

A4

December 14, 2021

```
[1]: import pandas
```

Here we store the file path information to make it easier to run with different files

```
[2]: folder_path = "C:\\Users\\Aaron_\\  
    ↪Jacobson\\Documents\\UWMSDS\\Fall2021\\DATA512\\DATA512-A4\\"  
folder_path = "C:\\  
    ↪\\Users\\AaronJacobsonDesk3\\Documents\\UWMSDS\\Classes\\Fall2021\\DATA512\\DATA512-A4\\DAT  
covid_cases_file = "RAW_us_confirmed_cases.csv\\RAW_us_confirmed_cases.csv"  
covid_policy_file = "U.S.  
    ↪_State_and_Territorial_Public_Mask_Mandates_From_April_10__2020_through_August_15__2021_by_  
    ↪csv"  
mask_use_file = "mask-use-by-county.csv"
```

Here we read in all the data based on the file paths above

```
[3]: df_cases = pandas.read_csv(folder_path + covid_cases_file)  
df_policy = pandas.read_csv(folder_path + covid_policy_file)  
df_masks = pandas.read_csv(folder_path + mask_use_file)
```

Here we filter the data to just the specified county

```
[4]: county_state = "New York"  
county_name = "Queens"  
  
df_cases = df_cases[df_cases['Province_State'] == county_state]  
df_cases = df_cases[df_cases['Admin2'] == county_name]  
df_cases
```

```
[4]:
```

	Province_State	Admin2	UID	iso2	iso3	code3	FIPS	\
1943	New York	Queens	84036081	US	USA	840	36081.0	
	Country_Region	Lat	Long_	...	10/25/21	10/26/21	10/27/21	\
1943	US	40.710881	-73.816847	...	316202	316390	316522	
	10/28/21	10/29/21	10/30/21	10/31/21	11/1/21	11/2/21	11/3/21	
1943	316522	316837	316837	316837	317545	317667	317855	

[1 rows x 663 columns]

Here we modified and transform the data so it can be easily displayed in the plot

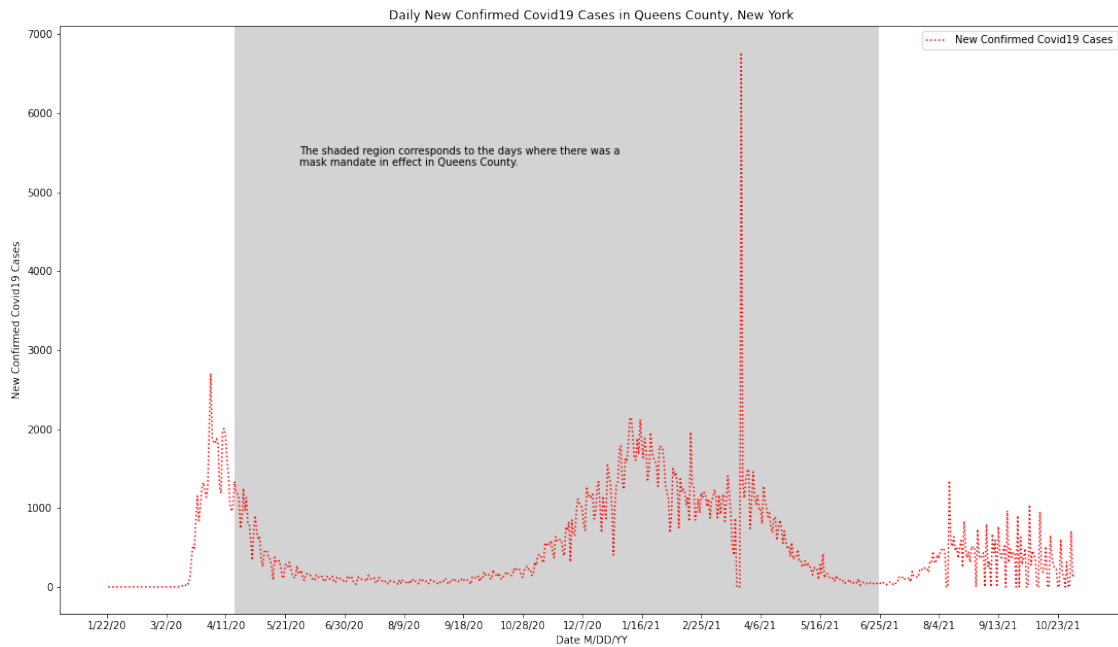
```
[5]: # remove unnecessary rows
df_cases_t = df_cases.drop([
    'Province_State',
    'Admin2',
    'UID',
    'iso2',
    'iso3',
    'code3',
    'FIPS',
    'Country_Region',
    'Lat',
    'Long_',
    'Combined_Key'], axis=1).transpose().rename({1943: 'Cases'}, axis=1)
# select the chosen county, hardcoded because of the difference in values
# between files
df_policy_queens = df_policy[df_policy['County_Name'] == 'Queens County']
# saving the FIPS code for later
county_fips = df_cases['FIPS'].values[0]
# using the FIPS code to uniquely id the right county
df_masks_queens = df_masks[df_masks['COUNTYFP'] == county_fips]
# calculating the daily new cases
df_cases_t['increase'] = df_cases_t['Cases'] - df_cases_t.shift(1)['Cases']
# removing unnecessary columns
df_policy_queens = df_policy_queens[['date', 'Face_Masks_Required_in_Public']]
# using the index to create a column for dates
df_cases_t['date'] = df_cases_t.index
# modifying the date column to match the format in the other dataframe
df_policy_queens['date'] = df_policy_queens['date'].str[:4] +
    df_policy_queens['date'].str[-2:]
# merging the dataframe with fewer dates into the dataframe with larger dates
# so that we keep all the data
df_merged = df_cases_t.merge(df_policy_queens, how='outer', left_on='date',
    right_on='date')
# creating a column that indicates whether masks are required that can be
# easily using in plotting
df_merged['masks'] = (df_merged['Face_Masks_Required_in_Public'] == 'Yes') * 1
```

Here we create the plot and save it

```
[6]: import matplotlib.pyplot as plt
fig, ax = plt.subplots()
fig.set_size_inches(18.5, 10.5)

plt.xlabel('Date M/DD/YY')
ax.plot(df_merged['increase'], color='red', linestyle=':')
ax.legend(['New Confirmed Covid19 Cases'])
```

```
plt.xticks(df_merged.index.values.tolist()[::40], df_merged['date'][::40])
plt.ylabel('New Confirmed Covid19 Cases')
ax.axvspan(df_merged[df_merged['masks'] == 1].head(1).index[0],
           df_merged[df_merged['masks'] == 1].tail(1).index[0], color='lightgrey')
plt.title('Daily New Confirmed Covid19 Cases in Queens County, New York')
plt.figtext(0.3, 0.7, 'The shaded region corresponds to the days where there was a
mask mandate in effect in Queens County.', color='black')
plt.show()
fig.savefig(folder_path + "figure.png")
```



[]: