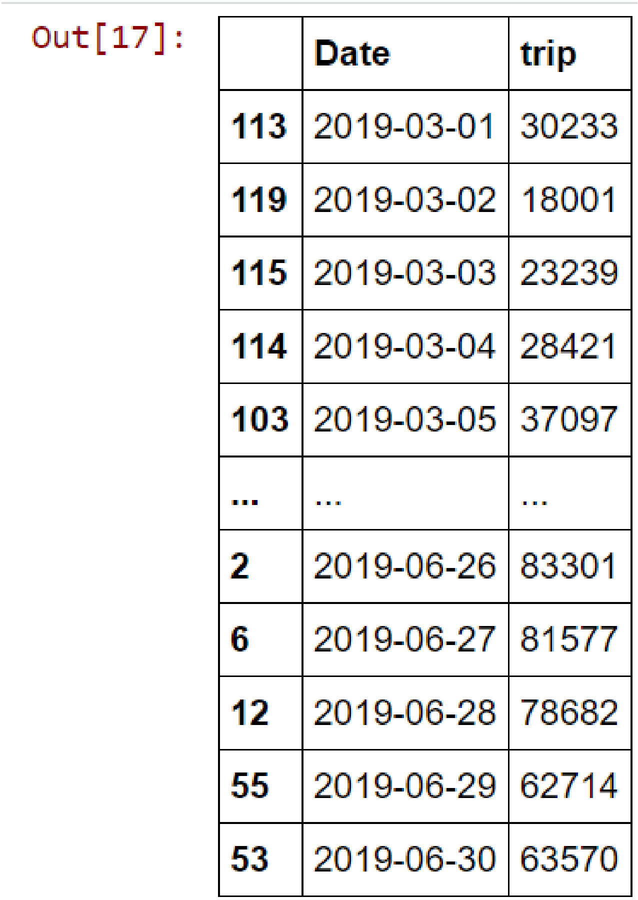


```
In [15]: import pandas as pd
         import os
         import glob
         import numpy as np
         import datetime
```

```
In [16]: df = pd.DataFrame()
         for mois in ['201903','201904','201905','201906']:
             df = df.append(pd.read_csv(mois+"-citibike-tripdata.csv", low_memory=False), ignore_index=False)
```

```
In [17]: df['starttime'] = pd.to_datetime(df['starttime'], format='%Y-%m-%d %H:%M:%S')
         clean_2019 = df[['starttime','start station name']]
         clean_2019['starttime'] = clean_2019['starttime'].dt.date

         citibike_2019 = pd.DataFrame(clean_2019.starttime.value_counts().reset_index().values, columns=['Date', 'trip'])
         citibike_2019 = citibike_2019.sort_values(by='Date')
         citibike_2019
```

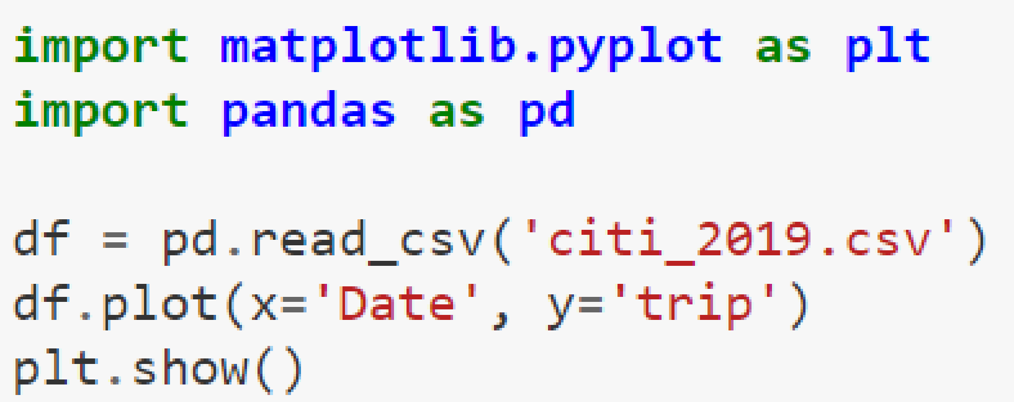
Citibike Usage <March – June 2019>



Out[17]:

