Introduction:

This project attempts to use a prediction/forecasting algorithm in order to fit a model on past Covid-19 data in order to try and predict future data. The model in question is an ARIMA model which stands for ‘autoregressive integrated moving average’. This model can be used to understand potential trends in past data, as well as try to predict future trends in data based on the past trends in data.

Data:

The data that will be used can be found at the following link:
https://github.com/CSSEGISandData/COVID-19/blob/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_US.csv

Specifically, we will be looking at confirmed Covid-19 cases within the United States (US), which includes data about all states and provinces spanning roughly over 3 years, starting from 1/22/20 to 3/9/2023. The data itself has dimensions of 3342 rows by 1154 columns. Not all information is relevant to the actual analysis, and the data is preprocessed to reduce down to necessary features rows and columns.

Relevant Feature Columns:

- Province_State: State or Province
- 1/22/20 to 3/9/2023: # of confirmed cases up to current date

There are multiple cities per state/province, so we group them by location and get the total sum of current cases for every day for all cities in each location.

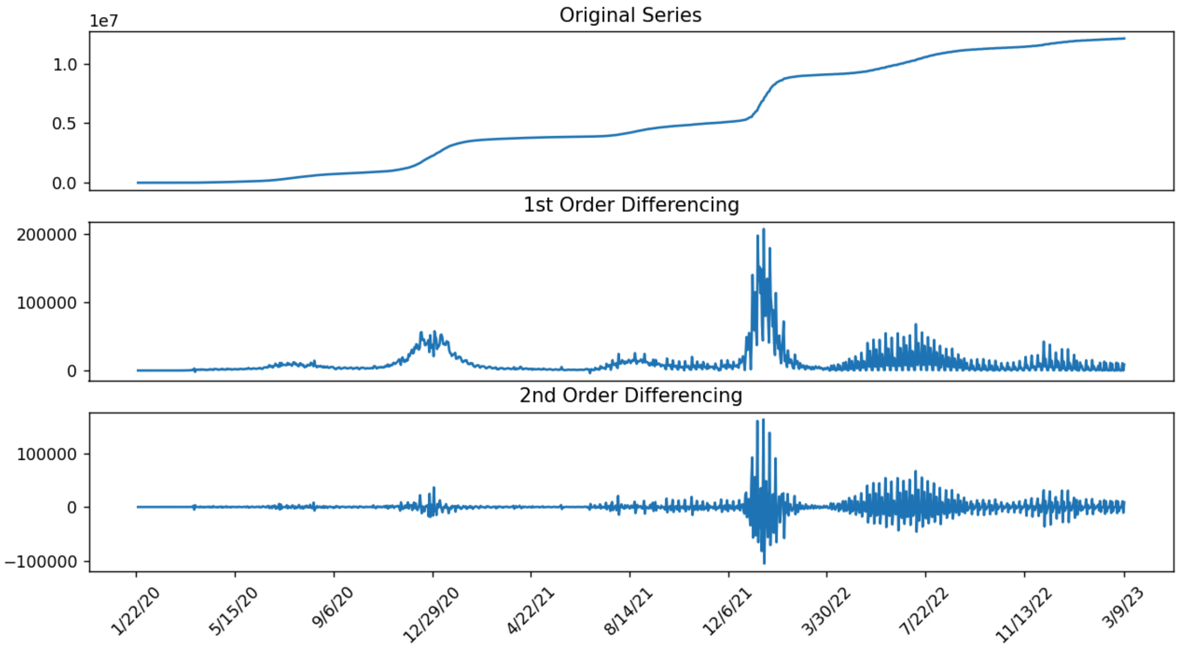
For this analysis, we will be only focusing our analysis on one state (due to the time constraints for this project), that being California. That being said, the methods used for this analysis are applicable to any state or province.

However we must also restructure the data to a format suitable for ARIMA. Currently, the data gives us a running sum of all the data up until the last day, which results in an upwards trend when plotted. Instead, we get the difference between each day to replace the running sum.

Methodology:

In order to use an ARIMA model, there are 3 parameters that have to be determined (***p***, ***d***, ***q***), each corresponding to the acronym AR (auto regressive), I (integrated), MA (moving average) (CFI Team, 2023). These values can be found either manually through plotting or automatically through a function called ‘auto_arima’ from the pmdarima package. In this analysis, I have opted to do it manually for easier clarity for those without a statistical background to follow along.

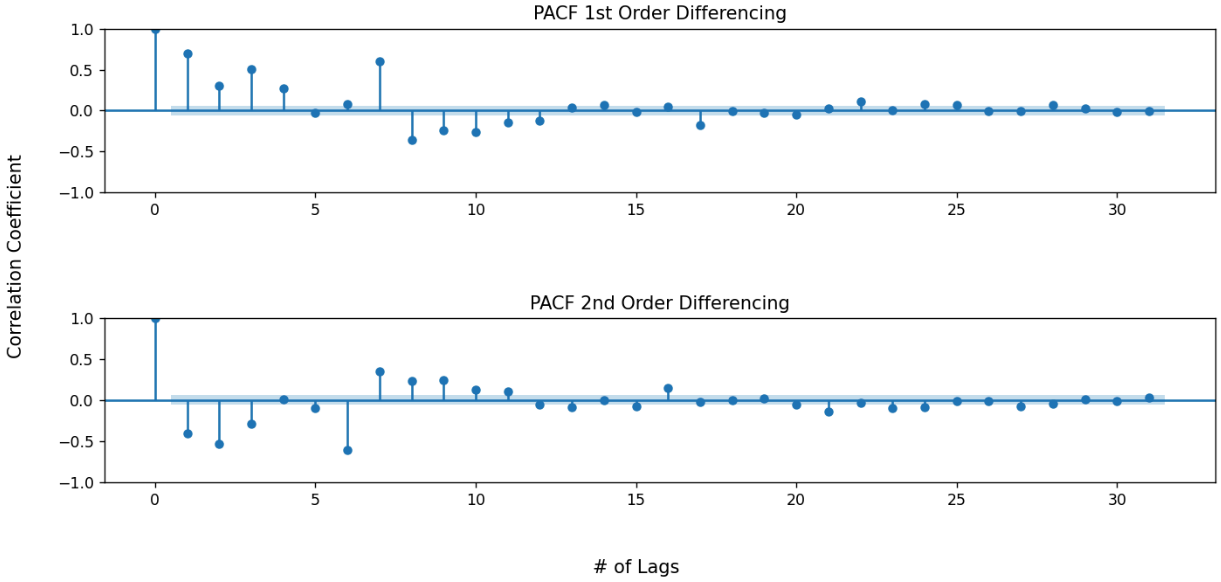
The first step is to find ***d***, the integrated (I) portion of the model. There is currently no method that can find the optimal value of ***d***, and must be found manually regardless of how the parameters are being chosen. This value determines if the data that we have currently is considered stationary, which is a requirement in ARIMA models. In simple terms, stationary data is defined as data that does not show trends or seasonality (recurring patterns over a fixed time period) (Rasheed, R. 2020). Shown below is our data with the original distribution, as well as our data with 1 and 2 levels of differencing. As a note, differencing is a technique used by data scientists to turn non-stationary data into stationary data:



We can see that the original data follows an upward trend over time, making it non-stationary. Performing 1 level of differencing still shows minor trends, while 2 levels of differencing shows none of the trends as seen originally. Both levels of differencing will be used and compared to one another.

We can also use an Augmented Dickey Fuller (ADF) test to mathematically check if the data is stationary or not (Prabhakaran, S., 2022). We set a threshold of 0.05 for the alpha value, with the null hypothesis (H0) being that the data is stationary. This only tells us whether or not the data is stationary or not, and not the degree that it needs to be differenced by.

