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.

Type your answer here, replacing this text.

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":

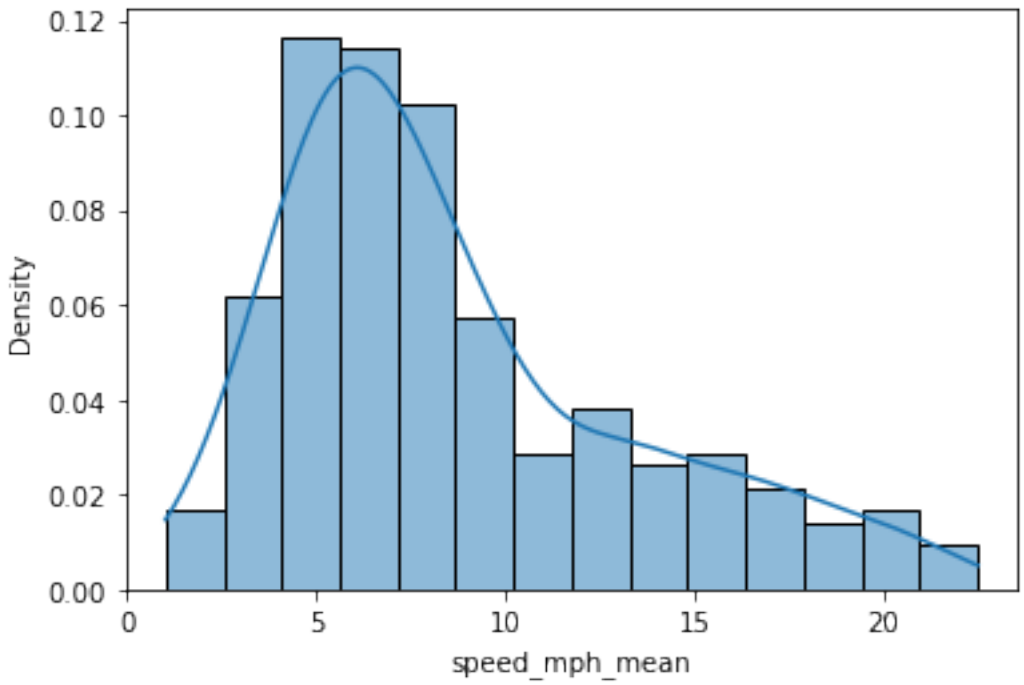
1. **Plot a histogram of the within-cluster standard deviation**.
2. **Compute across-cluster average of within-cluster standard deviation**.
3. **Compute across-cluster standard deviation of within-cluster average speeds**.
4. **Is this average variance reasonable?** To assess what "reasonable" means, 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.

Type your answer here, replacing this text.

In [15]:
```
import seaborn as sns
speed_variance_by_pluscode = speeds_to_gps.groupby(['plus_latitude_idx','plus_longitude_idx'])
# plot a histogram
#speed_variance_by_pluscode.plot.hist(bins=12, alpha=0.5)
sns.histplot(speed_variance_by_pluscode, kde = True, stat = 'density');
average_variance_by_pluscode = np.mean(speed_variance_by_pluscode)
variance_average_by_pluscode = speeds_to_gps.groupby(['plus_latitude_idx','plus_longitude_idx'])
```



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":

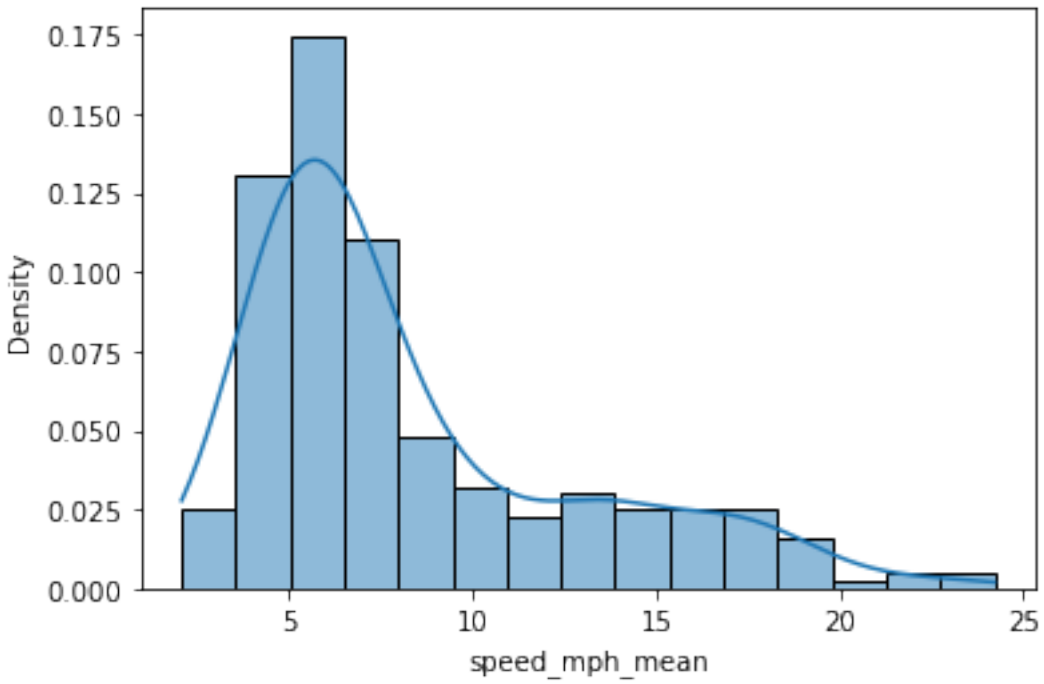
1. **Plot a histogram of the within-cluster standard deviation**.
2. **Compute across-cluster average of within-cluster standard deviation**.
3. **Compute across-cluster standard deviation of within-cluster average speeds**.
4. **Is this average variance reasonable?** To assess what "reasonable" means, 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.