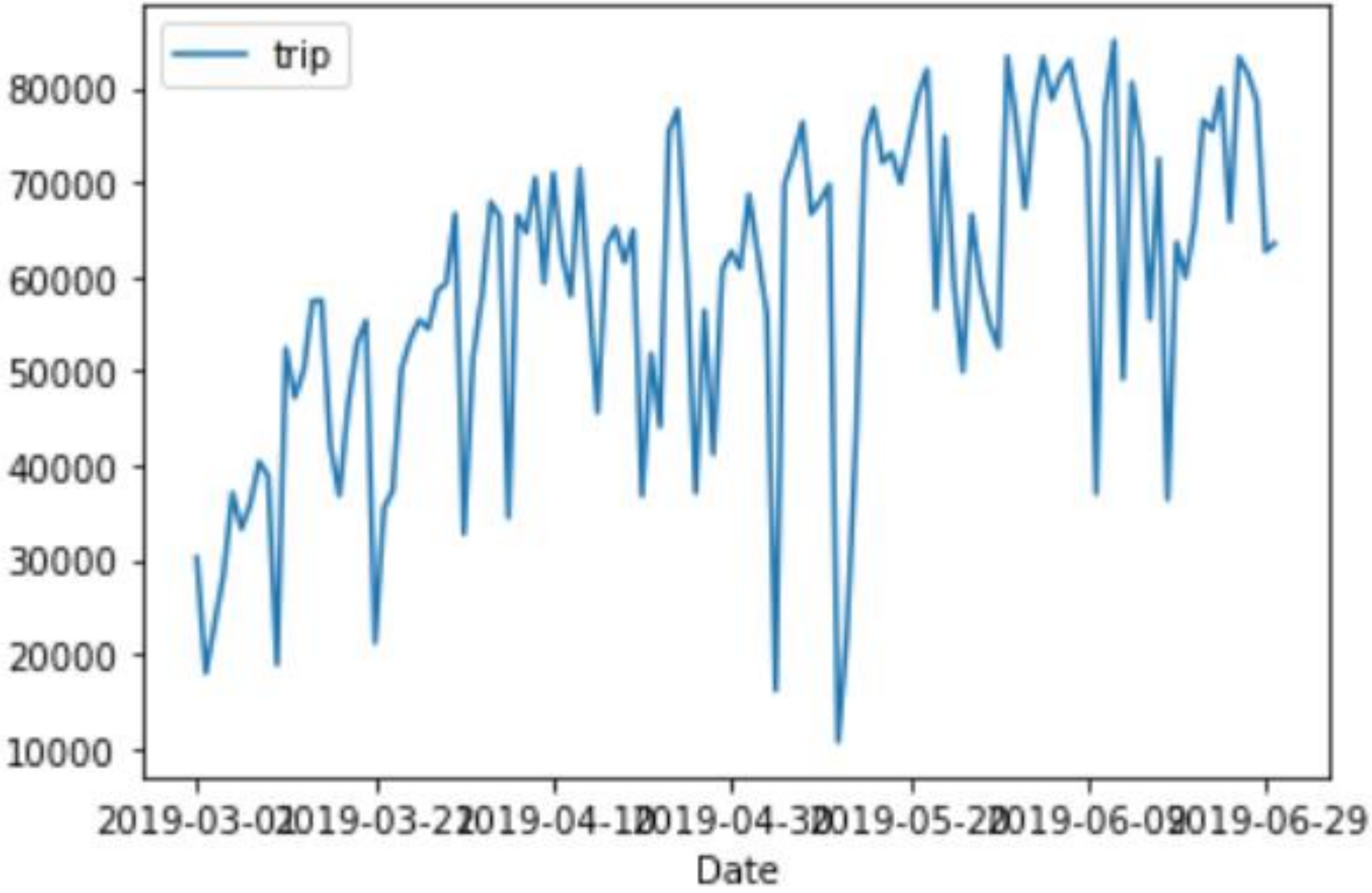
	Date	trip
113	2019-03-01	30233
119	2019-03-02	18001
115	2019-03-03	23239
114	2019-03-04	28421
103	2019-03-05	37097
...	...	...
2	2019-06-26	83301
6	2019-06-27	81577
12	2019-06-28	78682
55	2019-06-29	62714
53	2019-06-30	63570

Citibike Usage Graph <March – June 2019>

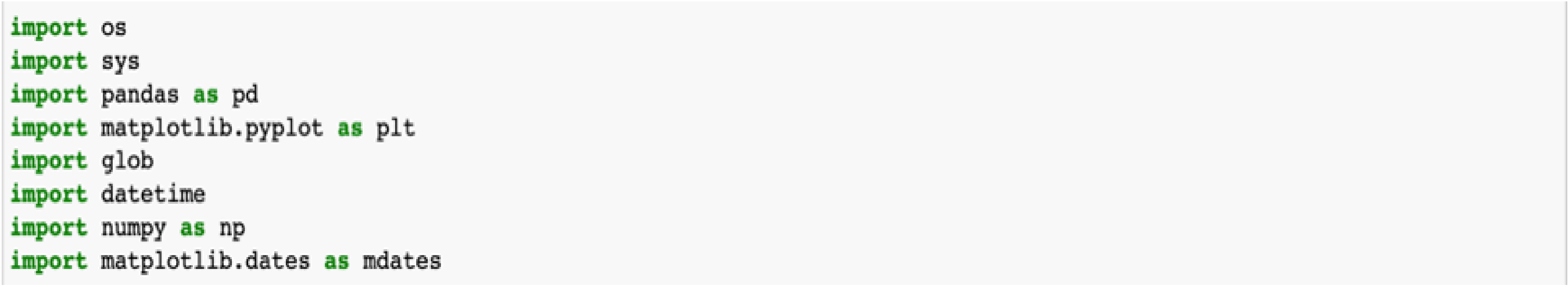


```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv('citi_2019.csv')
df.plot(x='Date', y='trip')
plt.show()
```

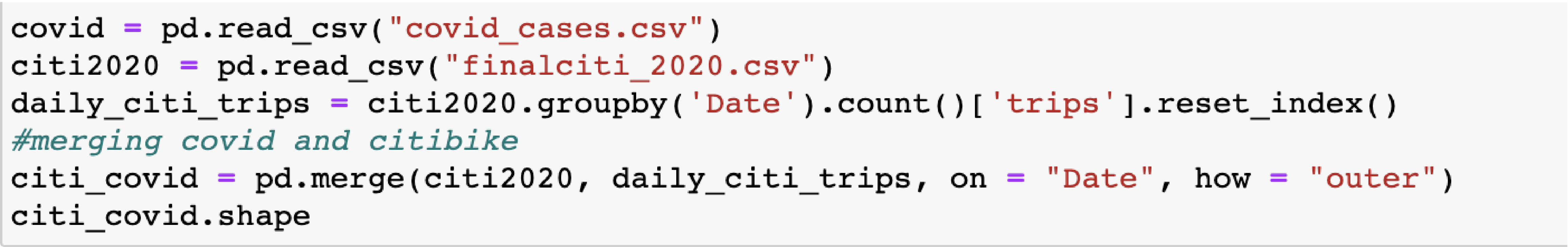
Citibike Usage Graph <March – June 2019>



