0.1 1.b. Map traffic speed to Google Plus Codes

Google Plus Codes divide up the world uniformly into rectangular slices (link). Let's use this to segment traffic speeds spatially. Take a moment to answer: **Is this spatial structure effective for summarizing traffic speed?** Before completing this section, substantiate your answer with examples of your expectations (e.g., we expect A to be separated from B). After completing this section, substantiate your answer with observations you've made.

Our expectation is that uniform rectangular slices are not the most effective spatial structure for summarizing traffic speeds, because it groups sections of geographical regions into uniform rectangles, regardless of the type of neighborhood / subregion that the rectangle includes. Commercial districts are mixed in with residential districts, as well as highways, parks, schools, and other features that heavily play into computing average traffic speeds.

After completing the rest of the section, we find that expectations were correct, as supported by plus codes having a lower across-cluster standard deviation of within-cluster means, as compared to other spatial structures such as census tracts. This shows that, using pluscodes, the within-cluster means are relatively similar, so it's harder to tell which subregions are high speed / low speed, preventing us from making meaningful insights about the areas.

0.1.1 1.b.v. How well do plus code regions summarize movement speeds?

The following will give us an idea of how well the average represents traffic speed per plus code region. For these questions, we'll refer to a "plus code region" as a "cluster":

1. **Plot a histogram of the within-cluster standard deviation**.
2. **Compute across-cluster average of within-cluster standard deviation**.
3. **Compute across-cluster standard deviation of within-cluster average speeds**.
4. **Is this average variance reasonable?** To assess what "reasonable" means, consider these questions and how to answer them: (1) Do plus codes capture meaningful subpopulations? (2) Do differences between subpopulations outweigh differences within a subpopulation? Use the statistics above to answer these questions, and compute any additional statistics you need. Additionally explain *why these questions are important to assessing the quality of a spatial clustering.*

Hint: Run the autograder first to ensure your variance average and average variance are correct, before starting to draw conclusions.

In the first cell, write your written answers. In the second cell, complete the code.

For plus codes, the average variance is not a reasonable statistic.

Plus codes do not capture meaningful subpopulations because they split a region up into uniform rectangles which can contain "slices" of many different traffic zones, such as highways (high speed), school zones (low speed), commercial & residential districts, etc. This causes within-cluster means to be less meaningful, because the mixture of these zones causes each cluster's mean to be pretty similar.

In theory, average variance represents how much the distribution of inner-cluster means is spread out. However, this statistic is inflated because the amount of data available for each cluster isn't consistent. For clusters with a lot of data points, we would expect the mean and standard deviations of traffic speeds to be accurate representations of real world traffic in that pluscode. However, statistically, clusters with less available data are more likely to have higher within-cluster standard deviations.

Therefore, computing the across-cluster mean of within-cluster standard deviations gives disproportionate weight to clusters with less data, because each of the clusters hold the same weight when computing an average across all clusters. This inflates the average variance figure, thereby making it a less reasonable and reliable statistic.

This is important in assessing the quality of spatial clustering because we need quantitative metrics to see if subpopulations carry any real meaning. For instance, a clustering method with a very high average variance tells us that when you look at any particular cluster, it is likely that the traffic speeds within that cluster vary heavily. Then, we can step back and assess if this clustering method has meaningful subpopulations, as we now see that we may have grouped too many high-speed and low-speed regions together in the same clusters.

In [ ]:
```
speed_variance_by_pluscode = speeds_to_gps.groupby(['plus_latitude_idx','plus_longitude_idx']).a
average_variance_by_pluscode = np.mean(speed_variance_by_pluscode['speed_mph_mean'])
variance_average_by_pluscode = speeds_to_gps.groupby(['plus_latitude_idx','plus_longitude_idx']]
plt.hist(speed_variance_by_pluscode['speed_mph_mean']);
plt.ylabel("Number of Pluscodes")
plt.xlabel("Within-Cluster Standard Deviation of Speed in mph")
plt.title("Speed Variance by Pluscode");
```

0.1.2 1.c.iv. How well do census tracts summarize movement speeds?

The following will give us an idea of how well the average represents traffic speed per plus code region. For these questions, we'll refer to a "census tract" as a "cluster":

1. **Plot a histogram of the within-cluster standard deviation**.
2. **Compute across-cluster average of within-cluster standard deviation**.
3. **Compute across-cluster standard deviation of within-cluster average speeds**.
4. **Is this average variance reasonable?** To assess what "reasonable" means, consider these questions and how to answer them: (1) Do plus codes capture meaningful subpopulations? (2) Do differences between subpopulations outweigh differences within a subpopulation? Use these ideas to assess whether the average standard deviation is high or not.