Next we find the value of ***p***, or the autoregressive (AR) portion of the model. Autoregression refers to the direct relationship between current observation and past values, referred to as lags. In other words, how correlated the current data is based on past values. Shown below is a plot of the **Partial Autocorrelation Function (PACF)** for both orders of differencing:



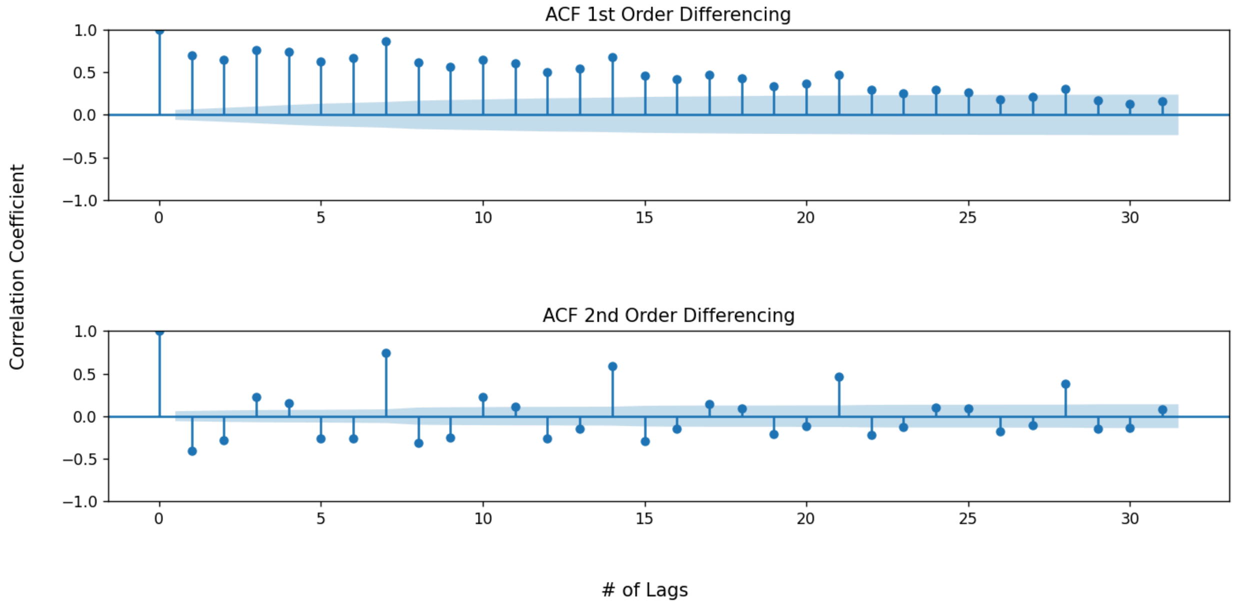
To determine the value of ***p***, we count the number of significant terms at each lag that show a high enough correlation that is outside of the shaded region. That region represents a 95% confidence interval from our data, and anything inside is considered statistically irrelevant in this scenario.

For the first order differencing, lags 1 through 4, 7 through 10 and 17 are the values I chose as the most significant. Summing them up, I get a ***p*** value of 9. Lags 11, 12, and 22 could also be considered significant as well, but that is up to the user whether or not they want to include values that are outside the area but still relatively close to the shaded region.

We do the same thing for the second order differencing, with lags 1-3, 6-9, 16, and 21 for a ***p*** value of 9. Once again, lags such as 5, 10, or 11 could be included or not based on the user's preferences.

As a note, a lag of 0 is omitted as the current observation's past value at lag 0 is just itself, so it will always be 100% correlated.

Lastly we find the value of ***q***, or the moving average (MA) portion of the model. Moving average refers to the direct and indirect relationship between current observation and all its past values. In a sense, we are looking at a 'window' of relationship values from current observation to previous values. We follow the exact same steps as earlier to find the ***q*** value, except we look at the **autocorrelation function (ACF) plot** instead of the partial autocorrelation function plot:



Following the steps from earlier, we get a ***q*** value of 21 (lags 1-21) for one level of differencing, and a ***q*** value of 13 (lags 1-3, 5-9, 12, 14-15, 21, 28). Once again, certain lags can be included or excluded based on the user's preference.