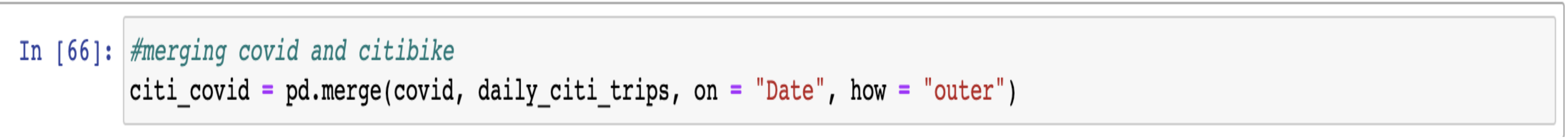
Citibike Usage <March – June 2020>: Cleaning



```
import os
import sys
import pandas as pd
import matplotlib.pyplot as plt
import glob
import datetime
import numpy as np
import matplotlib.dates as mdates
```

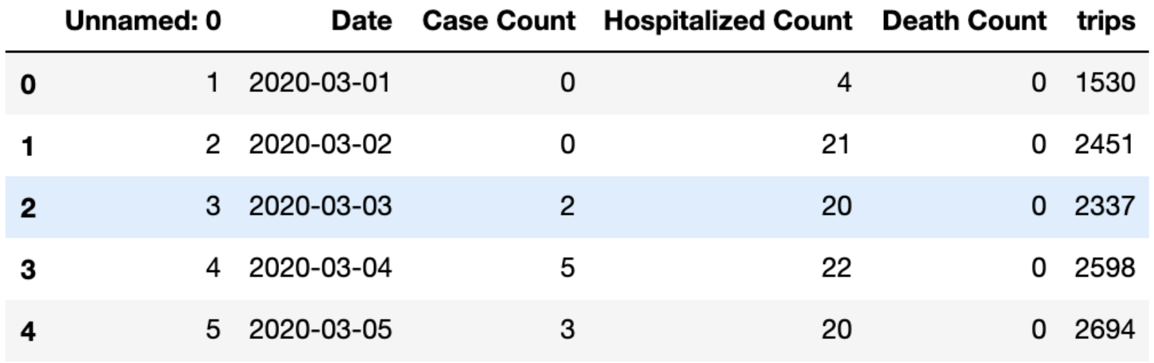


```
covid = pd.read_csv("covid_cases.csv")
citi2020 = pd.read_csv("finalciti_2020.csv")
daily_citi_trips = citi2020.groupby('Date').count()['trips'].reset_index()
#merging covid and citibike
citi_covid = pd.merge(citi2020, daily_citi_trips, on = "Date", how = "outer")
citi_covid.shape
```

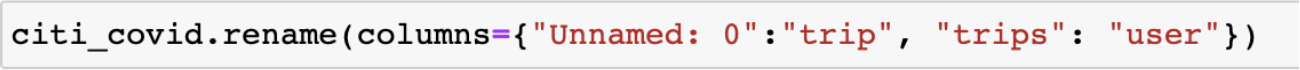


```
In [66]: #merging covid and citibike
         citi_covid = pd.merge(covid, daily_citi_trips, on = "Date", how = "outer")
```

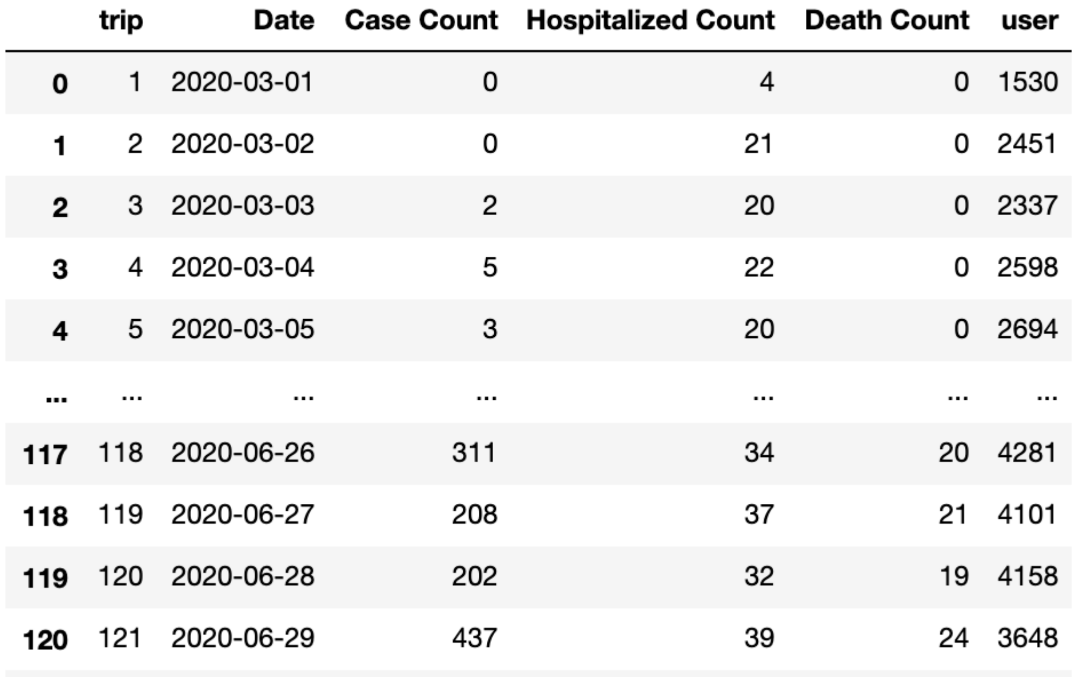
Citibike Usage <March – June 2020>: Cleaning



	Unnamed: 0	Date	Case Count	Hospitalized Count	Death Count	trips
0	1	2020-03-01	0	4	0	1530
1	2	2020-03-02	0	21	0	2451
2	3	2020-03-03	2	20	0	2337
3	4	2020-03-04	5	22	0	2598
4	5	2020-03-05	3	20	0	2694

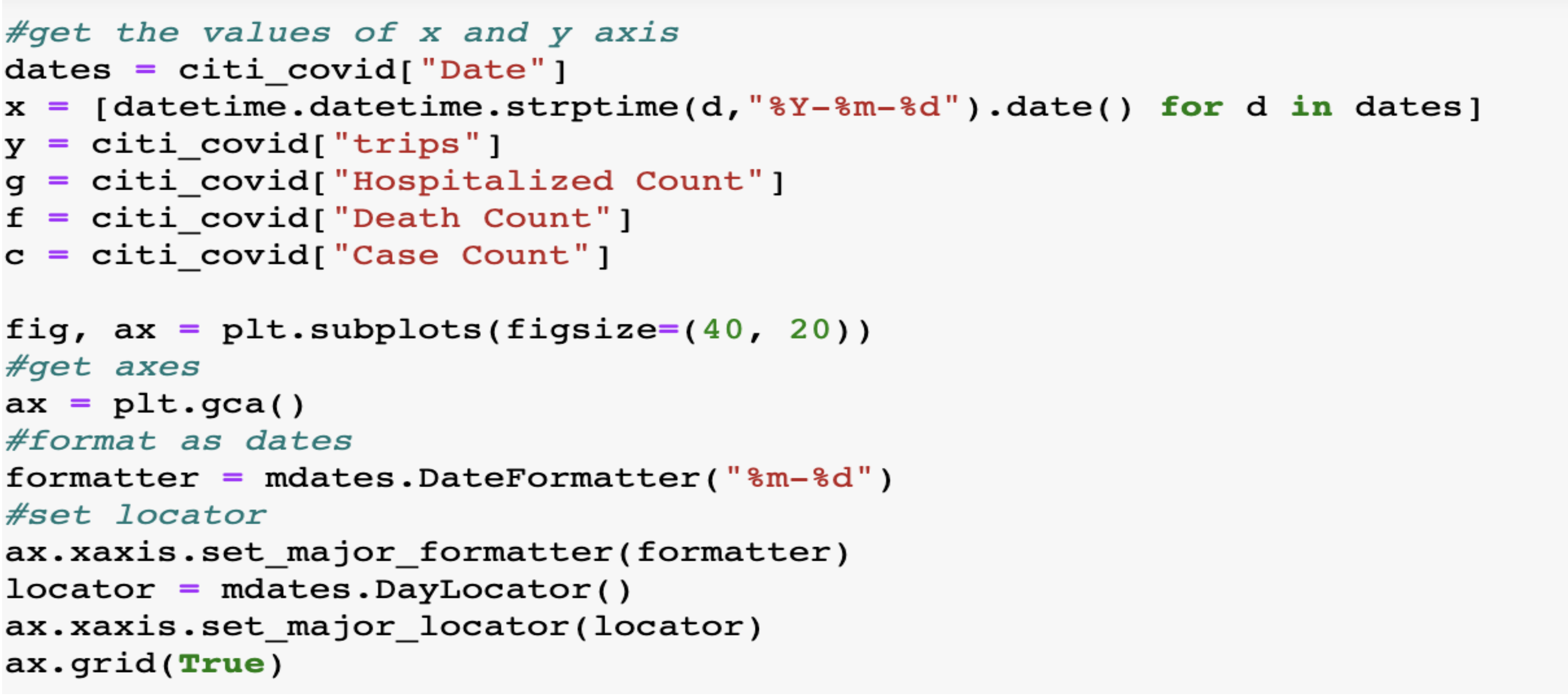


```
citi_covid.rename(columns={"Unnamed: 0":"trip", "trips": "user"})
```



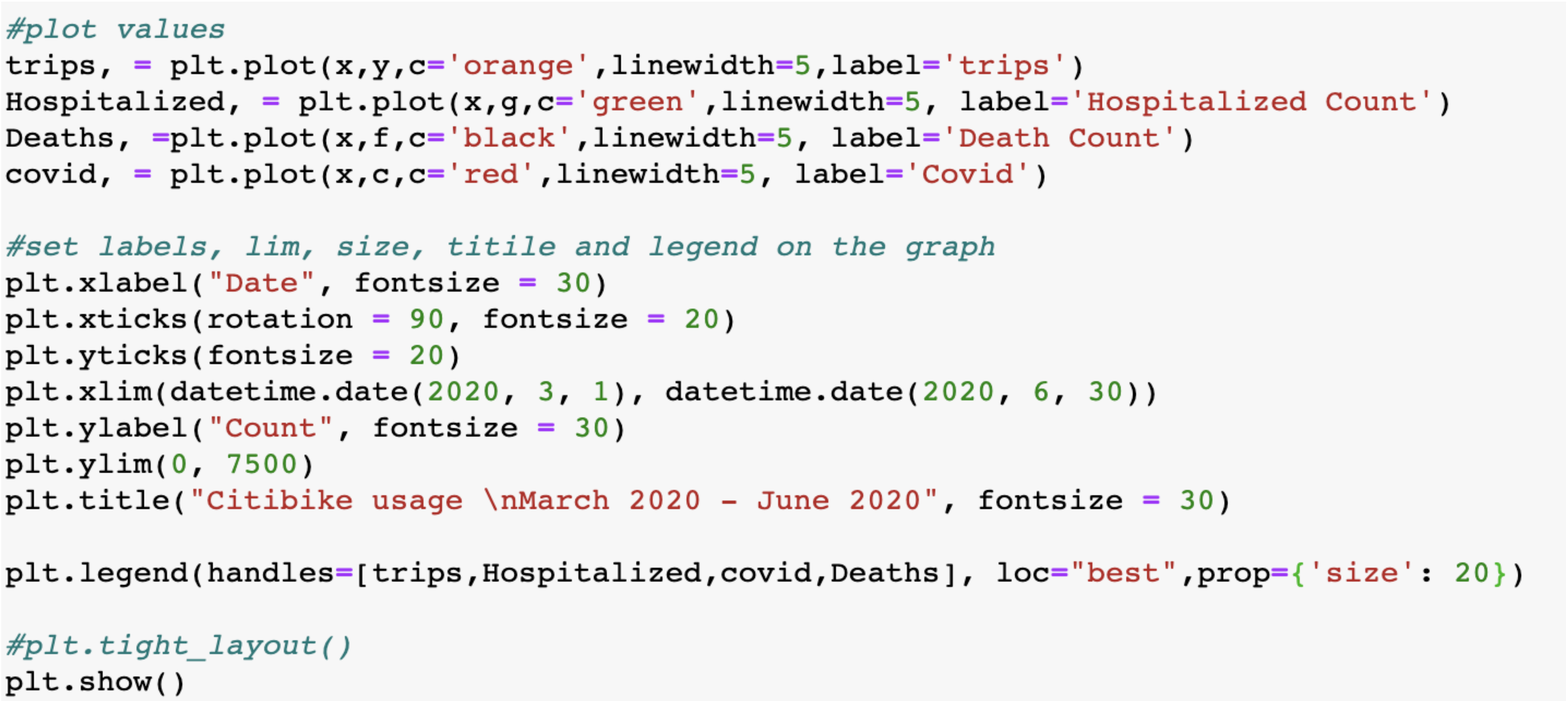
	trip	Date	Case Count	Hospitalized Count	Death Count	user
0	1	2020-03-01	0	4	0	1530
1	2	2020-03-02	0	21	0	2451
2	3	2020-03-03	2	20	0	2337
3	4	2020-03-04	5	22	0	2598
4	5	2020-03-05	3	20	0	2694
...	...	...	...	...	...	...
117	118	2020-06-26	311	34	20	4281
118	119	2020-06-27	208	37	21	4101
119	120	2020-06-28	202	32	19	4158
120	121	2020-06-29	437	39	24	3648

Citibike Usage <March – June 2020>: Cleaning

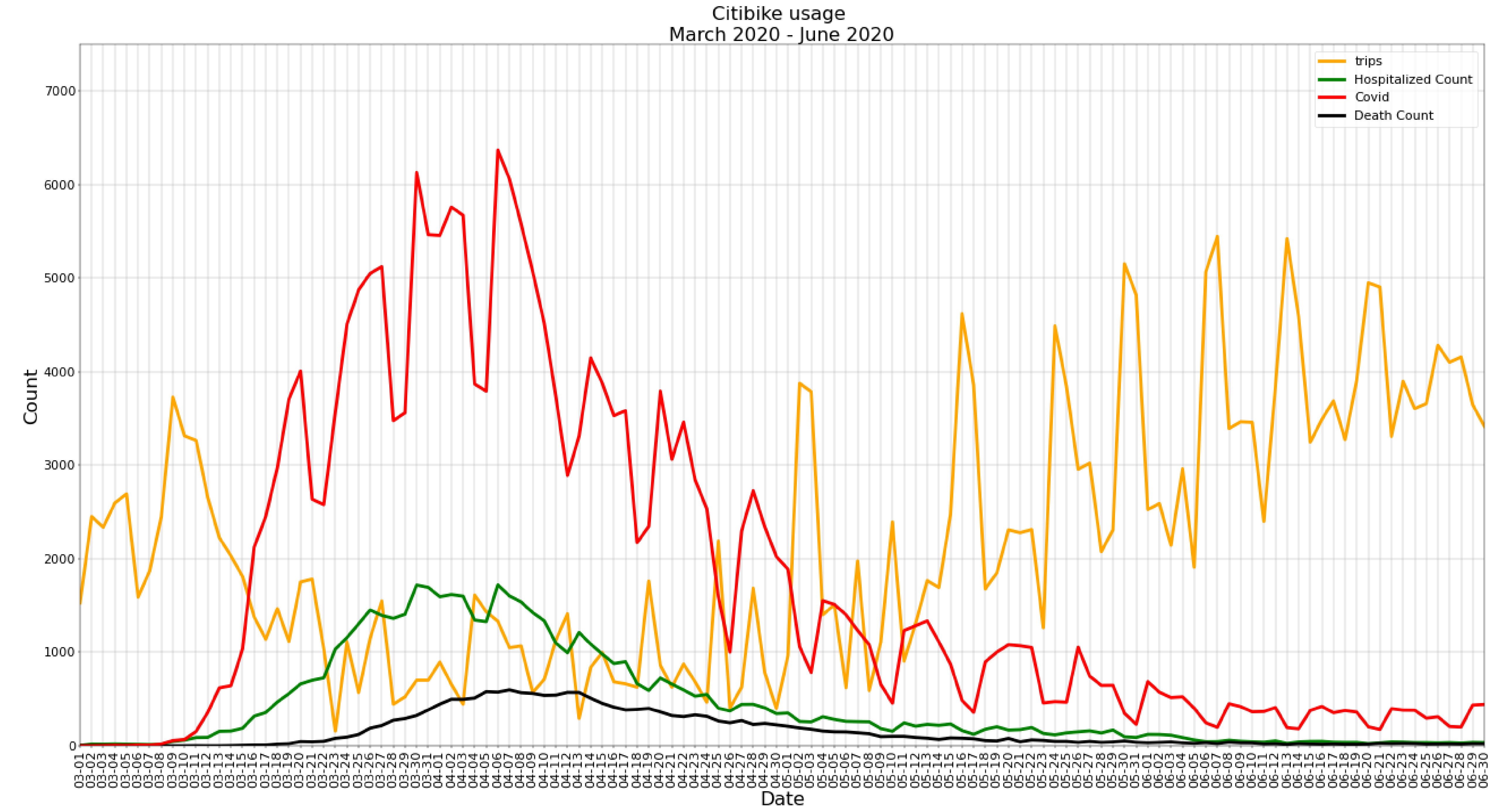


```python
#get the values of x and y axis
dates = citi_covid["Date"]
x = [datetime.datetime.strptime(d,"%Y-%m-%d").date() for d in dates]
y = citi_covid["trips"]
g = citi_covid["Hospitalized Count"]
f = citi_covid["Death Count"]
c = citi_covid["Case Count"]

fig, ax = plt.subplots(figsize=(40, 20))
#get axes
ax = plt.gca()
#format as dates
formatter = mdates.DateFormatter("%m-%d")
#set locator
ax.xaxis.set_major_formatter(formatter)
locator = mdates.DayLocator()
ax.xaxis.set_major_locator(locator)
ax.grid(True)
```

