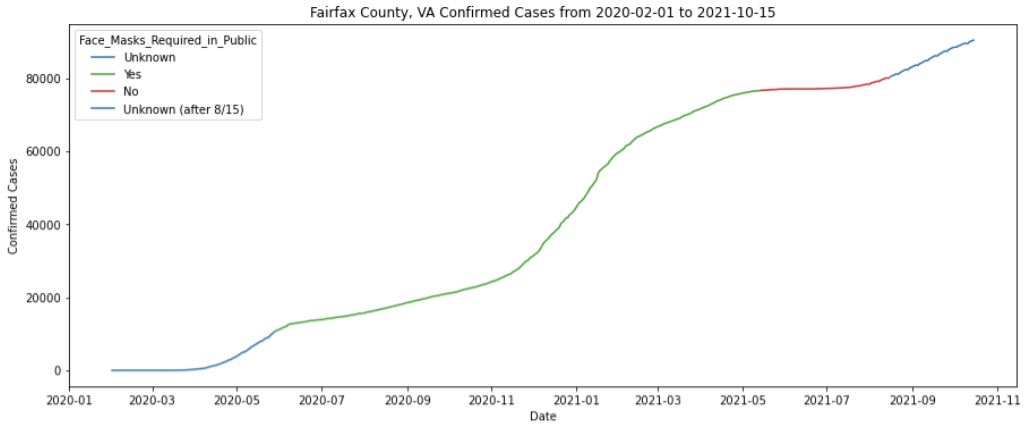
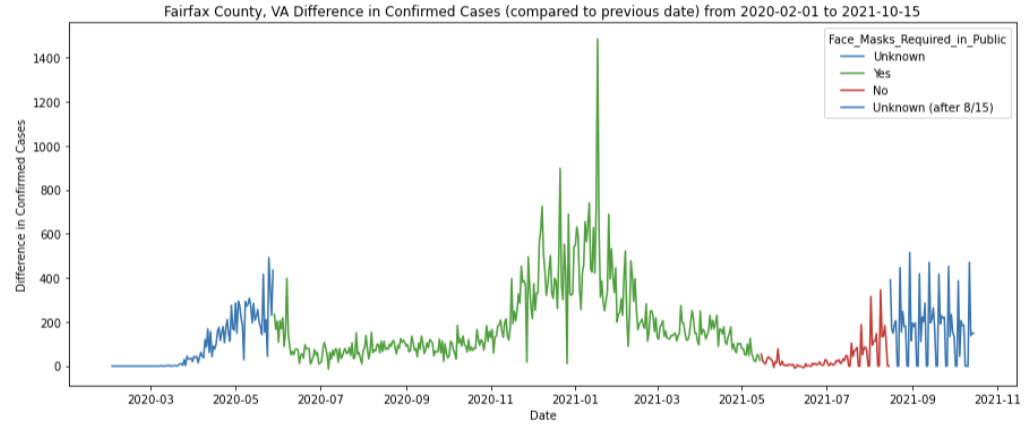
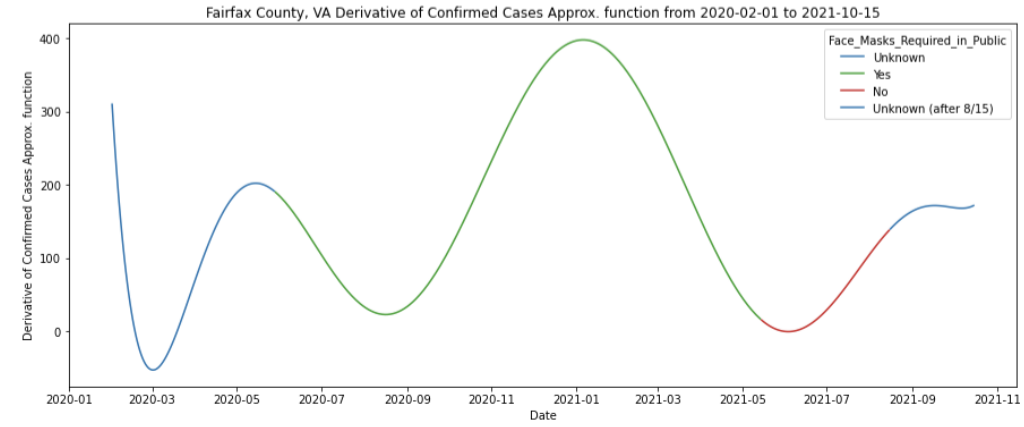
A4- Common Analysis

Sabrina Wang

Figure Explanation







I created three plots for this assignment. All three plots use the mask mandate data for color code and the confirmed cases data as the values. And the data analyzed are for Fairfax County, VA only.