Type your answer here, replacing this text.

In [24]:
```
speed_variance_by_tract = speeds_to_tract.dissolve(by = 'DISPLAY_NAME', aggfunc=pd.Series.std)
sns.histplot(speed_variance_by_tract, kde = True, stat = 'density');
average_variance_by_tract = np.mean(speed_variance_by_tract)
variance_average_by_tract = speeds_to_tract.groupby(['DISPLAY_NAME']).agg(np.mean)['speed_mph_
```



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' metric for a spatial structure is entirely dependent on what you are trying to model. For our dataset, a 'good' metric can be defined as one that is efficient as a representative of the subpopulations of the area that is being analyzed. A 'bad' metric for a spatial structure is one that is not representative of the subpopulation that we are concerned with. For example, Uber utilizes the H3 model. This model divides an area with a global grid system. However, unlike the Google Plus Codes System, the Uber H3 system divides an area with a hexagonal global grid system and a hierarchical indexing system. For Uber this is beneficial in city areas because it allows for easy application of surge prices in overcrowded areas as there is constant movement in cities and urban areas. Through this hexagonal structure, we are able to have access to specific subsets of areas, and compare one to another. For this example, a good metric would be a small standard deviation as this shows a small dispersion of data and a better representation of the population. Another model to visualize the data would be to use geography and census data like the 'traffic analysis zone'. This model creates zones based on census block information, thus allowing for a better representation of the subpopulations we are trying to model after. Thus, we would optimize our own metric for success in a spatial structure by ensuring that there is little to no dispersion among the data within and between each space in the spatial structure, as well as making sure that the data is representative of the subpopulation we are concerned with.

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.

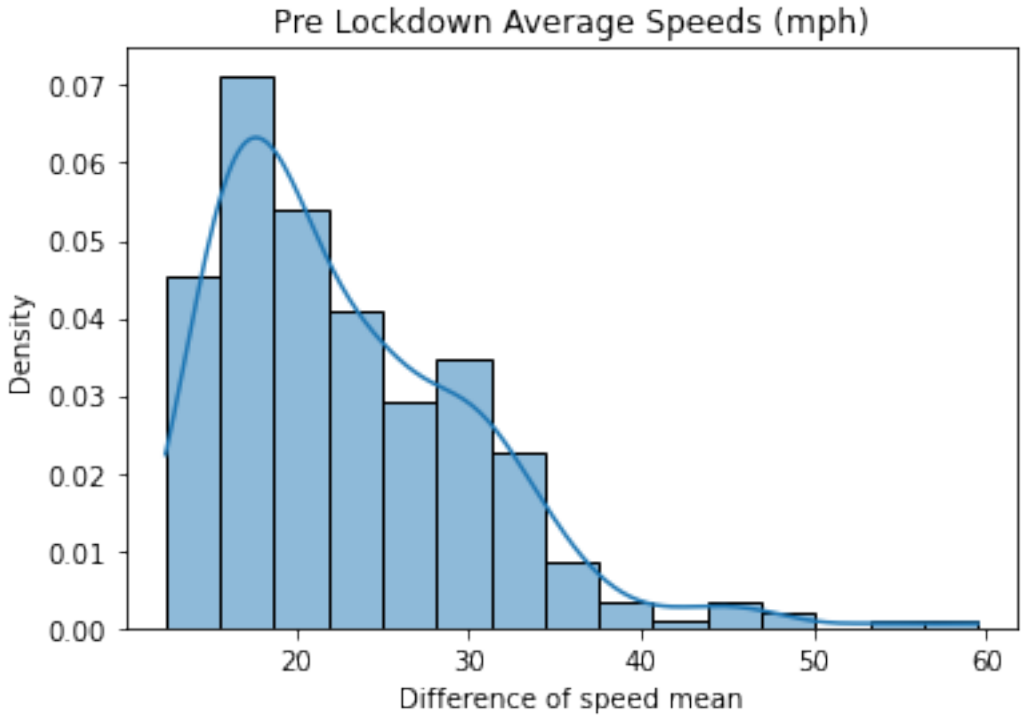
1. **Report a DataFrame which includes the *names* of the 10 census tracts with the lowest average speed**, along with the average speed for each tract.
2. **Report a DataFrame which includes the *names* of the 10 census tracts with the highest average speed**, along with the average speed for each tract.
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 that we found to have the highest and lowest average speeds pre-lockdown matched our expectations for the low and high speed traffic zones. This is because we expected most of the high-speed areas to be outside of San Francisco and in areas that do not have a lot of vehicles going through them throughout the day. On the other hand, we thought that the tracts with the lowest speeds would likely be in San Francisco, especially in the heart of the city, where many people either drive to go to work or tour popular spots. These relationships were present in our results, matching our expectations for the speed of the tracts. Our hypothesis for what the long tail distribution of high speed traffic corresponds to in San Francisco is that most of the areas in the city have a lot of vehicles on the road, so the majority of speeds are on the lower end. However, there are certain areas such as highways entering the city that have higher average speeds than the inner roads, resulting in a few datapoints with higher speeds, resulting in the creation of the long tail distribution.