Citibike Usage <March – June 2020>: Cleaning

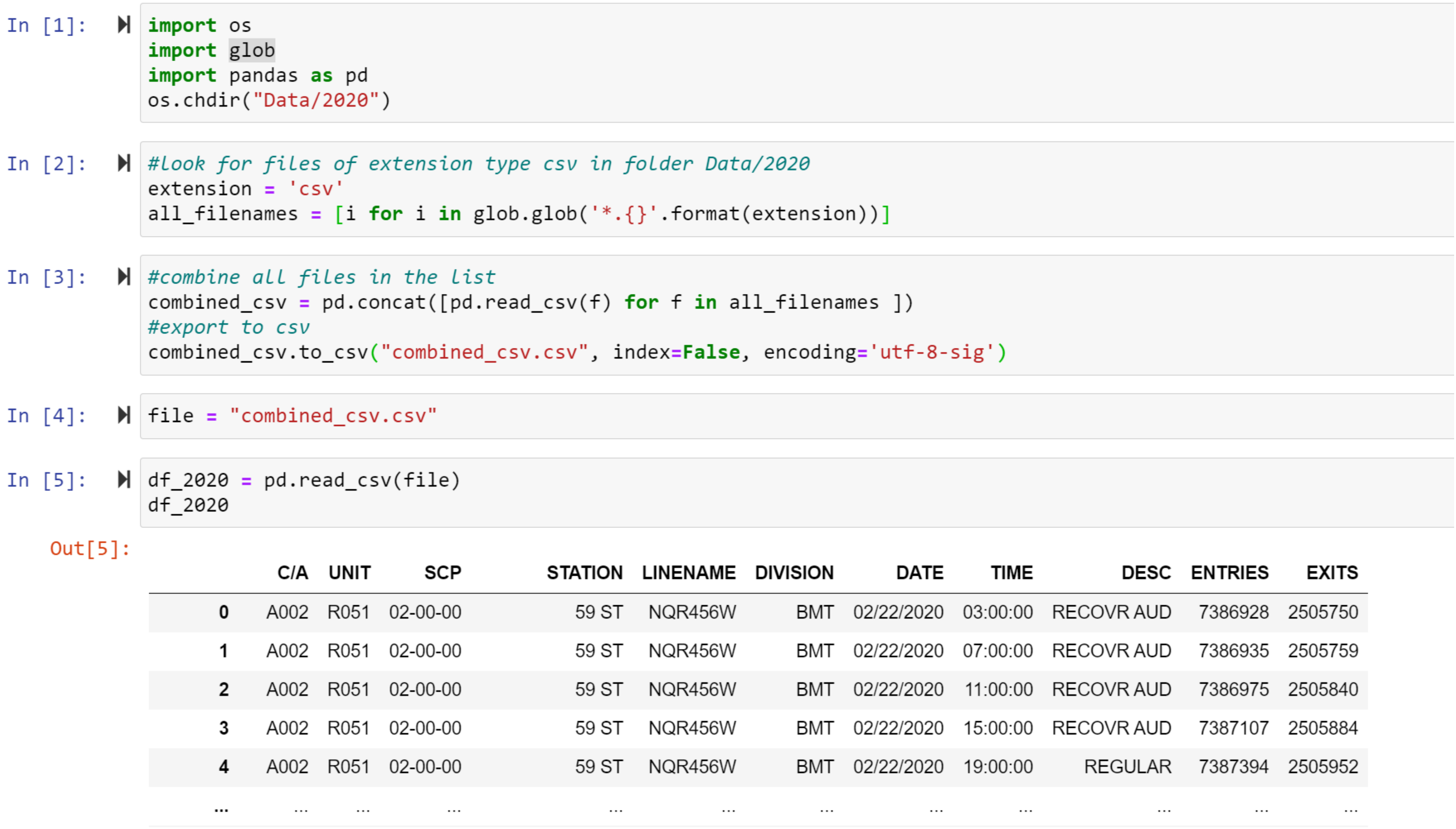


```
#plot values
trips, = plt.plot(x,y,c='orange',linewidth=5,label='trips')
Hospitalized, = plt.plot(x,g,c='green',linewidth=5, label='Hospitalized Count')
Deaths, =plt.plot(x,f,c='black',linewidth=5, label='Death Count')
covid, = plt.plot(x,c,c='red',linewidth=5, label='Covid')

#set labels, lim, size, titile and legend on the graph
plt.xlabel("Date", fontsize = 30)
plt.xticks(rotation = 90, fontsize = 20)
plt.yticks(fontsize = 20)
plt.xlim(datetime.date(2020, 3, 1), datetime.date(2020, 6, 30))
plt.ylabel("Count", fontsize = 30)
plt.ylim(0, 7500)
plt.title("Citibike usage \nMarch 2020 - June 2020", fontsize = 30)

plt.legend(handles=[trips,Hospitalized,covid,Deaths], loc="best",prop={'size': 20})

#plt.tight_layout()
plt.show()
```