The mask mandate data is provided on the US county level, and it contains the information on when individuals in a specific county were subject to state and territorial executive orders, administrative orders, resolutions, and proclamations for COVID-19 that require individuals to wear masks in public. The column used from this data is: County_Name (only used to select the targeted county), date, and Face_Masks_Required_In_Public, which is a string column with three possible entries, yes, no, or NULL. The date range of this data covers from April 10, 2020, to August 15, 2021. The confirmed case data is a daily updating version of the COVID-19 Data Repository by the Center for Systems Science and Engineering at Johns Hopkins University. At the time perform the analysis, the data covers the date range from January 22, 2020, to October 31, 2021. The confirmed cases data comes in a pivot format, so I first select the targeted county (Fairfax, VA) and unpivot the table into columns format so the date is listed as a column with another column showing the daily confirmed cases.

All three plots have the x-axes as the date, ranging from February 1, 2020, to October 15, 2021. There are three colors presented in those plots. Blue represents the mask mandate policy is missing or unknown, which is either shows up as NULL in the mandate data or the date is out of the range of the mask mandate dataset. The green color shows the presence of an official mask mandate policy requiring wearing masks in public announced by the government and red shows the masks are not required.

The first plot is confirmed cases of COVID-19 as the y-axis. It shows an overall increasing trend and the slope show how fast the confirmed cases increased. The second plot shows the daily difference, also known as daily new confirmed cases on the y-axis. This attribute is calculated by the difference of daily confirmed cases from the confirmed cases data set. Larger peaks indicate greater daily new confirmed cases, and it aligns with the sharpest increase in the first plot. The third plot has the value of the derivatives of the 9-th degree polynomial function fitted on the confirmed cases data as y-axis. The y-axis values are manually calculated by fitting the confirmed cases to a 9th-degree polynomial and then taking the derivatives of the polynomial function. It shows the same trend as the daily new confirmed cases since it is an approximation of the slope of the first plot.

All figure shows the increase of confirmed cases was at the highest between January 2021 to March 2021 and there is a slight bounce back after removing the mask mandates in public. Lately, the increase of confirmed cases started to decrease again.

Reflection

Two things I learned from these assignments are the issue with data availabilities and how to approach the analysis on daily change or derivatives.

For this part of the project, the task is to understand "How did masking policies change the progression of confirmed COVID-19 cases from February 1, 2020, through October 15, 2021". However, the mask mandates data set only contains data from April 10, 2020, to August 18, 2021. This brings up a common question data analysts need to face often: if the existing data cannot answer the pre-set question, do we proceed with the analysis with the missing data, or do we want to reframe the question to another one that can be better answered by existing data. My current approach to this issue is to fill the missing dates with NULL values (see code cell [6]), which later is mapped to be "Unknown" for visualization purposes (see code cell [10]).

[6]:
```
# Create the rows for missing date of mastk policy

before_april = pd.DataFrame([['Fairfax County','2020-02-01', np.
 ↪nan]],columns=['County_Name','date','Face_Masks_Required_in_Public'])
before_april['date']=pd.to_datetime(before_april['date'])
d = pd.to_datetime('2020-02-01')
while d < pd.to_datetime('2020-04-09'):
    before_april=before_april.append([{'County_Name':'Fairfax County','date':
 ↪d+pd.DateOffset(1), 'Face_Masks_Required_in_Public':np.nan}],␣
 ↪ignore_index=True)
    d = d+pd.DateOffset(1)


after_aug = pd.DataFrame([['Fairfax County','2021-08-16','Unknown (after 8/
 ↪15)']],columns=['County_Name','date','Face_Masks_Required_in_Public'])
after_aug['date']=pd.to_datetime(after_aug['date'])
d = pd.to_datetime('2021-08-16')
while d < pd.to_datetime('2021-10-15'):
    after_aug=after_aug.append([{'County_Name':'Fairfax County','date':d+pd.
 ↪DateOffset(1), 'Face_Masks_Required_in_Public':'Unknown (after 8/15)'}],␣
 ↪ignore_index=True)
    d = d+pd.DateOffset(1)


policy_selected = pd.concat([before_april, policy_selected, after_aug])
policy_selected = policy_selected.sort_values(by=['date'])
```

Another problem that comes from the same data availability issue is the lack of granularity of population data. Infection rate is an important indicator of the spread of the virus but it requires data on population for the region, which is Fairfax County, VA in my case. However, due to the limitation of the population census, the population data cannot reach the daily granularity. Only existing data on population is on the yearly level, which means there is only one single constant number for all days in 2020 and another different constant number for all days in 2021. I am

skeptical to calculate any sort of rate based on those "constant" populations since people will move in and out throughout the year. Using a constant population for a time series data with more than 300 data points does not seem reasonable. Especially when considering the jump between 2020 population data to 2021 population data, it can likely create a spurious trend due to the "population change". Based on all the above, I believe that for a more granular level analysis, confirmed cases are a better indicator than infection rate if no granular level population data is presented. Lastly, the data on mask usage is very interesting, however, like the population data, without enough granularity, I am not sure how useful it will be towards the time series type of analysis. I might be able to provide some qualitative sight but hardly can draw some quantitative results from it.

The second reflection is on how to approach the change in confirmed cases. Initially, I chose the approach of taking daily differences, which can also be interpreted as the new daily cases (see code cell [11]).

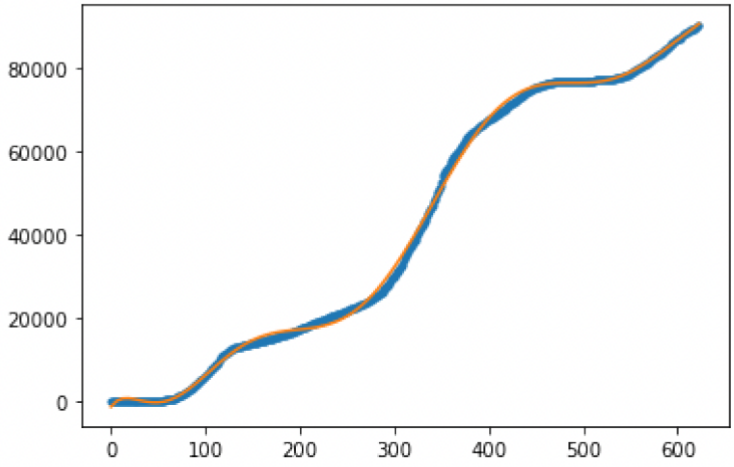
```
[11]: # Calculate the change in confirmed cases
      policy_confirmed['case_diff'] = policy_confirmed['cases'] -␣
       ↪policy_confirmed['cases'].shift(1)
```

This shows some interesting insight which is included in the other write-up. Furthermore, I took another step in trying to fit a polynomial to the data from February 1, 2020, to October 15, 2021. After a couple of tries, a polynomial of degrees of 9 seems better fitting the trend (see code cell [13-16].

[13]:
```
# A polynomial of degree 9 seems a good approx. of the trend of confirmed cases
z = np.polyfit(function_df['index'], function_df['cases'], 9)
p = np.poly1d(z)
```

[14]:
```
xp = np.linspace(0, 623, 100)
plt.plot(function_df['index'], function_df['cases'], '.', xp, p(xp), '-')
```

[14]:
```
[<matplotlib.lines.Line2D at 0x7f96c0cd1f10>,
 <matplotlib.lines.Line2D at 0x7f96c0cd1f40>]
```



[15]:
```
# Take derivative of the polynomial p
p_d = np.polyder(p)
```

[16]:
```
# Calculate the derivative values for index 0 to0 49
function_df['derivative']=p_d(function_df['index'])
```

I did not intend this model to perform any prediction thus I was not worried about the issue of overfitting. Then I took the derivative of the 9th-degree polynomial model and plot the derivative on the dates. It turns out the derivative shows pretty much the same trend as the daily differences. This is an expected but still interesting finding. It seems to me since the day-level granularity is small enough thus it behaves in a similar way as a derivative function.