Plot the histogram

```
In [33]: pre_graph = sns.histplot(averages_pre, kde = True, stat = 'density');
         pre_graph.set(xlabel='Difference of speed mean', title='Pre Lockdown Average Speeds (mph)');
```



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:

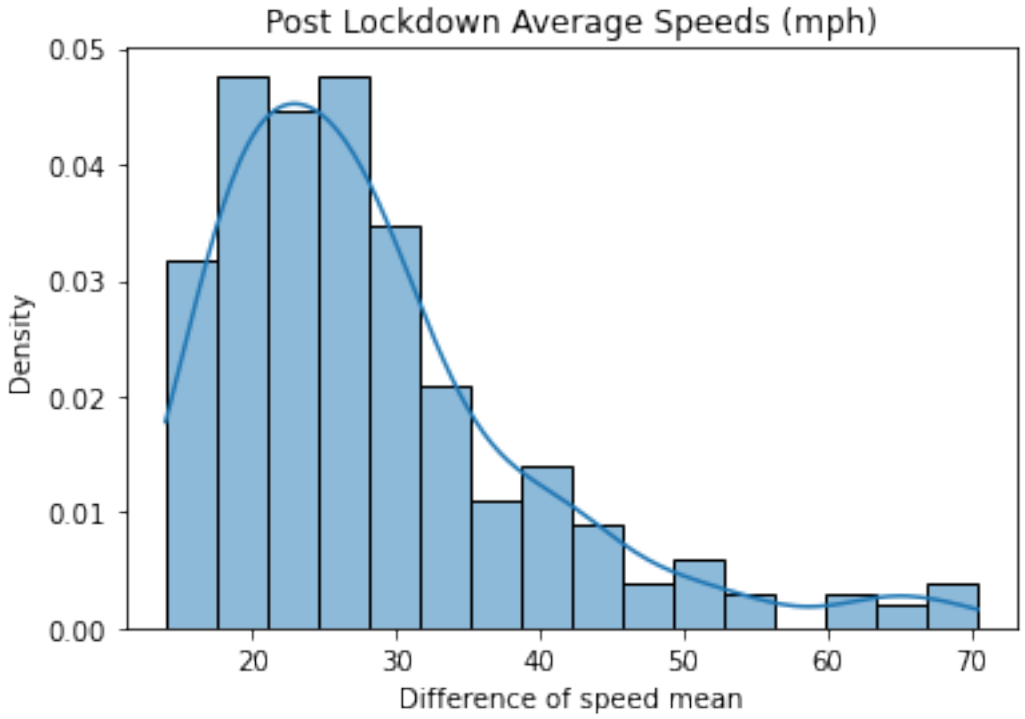
- **Plot a histogram for all average speeds, post-lockdown.**
- **What are the major differences between this post-lockdown histogram relative to the pre-lockdown histogram above**? Anything surprising? What did you expect, and what did you find?

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

There are several differences between the post-lockdown histogram and the pre-lockdown histogram of average speeds. The post-lockdown histogram shows that although the data is still skewed to the right, there is a more even distribution of speeds after the lockdown (the density of mean speeds in the bins tends to be more spread out than before, with the highest density being around 0.05 post-lockdown as compared to 0.07 pre-lockdown). Additionally, the fastest speed post-lockdown is higher than that from the highest speed pre-lockdown (up to 70 mph rather than just a maximum of up to 60 mph). Finally, there seems to be a higher median average speed post-lockdown than pre-lockdown, which makes sense because there are less people traveling after the lockdown. Something that was surprising about the histograms is that the data post-lockdown still has a skewed-right distribution even though there intuitively should not have been many cars on the road. The majority differences in the histogram aligned with our expectations that the average speeds will increase after the lockdown because most people will not be traveling, allowing there to be less traffic and the few people that are traveling to drive faster than before.

Plot the histogram

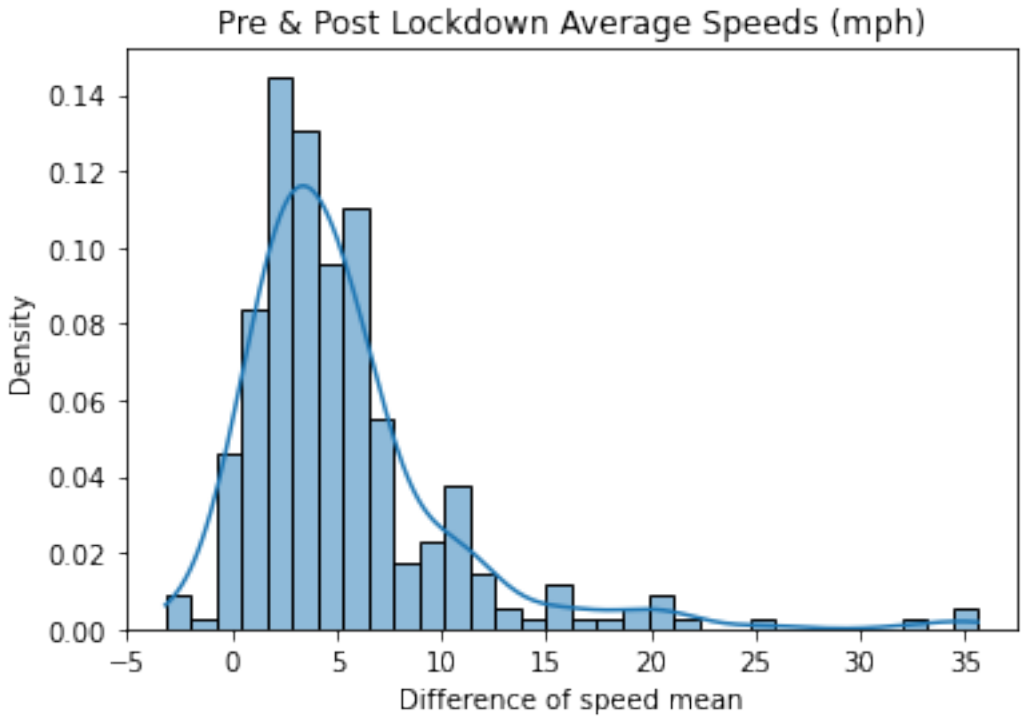
```
In [36]: st = sns.histplot(averages_post, kde = True, stat = 'density');
         st.set(xlabel='Difference of speed mean', title='Post Lockdown Average Speeds (mph)');
```



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 [37]:
```
# The autograder expects differences to be a series object with index
# MOVEMENT_ID.
averages_pre_post_df = averages_pre_named.merge(averages_post_named, how='inner', left_on='DIS
differences = averages_pre_post_df['speed_mph_mean_y'] - averages_pre_post_df['speed_mph_mean_
differences

# plot the differences
ax = sns.histplot(differences, kde=True, stat='density');
ax.set(xlabel='Difference of speed mean', title='Pre & Post Lockdown Average Speeds (mph)');
```



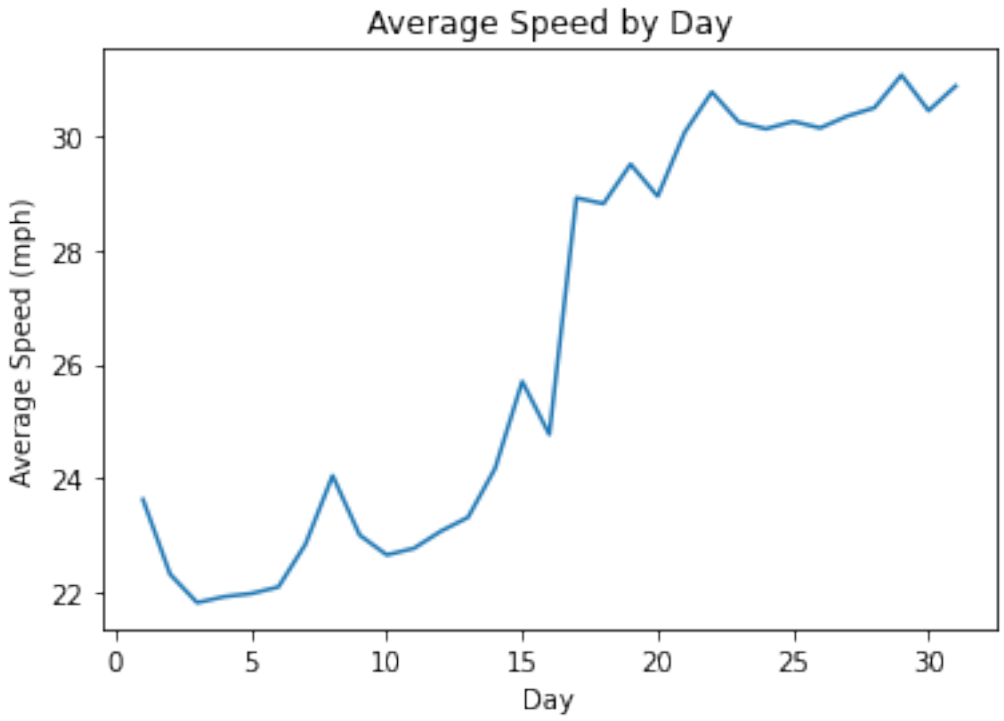
In [38]:
```
grader.check("q2aiii")
```

Out[38]:
```
q2aiii results: All test cases passed!
```

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 [39]:
```
# 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']).mean()['speed_mph_mean']
speeds_daily.plot();
plt.xlabel('Day');
plt.ylabel('Average Speed (mph)')
plt.title('Average Speed by Day');
```



Write your written answer in the cell below

The change in speed is not smooth and gradually increasing; rather, it is sharp and suddenly increases around days 14-16. This is because the lockdown initially goes into effect around day 14, so many people stop traveling to work. Then, on day 16, many counties ban students from going to classes in-person, so the traffic associated with students going to school is also eliminated, causing a sharp spike in the average speed after that day. The rest of the data is relatively constant and increases mostly smoothly because no major policy changes or COVID-related announcements happen for the rest of the month.

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 expect a positive correlation between change in speed and the pre-lockdown average speeds because average speeds increase overall after the lockdown went into effect. The correlation in Q1 is a lot lower than the correlation in Q2. This is because using census plots is not the most effective way of understanding changes in speed overall. Because lots of different types of roads/areas can be covered in one census plot, it may be harder to understand the changes in traffic based on what the speed was like before the lockdown. This insight shows that speeds pre-lockdown could be used to evaluate the traffic that would result post-lockdown because highly trafficked areas before the lockdown were still more trafficked than others after the lockdown even though there was a drop in traffic overall.

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.

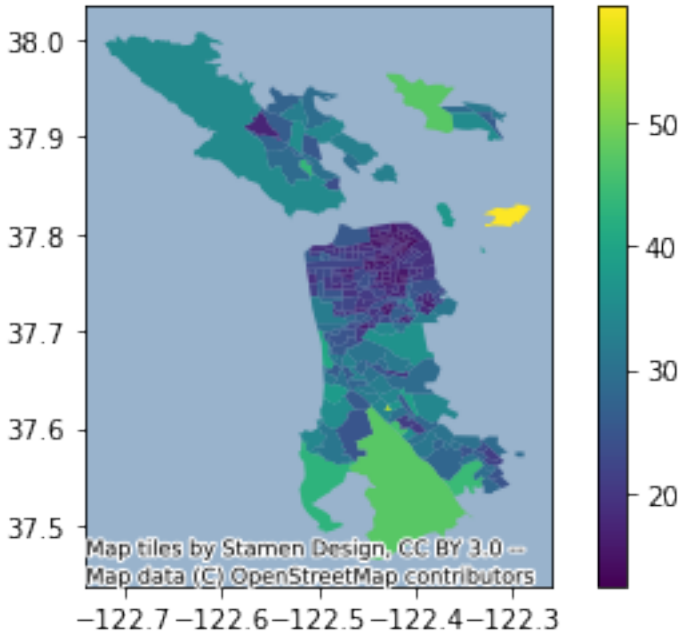
Hint: You can use `contextily` to superimpose your chloropleth map on a real geographic map.

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.

From the spatial heatmap of the grouped average daily speeds per census tract, we can see that downtown San Francisco has some of the lowest speeds in the areas. This is likely because most buildings, tourist attractions, and offices are located in that part of the city, so there is more traffic and therefore lower average speeds. Additionally, we can also see that most of the larger regions in the map have higher average speeds than the smaller regions. This is probably because the areas with higher speeds have fewer people (which is why the population-based regions are larger), so the traffic is lower and the speeds of vehicles will be higher.

In [43]:
```
avg_pre_geo = gpd.GeoDataFrame(averages_pre_named)
avg_pre_geo = avg_pre_geo.to_crs(epsg = 4326)
ad = avg_pre_geo.plot(column='speed_mph_mean', legend = True);
cx.add_basemap(ad, zoom = 16)
```



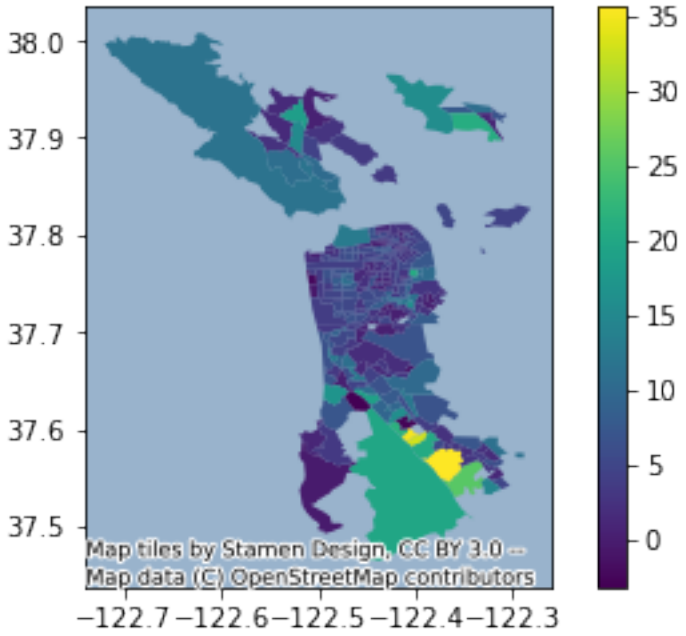
0.2.7 2.b.ii. Visualize change in average daily speeds pre vs. post lockdown.

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.

From the spatial heatmap of the differences in average speeds for each census tract, we can see that almost all areas saw an increase in average speeds after the lockdown. The areas that saw the most change were the regions that are not located in downtown San Francisco, so there were likely very few people going there at all due to travel or tourism-related reasons, causing an increase in the average speeds. Additionally, although there was a small increase in average speeds in the most densely populated and trafficked parts of the city, these regions were not affected very much. This is probably because there were still essential workers that had to travel to work throughout the lockdown, preventing the traffic levels from decreasing significantly. Finally, some areas might have seen reduced average speeds. This might be because these are residential areas so there are more people driving around near their homes and at higher frequencies than they used to before the lockdown, causing there to be more traffic in these specific census tracts.

In [44]:
```
averages_pre_post_df['differences'] = differences
averages_pre_post_df
avg_pre_post_geo = gpd.GeoDataFrame(averages_pre_post_df.loc[:, ['DISPLAY_NAME', 'differences'
ad_post = avg_pre_post_geo.plot(column='differences', legend = True);
cx.add_basemap(ad_post, zoom = 16)
```



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

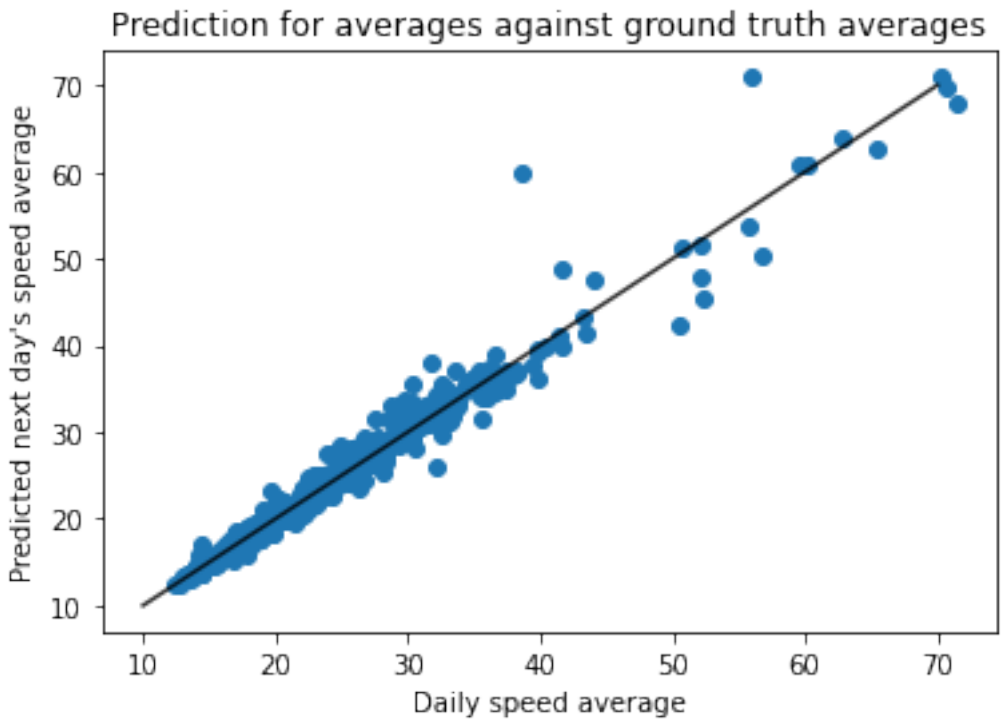
- $y_{(i,t)}$ is the daily speed average for day t and census tract i
- $X_{(i,t)}$ is a vector of daily speed averages for days $t-5, t-4, t-3, t-2, t-1$ for census tract i

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.

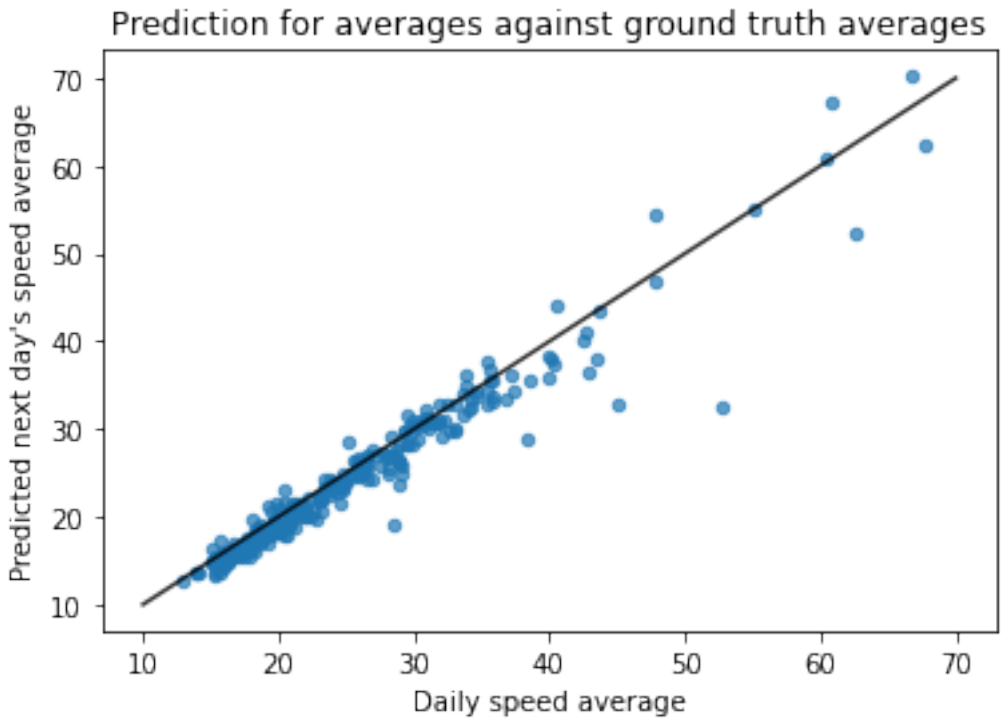
In [64]:
```
model = LinearRegression(fit_intercept = True)
reg = model.fit(X_train, y_train) # set to trained linear model
score = reg.score(X_val, y_val) # report r^2 score

# create the scatter plot below
fitted_values = model.predict(X_val) #X_val
plt.scatter(y_val, fitted_values)
plt.xlabel('Daily speed average')
plt.ylabel("Predicted next day's speed average")
plt.title("Prediction for averages against ground truth averages")
plt.plot([10,70] , [10,70], 'k-', alpha = 0.80)
```

Out[64]:
```
[<matplotlib.lines.Line2D at 0x7faf0c4b3b20>]
```


Prediction for averages against ground truth averages
Predicted next day's speed average
Daily speed average
10
20
30
40
50
60
70

Make scatter plot below.

```
In [71]: fitted_values = model.predict(x_pre_nan) #X_val
         plt.scatter(y_post_nan, fitted_values, s= 20, alpha = 0.7);
         plt.xlabel('Daily speed average')
         plt.ylabel("Predicted next day's speed average")
         plt.title("Prediction for averages against ground truth averages")
         plt.plot([10,70] , [10,70], 'k-', alpha = 0.80);
```



0.2.9 4.b.ii. Report model performance temporally

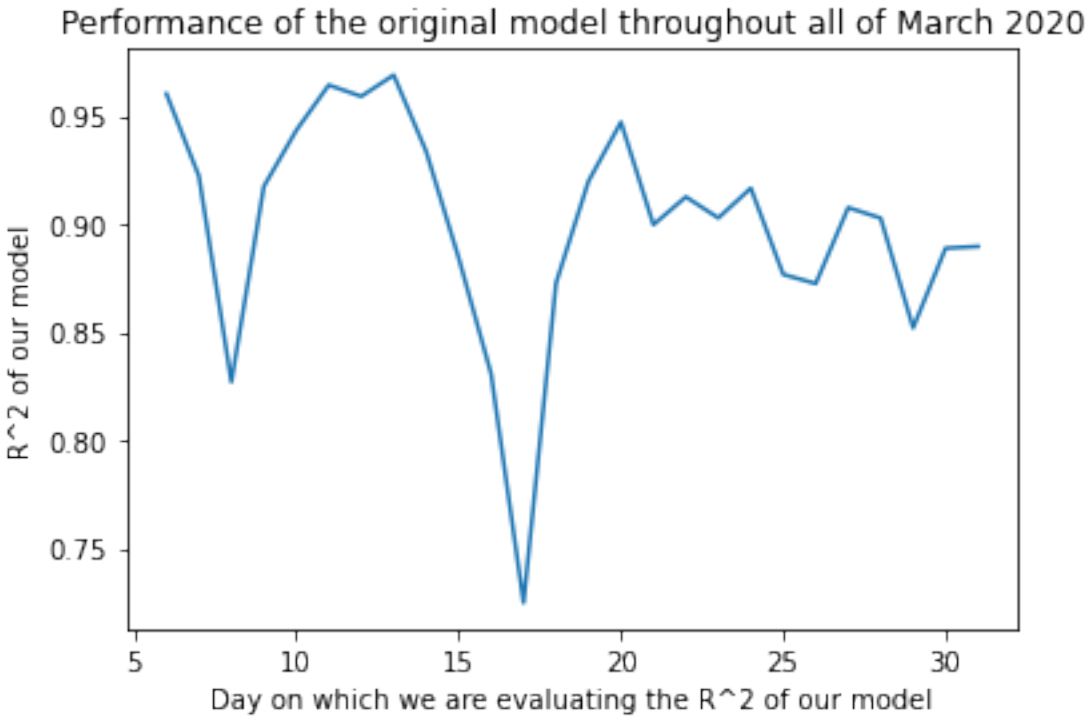
1. **Make a line plot** showing performance of the original model throughout all of March 2020.
2. **Report the lowest point on the line plot**, reflecting the lowest model performance.
3. **Why is model performance the worst on the 17th?** Why does it begin to worsen on march 15th? And continue to worsen? 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.