We use a PACF for choosing the parameter of AR as we want a direct linear relation between the current observation against any lag, without any trends affecting the data in between. By that logic, we use an ACF plot for choosing the parameter of MA as we want both the direct and indirect relationship between the current observation with all lags, accounting for trends and seasonality between them (GeeksforGeeks, 2023).

Results:

(Raw result output is displayed in a separate file, below only shows results that are relevant to analysis)

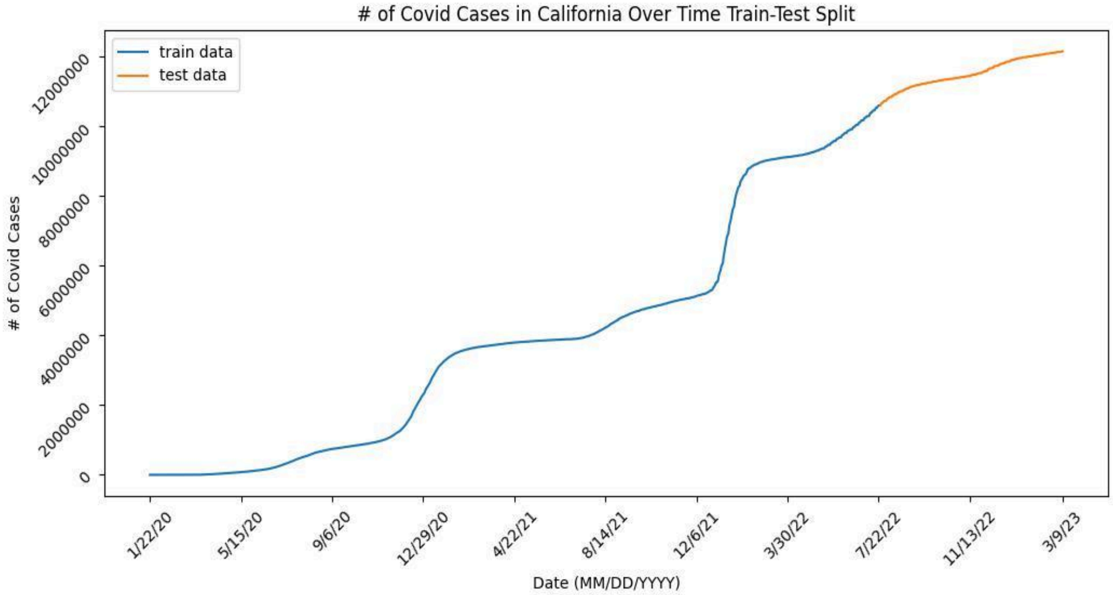
Augmented Dickey-Fuller test: (0.16602147708716247, True)
p-value > 0.05, fail to reject null hypothesis (H0), the data has a unit root and is non-stationary.

Since the ADF test failed to reject the null hypothesis, we know that our data is non-stationary and at least 1 level of difference will be needed.