Note: We are using the speed metric of miles per hour here.

Just like before, please written answers in the first cell and coding answers in the second cell.

This average variance is much more reasonable than the pluscode average variance. While the calculation for both ends up being similar (8.68 vs. 8.30), we see that average variance as a statistic is much more meaningful when used on census tracts.

This is largely because census tracts are divided up to have roughly the same population in each tract, which reduces the problem we explained earlier, where clusters with low populations (number of data points) are disproportionately factored into the calculation of across-cluster average variance. In other words, because census tracts have more similar populations than pluscodes, we see that differences (deviation) within a cluster no longer outweigh the differences across all clusters, thereby making the average variance a much more reliable statistic.

Similarly, census tracts summarize movement speeds better than plus code regions because the subregions are defined based on neighborhood type and population counts. For instance, specific suburbs are zoned together in a single census tract, seperated from commercial districts, industrial areas, etc.

This makes each census tract more different from each other, which can be seen by the clusters' means varying more from one another. This is why we see an increase in across-cluster standard deviation of within-cluster average speeds.

```
In [ ]: speed_variance_by_tract = speeds_to_tract.groupby(['MOVEMENT_ID']).agg(np.std)
        average_variance_by_tract = np.mean(speed_variance_by_tract['speed_mph_mean'])
        variance_average_by_tract = speeds_to_tract.groupby(['MOVEMENT_ID']).mean()["speed_mph_mean"].s
        plt.hist(speed_variance_by_tract['speed_mph_mean'])
        plt.ylabel("Number of Census Tracts")
        plt.xlabel("Within-Cluster Standard Deviation of Speed in mph")
        plt.title("Speed Variance by Tract");
```

0.2 1.d. What would be the ideal spatial clustering?

This is an active research problem in many spatiotemporal modeling communities, and there is no single agreed-upon answer. Answer both of the following specifically knowing that you'll need to analyze traffic patterns according to this spatial clustering:

1. **What is a good metric for a spatial structure?** How do we define good? Bad? What information do we expect a spatial structure to yield? Use the above parts and questions to help answer this.
2. **What would you do to optimize your own metric for success in a spatial structure?**

See related articles:

- Uber's H3 link, which divides the world into hexagons
- Traffic Analysis Zones (TAZ) link, which takes census data and additionally accounts for vehicles per household when dividing space

A good spatial structure is one where we can look at each cluster and generate area-specific insights, for instance looking at how traffic speeds change in different neighborhoods / subregions over time. In other words, when each cluster has a "type" (suburb, low-income or high income, commercial districts, downtown, highways, etc), we can make meaningful conclusions about how each "type" is affected by something like a lockdown.

A bad spatial structure, in this case, is one where lines are drawn somewhat arbitrarily, grouping together different types of neighborhoods and districts in the same cluster. Lack of differentiation prevents meaningful insight across clusters. Also, a bad spatial cluster is one where the size of each cluster is too small or too big. If the clusters are too small, we don't generate much more insight than simply looking at each node. If the clusters are too large, entire districts / zones are grouped together, preventing us from making detailed insights.

A good metric for spatial clustering would be a high variance of within-cluster means. If we see a high variance of within-cluster means, it indicates that the clusters represent different types of neighborhood / subregions, i.e. residential vs. industrial / highways. Another good metric would be across-cluster standard deviation of within-cluster number of data points. This would tell us if the amount of available data is similar across all clusters. We would likely want this variance to be low, meaning each cluster has roughly the same number of data points, allowing our calculations made across all clusters to be evenly-weighted.

Addressing the more general question, we think the ideal spatial clustering for traffic speeds would be something like census tracts, where the entire region is separated by neighborhood type. Furthermore, if each cluster had the same amount of data available, statistics such as across-cluster standard deviation and mean would be more meaningful, because each cluster would be weighed equally, giving a more accurate representation instead of low-data clusters holding disproportionate weight.

If we were creating our own spatial structure, we would want to "draw" and "redraw" our lines, adjusting in different ways to optimize for the metrics outlined above. For instance, if our across-cluster standard deviation of within-cluster means is too low, our subregions might be too similar to one another. In this

case, we might want to split the region into more clusters, allowing each one to represent a more specific part of the city.

0.2.1 2.a.i. Sort census tracts by average speed, pre-lockdown.