Citibike usage
March 2020 - June 2020
trips
Hospitalized Count
Covid
Death Count
Count
0
1000
2000
3000
4000
5000
6000
7000
Date
03-01
03-02
03-03
03-04
03-05
03-06
03-07
03-08
03-09
03-10
03-11
03-12
03-13
03-14
03-15
03-16
03-17
03-18
03-19
03-20
03-21
03-22
03-23
03-24
03-25
03-26
03-27
03-28
03-29
03-30
03-31
04-01
04-02
04-03
04-04
04-05
04-06
04-07
04-08
04-09
04-10
04-11
04-12
04-13
04-14
04-15
04-16
04-17
04-18
04-19
04-20
04-21
04-22
04-23
04-24
04-25
04-26
04-27
04-28
04-29
04-30
05-01
05-02
05-03
05-04
05-05
05-06
05-07
05-08
05-09
05-10
05-11
05-12
05-13
05-14
05-15
05-16
05-17
05-18
05-19
05-20
05-21
05-22
05-23
05-24
05-25
05-26
05-27
05-28
05-29
05-30
05-31
06-01
06-02
06-03
06-04
06-05
06-06
06-07
06-08
06-09
06-10
06-11
06-12
06-13
06-14
06-15
06-16
06-17
06-18
06-19
06-20
06-21
06-22
06-23
06-24
06-25
06-26
06-27
06-28
06-29
06-30

Subway Usage

Subway Usage: Cleaning



In [1]:

```
import os
import glob
import pandas as pd
os.chdir("Data/2020")
```

In [2]:

```
#look for files of extension type csv in folder Data/2020
extension = 'csv'
all_filenames = [i for i in glob.glob('*.{}'.format(extension))]
```

In [3]:

```
#combine all files in the list
combined_csv = pd.concat([pd.read_csv(f) for f in all_filenames ])
#export to csv
combined_csv.to_csv("combined_csv.csv", index=False, encoding='utf-8-sig')
```

In [4]:

```
file = "combined_csv.csv"
```

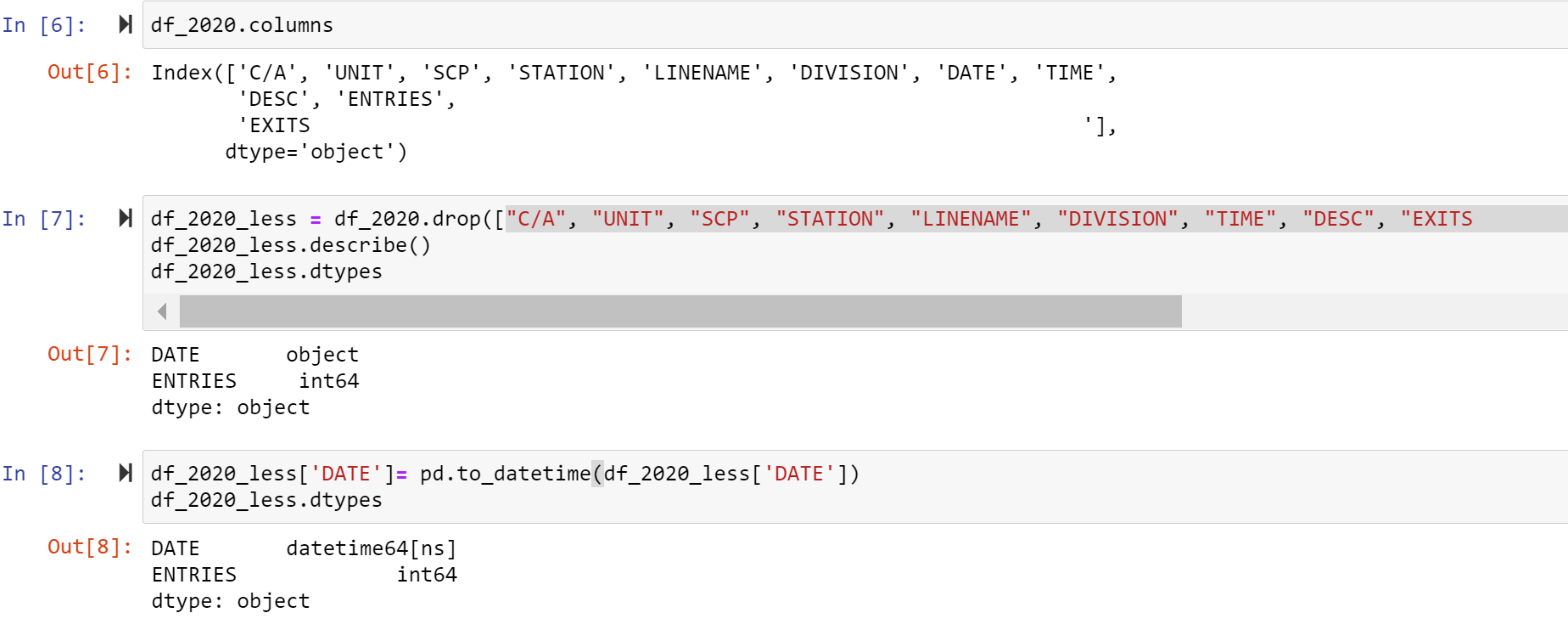
In [5]:

```
df_2020 = pd.read_csv(file)
df_2020
```

Out[5]:

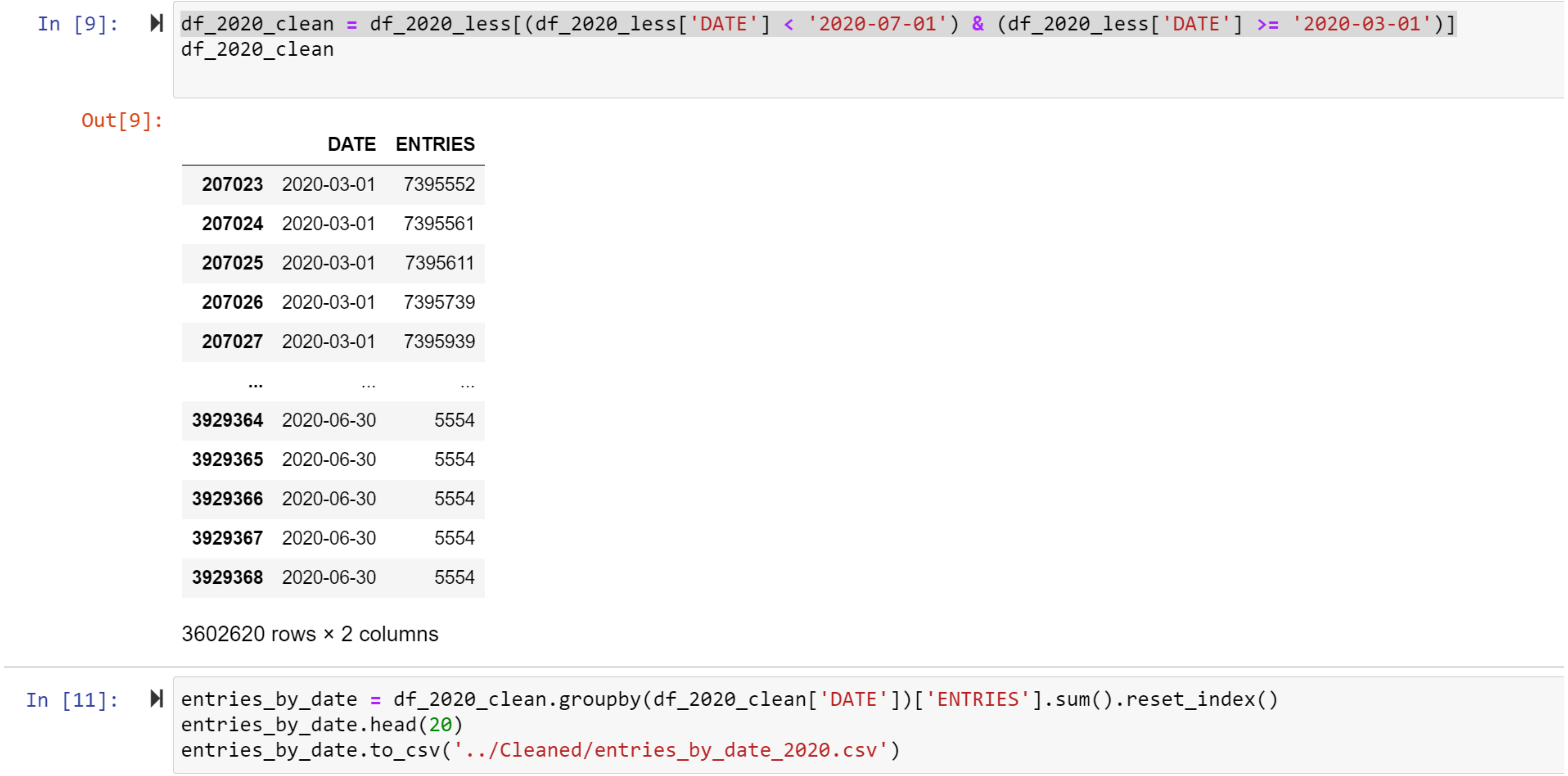
	C/A	UNIT	SCP	STATION	LINENAME	DIVISION	DATE	TIME	DESC	ENTRIES	EXITS
0	A002	R051	02-00-00	59 ST	NQR456W	BMT	02/22/2020	03:00:00	RECOVR AUD	7386928	2505750
1	A002	R051	02-00-00	59 ST	NQR456W	BMT	02/22/2020	07:00:00	RECOVR AUD	7386935	2505759
2	A002	R051	02-00-00	59 ST	NQR456W	BMT	02/22/2020	11:00:00	RECOVR AUD	7386975	2505840
3	A002	R051	02-00-00	59 ST	NQR456W	BMT	02/22/2020	15:00:00	RECOVR AUD	7387107	2505884
4	A002	R051	02-00-00	59 ST	NQR456W	BMT	02/22/2020	19:00:00	REGULAR	7387394	2505952
...	...	...	...	...	...	...	...	...	...	...	...

Subway Usage: Cleaning



```
In [6]:  df_2020.columns

Out[6]: Index(['C/A', 'UNIT', 'SCP', 'STATION', 'LINENAME', 'DIVISION', 'DATE', 'TIME',
               'DESC', 'ENTRIES',
               'EXITS                                                               '],
              dtype='object')

In [7]:  df_2020_less = df_2020.drop(["C/A", "UNIT", "SCP", "STATION", "LINENAME", "DIVISION", "TIME", "DESC", "EXITS
         df_2020_less.describe()
         df_2020_less.dtypes

Out[7]: DATE          object
        ENTRIES        int64
        dtype: object

In [8]:  df_2020_less['DATE']= pd.to_datetime(df_2020_less['DATE'])
         df_2020_less.dtypes

Out[8]: DATE          datetime64[ns]
        ENTRIES                int64
        dtype: object
```

Subway Usage: Cleaning



In [9]:

```
df_2020_clean = df_2020_less[(df_2020_less['DATE'] < '2020-07-01') & (df_2020_less['DATE'] >= '2020-03-01')]
df_2020_clean
```

Out[9]:

	DATE	ENTRIES
207023	2020-03-01	7395552
207024	2020-03-01	7395561
207025	2020-03-01	7395611
207026	2020-03-01	7395739
207027	2020-03-01	7395939
...	...	...
3929364	2020-06-30	5554
3929365	2020-06-30	5554
3929366	2020-06-30	5554
3929367	2020-06-30	5554
3929368	2020-06-30	5554

3602620 rows × 2 columns

In [11]:

```
entries_by_date = df_2020_clean.groupby(df_2020_clean['DATE'])['ENTRIES'].sum().reset_index()
entries_by_date.head(20)
entries_by_date.to_csv('../Cleaned/entries_by_date_2020.csv')
```