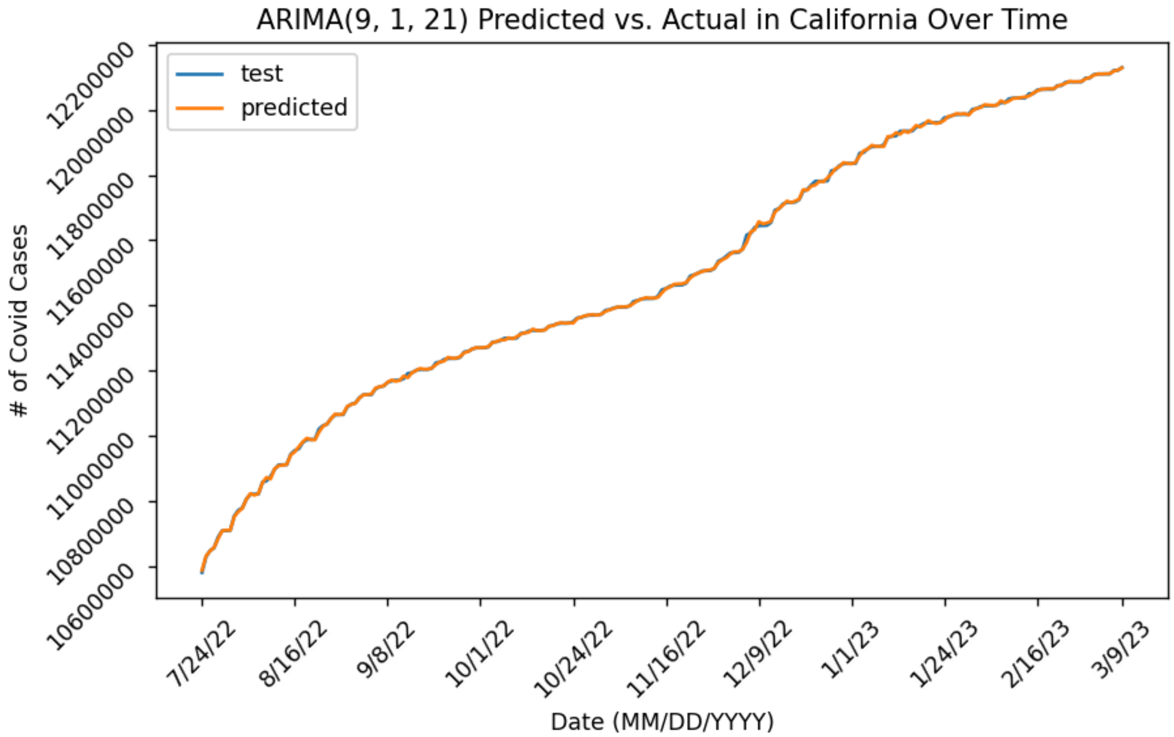
Using the methods from above, we now have our (***p***, ***d***, ***q***) values for two models:

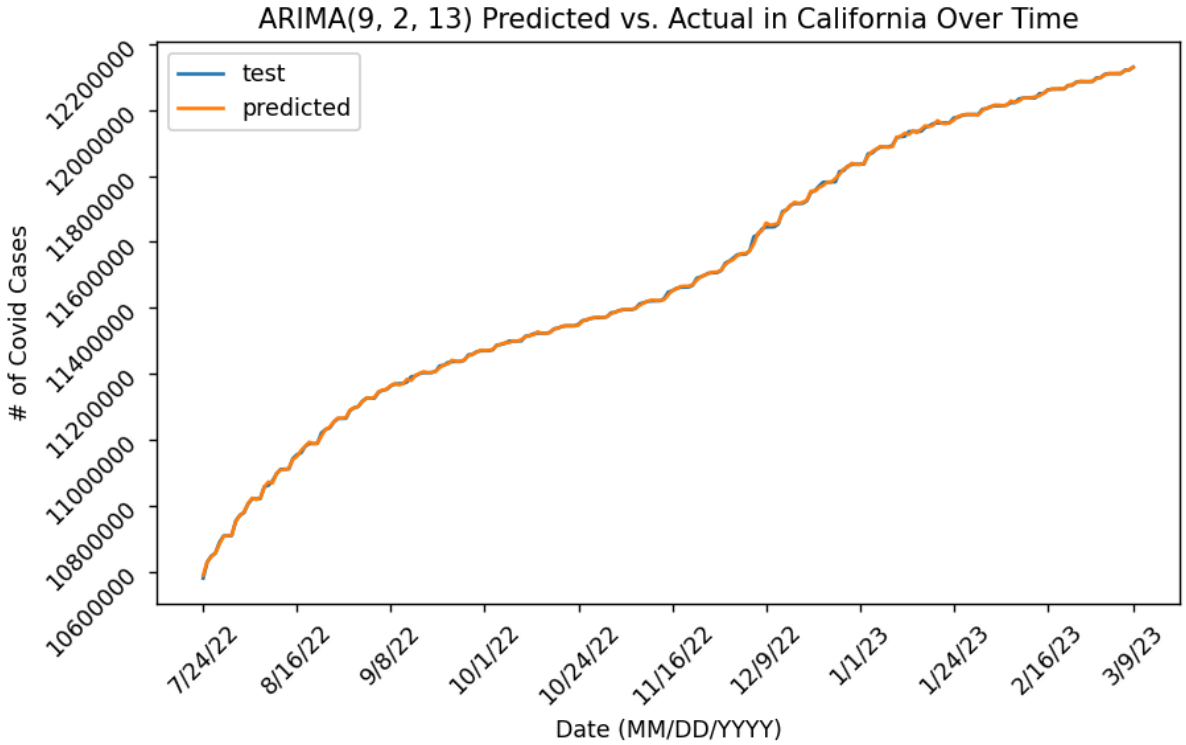
- ARIMA Model 1: [***p***=9, ***d***=1, ***q***=21]
- ARIMA Model 2: [***p***=9, ***d***=2, ***q***=13]