Consider the pre-lockdown period to be March 1 - 13, before the first COVID-related restrictions (travel bans) were announced on March 14, 2020.

1. **Report a DataFrame which includes the *names* of the 10 census tracts with the lowest average speed**, along with the average speed for each tract.
2. **Report a DataFrame which includes the *names* of the 10 census tracts with the highest average speed**, along with the average speed for each tract.
3. Do these names match your expectations for low speed or high speed traffic pre-lockdown? What relationships do you notice? (What do the low-speed areas have in common? The high-speed areas?) For this specific question, answer qualitatively. No need to quantify. **Hint**: Look up some of the names on a map, to understand where they are.
4. **Plot a histogram for all average speeds, pre-lockdown**.
5. You will notice a long tail distribution of high speed traffic. What do you think this corresponds to in San Francisco? Write down your hypothesis.

Hint: To start off, think about what joins may be useful to get the desired DataFrame.

The names of the lowest-speed census tracts definitely do match my expectation because they're all in high-congestion areas of San Francisco. As someone who grew up here, I expected neighborhoods like the Tenderloin, Mission District, and the Financial District to be very slow because a huge volume of cars and trucks occupy the streets throughout most of the day, slowing down traffic. Furthermore, many of these neighborhoods contain unusual roads and traffic conditions, such as trolleys, intersecting one-way streets, double bike lanes, looping freeway entrances, and other features that often cause confusion and slow down traffic due to uncertainty.

The names of the high-speed census tracts also match my expectation because they're almost all near highways or in industrial parts of the Bay Area. For tracts that include highways, we would expect a much higher average traffic speed because highways have much higher speed limits. For tracts in industrial areas, such as Petrolite Street in Richmond, CA, we see many large open roads almost entirely occupied by working trucks and cars. It makes sense that these areas have high average traffic speeds because there is a lot of open space and not much residential / commercial activity.

Plot the histogram

```
In [ ]: plt.hist(averages_pre)
        plt.xlabel("Average Speed Within Census Tract (mph)")
        plt.ylabel("Number of Census Tracts")
        plt.title("Hisogram of Bay Area Census Tracts Average Speed Pre Lockdown");
```

0.2.2 2.a.ii. Sort census tracts by average speed, post-lockdown.

I suggest checking the top 10 and bottom 10 tracts by average speed, post-lockdown. Consider the post-lockdown period to be March 14 - 31, after the first COVID restrictions were established on March 14, 2020. It's a healthy sanity check. For this question, you should report:

- **Plot a histogram for all average speeds, post-lockdown.**
- **What are the major differences between this post-lockdown histogram relative to the pre-lockdown histogram above**? Anything surprising? What did you expect, and what did you find?

Write the written answers in the cell below, and the coding answers in the cells after that.

The biggest difference between the histograms is that average speed went up significantly after the lockdown. Before the lockdown, the top 10 census tracts with the highest average speed ranged from (38.9 mph – 59.5 mph), averaging 46.4 mph across the 10 tracts. After the lockdown, the top 10 tracts ranged from (56.0 mph – 70.5 mph), averaging 64.9 mph across the 10 tracts.

While the shape of the distributions is similar, the average speed in the highest tracts went way up after the lockdown. This is surprising to me, because I would have expected high speeds to be associated with being in a rush to get somewhere, but almost all schools, places of work, stores, etc. were shut down. However, I can also see how if fewer total cars were out on the roads, people's driving speeds aren't limited by the density of traffic around them, which could explain higher average speeds.

Plot the histogram

```
In [ ]: plt.hist(averages_post)
        plt.xlabel("Average Speed Within Census Tract (mph)")
        plt.ylabel("Number of Census Tracts")
        plt.title("Hisogram of Bay Area Census Tracts Average Speed Post Lockdown");
```

0.2.3 2.a.iii. Sort census tracts by change in traffic speed from pre to post lockdown.

For each segment, compute the difference between the pre-lockdown average speed (March 1 - 13) and the post-lockdown average speed (March 14 - 31). **Plot a histogram of all differences.** Sanity check that the below histogram matches your observations of the histograms above, on your own.