The lowest point on the line plot is March 17, 2020, which reflects the lowest model performance. The model performance is worst on the 17th because the 17th was the day that the COVID-19 shelter in place order took effect in six Bay Area counties, including San Francisco county. In terms of our dataset, the shelter in place order restricted San Francisco residents to only be able to perform essential activities. As a result, Uber activity was restricted, as less people were ordering and taking Ubers, and more people were electing to protect themselves and stay at home. This is reflected in the dip in model performance on the line plot. The performance of the model initially begins to worsen on March 15, 2020 because the 15th marks when Governor Newsom announced the closure and restriction of bars, restaurants, and nightclubs. He also advised people over 65 and those with health conditions to stay at home. As a result of these initial restrictions, the amount of people taking Uber began to decrease, and thus the performance of the model began to worsen. It continues to worsen as more restrictions are placed, and more deaths are reported, thus causing less people to utilize Uber. The dip in model performance on the 9th is also foreshadowed in many other visuals in this project. For example, in 2aiii in part 1, we are asked to plot the difference between the pre lockdown average speed and the post lockdown average speed. In the histogram, the difference of average speed in terms of density is shown to have a major decrease post lockdown (March 14-30) compared to pre lockdown (March 1-13) speeds. This decrease is shown to start around March 9th, as restrictions began to be implemented at around this time. This visualization can be used as evidence and foreshadowing of a dip of model performance on March 9th. The model recovers on its own after some time has passed because it is able to account for the changes in traffic and Uber-ride activity after the lockdown goes into effect. After seeing the changes in the number of rides overall, we can make better predictions about future rid usage as well given the fact that the model now has accurate post-lockdown data to base its predictions on, unlike before, when the model was just using pre-lockdown data that did not account for the effects of the lockdown on rides and traffic.

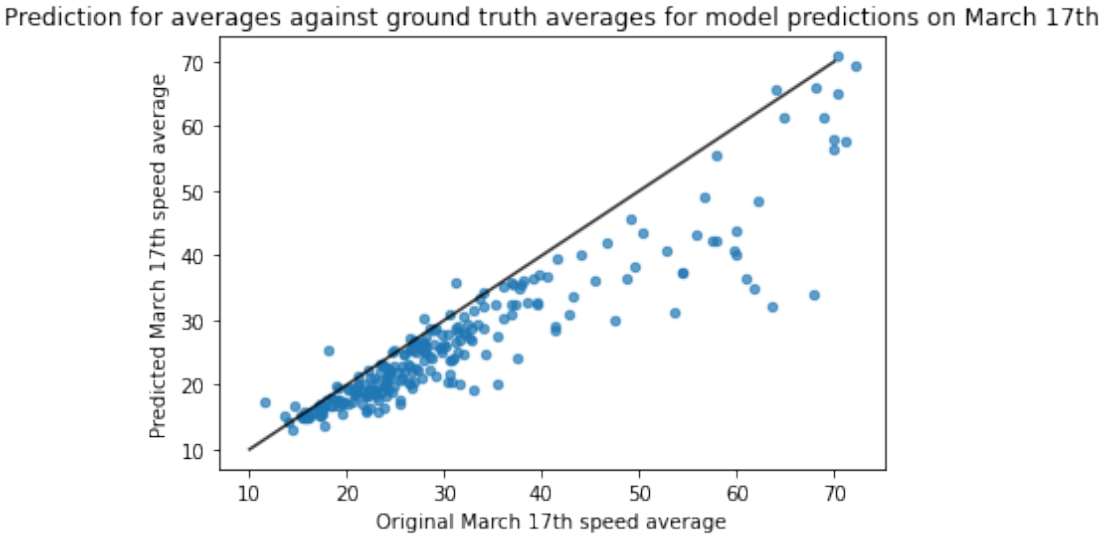
Generate line plot.

```
In [72]: r2=[]
         ld=[]
         for i in range(0,26):
             X_train1 = time_series.iloc[:,range(i, i+5)].to_numpy()
             y_test1 = time_series.iloc[:,i+5].to_numpy()
             X_train1, y_test1 = remove_nans(X_train1, y_test1)
             score = reg.score(X_train1, y_test1)
             r2.append(score)
             ld += [i + 6]

         plt.plot(ld, r2);
         plt.xlabel('Day on which we are evaluating the R^2 of our model');
         plt.ylabel('R^2 of our model');
         plt.title('Performance of the original model throughout all of March 2020');
```



Generate a scatter plot.

```
In [73]: X17_train = time_series.loc[:, 12:16]
         y17_test = time_series.loc[:, 17]

         X17_train1, y17_test1 = remove_nans(X17_train, y17_test)
         pred_val = reg.predict(X17_train1) #X_val

         plt.scatter(y17_test1, pred_val, s = 20, alpha = 0.7);
         plt.xlabel('Original March 17th speed average')
         plt.ylabel("Predicted March 17th speed average")
         plt.title("Prediction for averages against ground truth averages for model predictions on Marc
         plt.plot([10,70] , [10,70], 'k-', alpha = 0.80);
```



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 more effective for the 17th because of the model recovery that happened after the lockdown on March 16th. There was a large dip in model performance on March 16th due to the sudden decrease of people utilizing Uber's services. However, after March 16th, the model was able to make better predictions on data that was not entirely based on pre lockdown data. The model had accurate post lockdown data to base predictions on, unlike when the model was using pre lockdown data that did not account for the effects of the lockdown on rides and traffic. As a result of the improved model, the delta model was more effective for the 17th and our r^2 score improved by 10%+. Thus, the overall efficiency of the model evaluated on the 17th is a result of the model improvement and recovery that came from better predictions after lockdown were imposed on March 16th, 2020.