A 80-20 train/test split is performed with the first 80% of the data being the training data, and the last 20% being the test data as shown below:



We then plot the two model's predicted vs. actual values visually to see how well they do:





As we can see, both models seem to fit the test data pretty well visually. In order to compare the accuracy of the model, I opted to use a few metrics such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). Both MAE and RMSE are metrics that calculate the residuals (gap/differences between predicted and true values), with MAE being the absolute difference and RMSE being the root of the squared differences, while MAPE is a percentage form of the MAE. As with most error metrics, the lower the score the better the model.

For the two plots above, we get the following metrics:

```
ARIMA Model with parameters (9, 1, 21)
----------------------------------------------------
Mean Absolute Error: 2554.4926
Mean Absolute Percentage Error: 0.0002218
Root Mean Squared Error: 3769.0348
```

```
ARIMA Model with parameters (9, 2, 13)
----------------------------------------------------
Mean Absolute Error: 2482.2205
Mean Absolute Percentage Error: 0.0002156
Root Mean Squared Error: 3713.3582
```

On average the predicted values were 2554 away from the true value for ARIMA(9, 1, 21), and 2482 away from the true value for ARIMA(9, 2, 13) when considering the MAE. For both values, that’s roughly about 0.02% of the true value away from the true value according to the MAPE.

On average the predicted values had a RMSE of 3769 for ARIMA(9, 1, 21) and a RMSE of 3713 for ARIMA(9, 2, 13).

Conclusion:

In terms of performance, both models are comparable to each other. Visually both of the models look like it fits the test data extremely well. When looking at the error metrics, we can see that ARIMA(9, 2, 13) does slightly better than ARIMA(9, 1, 21), with a slightly lower MAE, MAPE, and lower RMSE as well.

Some ways to see if these were the best models would be to test more values of ARIMA and compare their metrics. In that sense, automatically finding the optimal parameters of ARIMA then verifying it manually through a plot would be the best route to achieve this. However, any values of ***p*** or ***q*** over 5 becomes computationally expensive at an exponential rate and take much longer to train, which is why it wasn’t performed in this analysis. However if one wants to find a more accurate model, the above method would be a way to achieve it.

Sources:

CFI Team. (2023, November 21). *Autoregressive Integrated moving average (ARIMA)*.

Autoregressive Integrated Moving Average (ARIMA). https://corporatefinanceinstitute.com/resources/data-science/autoregressive-integrated-moving-average-arima/#:~:text=Understanding%20the%20ARIMA%20Model&text=The%20%E2%80%9CI%E2%80%9D%20stands%20for%20integrated,observations%20from%20the%20previous%20values

GeeksforGeeks. (2023, November 22). *Autocorrelation and Partial Autocorrelation*.

https://www.geeksforgeeks.org/autocorrelation-and-partial-autocorrelation/#

Prabhakaran, S. (2022, April 4). *Augmented Dickey Fuller Test (ADF Test) – Must Read*

Guide. Machine Learning Plus. https://www.machinelearningplus.com/time-series/augmented-dickey-fuller-test/

Rasheed, R. (2020, July 12). *Why does stationarity matter in time series analysis?*.

Medium. https://towardsdatascience.com/why-does-stationarity-matter-in-time-series-analysis-e2fb7be74454