In [ ]:
```
# The autograder expects differences to be a series object with index
# MOVEMENT_ID.
diff = averages_pre_named.merge(averages_post_named, left_on = "DISPLAY_NAME",right_on = "DISPL
diff['differences'] = diff['speed_mph_mean_y'] - diff['speed_mph_mean_x']
differences = diff['differences']
# plot the differences
plt.hist(differences);
plt.xlabel("Difference in Average Speed between Pre-Lockdown and Post-Lockdown")
plt.ylabel("Number of Census Tracts")
plt.title("Differences in Average Speeds Before and After Lockdown");
```

In [ ]:
```
grader.check("q2aiii")
```

0.2.4 2.a.iv. Quantify the impact of lockdown on average speeds.

1. **Plot the average speed by day, across all segments**. Be careful not to plot the average of census tract averages instead. Recall the definition of segments from Q1.
2. Is the change in speed smooth and gradually increasing? Or increasing sharply? Why? Use your real-world knowledge of announcements and measures during that time, in your explanation. You can use this list of bay area COVID-related dataes: https://abc7news.com/timeline-of-coronavirus-us-covid-19-bay-area-sf/6047519/

In [ ]:
```
# Autograder expects this to be a series object containing the
# data for your line plot -- average speeds per day.
speeds_daily = speeds_to_tract.groupby('day').agg(np.mean)['speed_mph_mean']
plt.plot(speeds_daily)
plt.xlabel("Day in March 2020")
plt.ylabel("Average Speed Across All Segments")
plt.title("Average Speed Across all Nodes in March 2020");
```

Write your written answer in the cell below

The change in speed is increasingly sharpy, specifically accelerating around March 17th. This is likely because on March 17, shelter in place went into effect in almost all Bay Area counties, notably San Francisco and Alameda counties being relevant to this dataset.

Shelter in place caused most schools, places of work, stores, etc. to close down, meaning people didn't have anywhere to go. This likely caused decreases in traffic volume across the board, which would explain the sharp increase in average traffic speeds, because people's driving speeds were not limited by congestion and traffic density.

0.2.5 2.a.v. Quantify the impact of pre-lockdown average speed on change in speed.

1. Compute the correlation between change in speed and the *pre*-lockdown average speeds. Do we expect a positive or negative correlation, given our analysis above?
2. Compute the correlation between change in speed and the post-lockdown average speeds.
3. **How does the correlation in Q1 compare with the correlation in Q2?** You should expect a significant change in correlation value. What insight does this provide about traffic?

Written answers in the first cell, coding answerts in the following cell.

We would expect the correlation between change in speed and pre-lockdown average speeds to be positive, because it makes sense that roads with high speeds before the lockdown also increased in speed post-lockdown. As traffic speeds generally increased across the board, roads such as highways with already high "pre-lockdown speeds" likely freed up even more, shown by a positive "change in speed".

For a similar reason, we would expect the correlation between change in speed and post-lockdown average speeds to be positive, because roads with high "post-lockdown speeds" likely saw a large increase in speed as compared to pre-lockdown (positive change in speed).

After computing both correlation coefficients, we see that both correlations are positive, as expected, but post-lockdown speed is much higher correlated with change in speed (0.79 vs. 0.46). This is interesting, as it tells us that, looking at all the roads post lockdown, slower streets did not increase very much, while higher-speed streets increased the most.

This is likely because the roads with the highest capacity for speeds such as highways, saw the biggest increase in speeds. This makes sense because, pre-lockdown, highways might not necessarily have high average speeds, depending on how much traffic congestion they normally get. Then, when the lockdown decreased traffic congestion overall, these roads with high capacities for speed saw the biggest increase in speeds, as well as having a high post-lockdown average speed.

In other words, the lockdown made it so traffic speeds better reflect speed limits. As such, high speed-limit roads, such as highways, saw the biggest increases in speeds because they have the capacity for high-speed traffic. Similarly, streets with low post-lockdown average speeds did not increase by much, likely because their low speeds weren't due to traffic congestion, but instead by low speed limits, such as roads in school zones, steep hills, etc.

All of this generates useful insights about real traffic phenomena, as we are putting together more detailed "profiles" for each road and how they are affected differently by something like a county-wide lockdown.

0.2.6 2.b.i. Visualize spatial heatmap of average traffic speed per census tract, pre-lockdown.

Visualize a spatial heatmap of the grouped average daily speeds per census tract, which you computed in previous parts. Use the geopandas chloropleth maps. **Write your observations, using your visualization, noting down at least 2 areas or patterns of interest**. These may be a local extrema, or a region that is strangely all similar.

Hint: Use `to_crs` and make sure the `epsg` is using the Pseudo-Mercator projection.

Hint: You can use `contextily` to superimpose your chloropleth map on a real geographic map.

Hint You can set a lower opacity for your chloropleth map, to see what's underneath, but be aware that if you plot with too low of an opacity, the map underneath will perturb your chloropleth and meddle with your conclusions.

Written answers in the first cell, coding answers in the second cell.

The region near the SFO airport increased the most in speed. This makes sense because, after the lockdown, there weren't many people traveling using airplanes, meaning highways and other roads near SFO became much less congested. Due to this, traffic speeds increased drastically.

Another interesting pattern we noticed was that the average speed in downtown San Fransisco did not have a drastic change before and after the lockdown. This seemed quite strange due to the fact that downtown is considered an extremely crowded region, which made me assume that after the lockdown there should be much less people traveling downtown. I would assume that post lockdown, the speed of traffic should have changed drastically. That said, these roads might have other conditions that slow down traffic other than density of traffic. Steep hills, narrow roads, low speed limits, etc., likely kept traffic speeds low even when there wasn't much congestion post-lockdown.

```
In [ ]: fig, ax =plt.subplots(1, 1)
        avgs_pre_heatmap = gpd.GeoDataFrame(averages_pre_named).to_crs(3857)
        avgs_pre_heatmap.plot('speed_mph_mean', ax = ax, legend = True)
        cx.add_basemap(ax, crs = avgs_pre_heatmap.crs.to_string(), source = cx.providers.Stamen.TonerLi
```

0.2.7 2.b.ii. Visualize change in average daily speeds pre vs. post lockdown.

Visualize a spatial heatmap of the census tract differences in average speeds, that we computed in a previous part. **Write your observations, using your visualization, noting down at least 2 areas or patterns of interest.** Some possible ideas for interesting notes: Which areas saw the most change in average speed? Which areas weren't affected? Why did some areas see *reduced* average speed?

First cell is for the written answers, second cell is for the coding answers.

One interesting pattern I noticed from the plot was that downtown San Francisco had the lowest average speed, which was close to 20 mph. This makes sense because downtown San Francisco is the most populated place with the most traffic, so the average speed during pre-lockdown should be slower than post-lockdown.

Another observation is it looks like the area around Pacifica, the south-west most region on the heatmap, actually saw a reduced average speed after the lockdown. This could be because that area is home to a huge number of hiking trails, beaches, and other outdoor activities that seemingly saw an increase in traffic after the lockdown. This is likely because people were extremely limited in activities, and being outdoors was one of the only ways people could spend their time outside of their home.

In [ ]:
```
fig, ax =plt.subplots(1, 1)
speed_change_heatmap = gpd.GeoDataFrame(diff, geometry = 'geometry_x').to_crs(3857)
speed_change_heatmap.plot('differences', ax = ax, legend = True)
cx.add_basemap(ax, crs = avgs_pre_heatmap.crs.to_string(), source = cx.providers.Stamen.TonerLi
```

0.2.8 4.a.ii. Train and evaluate linear model on pre-lockdown data.

1. **Train a linear model that forecasts the next day's speed average** using your training dataset `X_train`, `y_train`. Specifically, predict $y_{(i,t)}$ from $X_{(i,t)}$, where

- $y_{(i,t)}$ is the daily speed average for day t and census tract i
- $X_{(i,t)}$ is a vector of daily speed averages for days $t-5, t-4, t-3, t-2, t-1$ for census tract i

2. **Evaluate your model** on your validation dataset `X_val`, `y_val`.
3. **Make a scatter plot**, plotting predicted averages against ground truth averages. Note the perfect model would line up all points along the line $y = x$.

Our model is quantitatively and qualitatively pretty accurate at this point, training and evaluating on pre-lockdown data.

In [ ]:
```
reg = LinearRegression().fit(X_train, y_train) # set to trained linear model
score = reg.score(X_val, y_val) # report r^2 score
predict = reg.predict(X_val)
# create the scatter plot below
plt.scatter(predict, y_val)
plt.xlabel('Predicted Averages')
plt.ylabel('Ground Truth Averages')
plt.title('Predicted Averages Against Ground Truth Averages');
```

Make scatter plot below.

```
In [ ]: predict = reg.predict(time_series_x_pre)
        # create the scatter plot below
        plt.scatter(predict, time_series_y_post)
        plt.xlabel('Predicted Averages')
        plt.ylabel('Ground Truth Averages')
        plt.title('Predicted Averages Against Ground Truth Averages');
```

0.2.9 4.b.ii. Report model performance temporally

1. **Make a line plot** showing performance of the original model throughout all of March 2020.
2. **Report the lowest point on the line plot**, reflecting the lowest model performance.
3. **Why is model performance the worst on the 17th?** Why does it begin to worsen on march 15th? And continue to worsen? Use what you know about covid measures on those dates. You may find this webpage useful: https://abc7news.com/timeline-of-coronavirus-us-covid-19-bay-area-sf/6047519/
4. **Is the dip in performance on the 9th foreshadowed** by any of our EDA?
5. **How does the model miraculously recover on its own?**
6. **Make a scatter plot**, plotting predicted averages against ground truth averages *for model predictions on March 17th*. Note the perfect model would line up all points along the line $y = x$. When compared against previous plots of this nature, this plot looks substantially worse, with points straying far from $y = x$.

Note: Answer questions 2-5 in the Markdown cell below. Q1 and Q6 are answered in the two code cells below.

2.The lowest point on the line plot seems to be on the 17th day in March with a model score below 0.75.

3.Given that we know that covid lockdowns began on March 14th, the model performance begins to worsen before March 15th. This is because our model is predicting 5 days prior to the present day. The models performance is the worst on the 17th because the model we are using to predict uses the traffic speeds of the last 5 days and because the shutdown was on the 14ths, 2 of the speeds included were prior to the shutdown, 2 of them were post shutdown and one of them was the exact day of the shutdown. Therefore, this model was a bad predictor of model performance temporally.

4.The dip in performance on the 9th is foreshadowed by our EDA because our prediction is predicting 5 days prior to the actual day. The covid lockdown began on March 14th, and March 9th is 5 days before the actual covid lockdown, which means that our EDA did foreshadow this dip in the model performance.

5.The model miraculously recovers on its own because as we get further in the month and the model uses the 5 day prior method, the 5 days before a later day in the month would use only data from post-lockdown which makes it have a better model performance.

Generate line plot.

```
In [ ]: array = []
        for i in np.arange(0, 25):
            x_train_b = time_series.iloc[:, i:i+5].to_numpy()
            y_train_b = time_series.iloc[:, i+5:i+6].to_numpy()
            x_train_b, y_train_b = remove_nans(x_train_b, y_train_b)
            array = np.append(array, reg.score(x_train_b, y_train_b))
        plt.plot(np.arange(6, 31), array)
        plt.xlabel("Day in March")
        plt.ylabel("Model Score")
        plt.title("Model Score vs. Day in March");
```

Generate a scatter plot.

```
In [ ]: X, y = remove_nans(time_series.iloc[:, 11:16].to_numpy(), time_series.iloc[:, 16:17].to_numpy()
        predict = reg.predict(X)
        plt.scatter(x = predict, y = y)
        plt.xlabel("Predicted Averages")
        plt.ylabel("Ground Truth Averages")
        plt.title("Predicted Averages Against Ground Truth Averages");
```

0.2.10 4.c.i. Learn delta off of a moving bias

According to our previous work in EDA, the average speed shoots upwards sharply. As a result, our trick to learn delta the around the average and to naively assume that the average of day t is the average for day $t + 1$. We will do this in 4 steps:

1. **Create a dataset for your delta model**.
2. **Train your delta model** on pre-lockdown data.
3. **Evaluate your model on pre-lockdown data**, to ensure that the model has learned to a satisfactory degree, in the nominal case. Remember the naive model achieved 0.97 r^2 on pre-lockdown data.
4. **Evaluate your model on the 17th**, to compare against the naive model also evaluated on that day. Notice that your r^2 score has improved by 10%+. Why is your delta model so effective for the 17th?
5. **Evaluate your model on the 14th**, to compare against the naive model also evaluated on that day. Notice that your r^2 score is now complete garbage. Why is your delta so ineffective for the 14th?

Hint: As you build your datasets, always check to make sure you're using the right days! It's easy to have a one-off error that throws off your results.

Write your written questions in the next cell, then write the code in the following cells.

The delta model is so effective for the 17th because we used the 5 day prior model which has the speeds of pre-lockdown, the actual lockdown date and post-lockdown, which was a great way to predict a given day post-lockdown.

The delta model is so ineffective for the 14th because this was the exact day of the lockdown and if we still used the 5 day prior model, the 5 days before the 14ths were all pre-lockdown, so it would not have predicted a massive drop due to the lockdown on that day. In other words, because we used 5 days that were pre-covid lockdown to predict a given date(the given date being the exact day of the shutdown), it would not have predicted this happening.