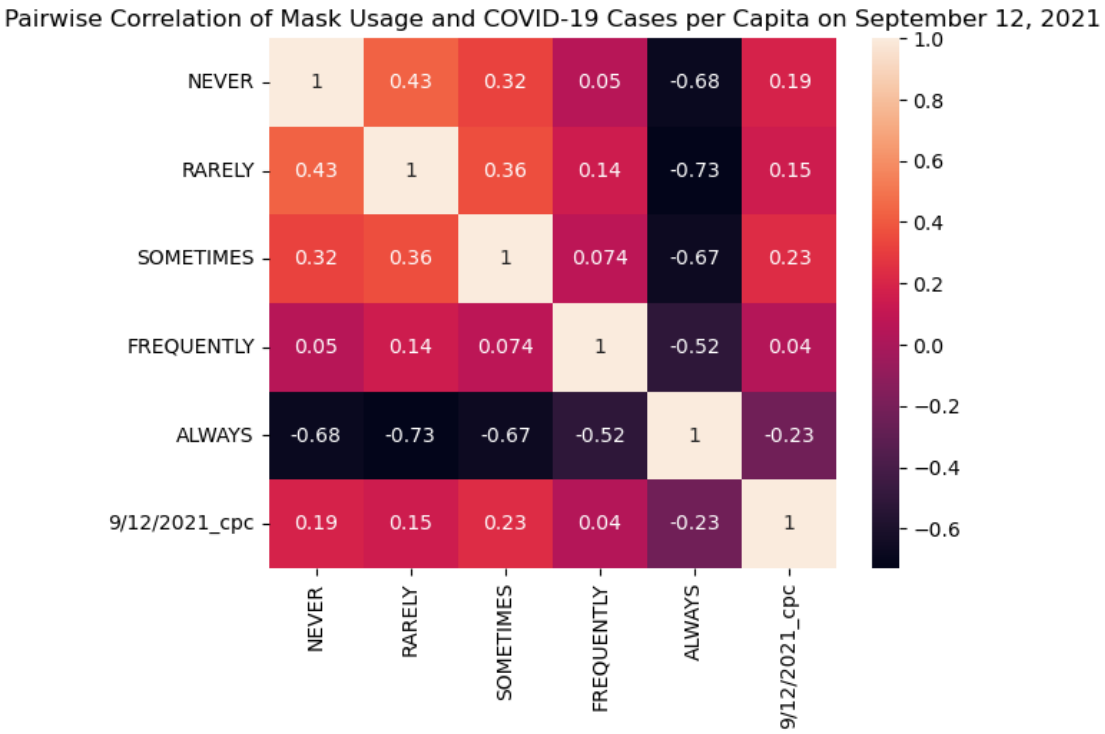
0.0.1 Question 2c

Our goal is to use county-wise mask usage data to predict the number of COVID-19 cases per capita on September 12th, 2021 (i.e., the column `9/12/2021_cpc`). But before modeling, let's do some EDA to explore the multicollinearality in these features, and then we will revisit this question in part 4.

Create a visualization that shows the pairwise correlation between each combination of columns in `mask_data`. For 2-D visualizations, consider Seaborn's heatmap. Remember to add a title to your plot.

Hint: You should be plotting 36 values corresponding to the pairwise correlations of the six columns in `mask_data`.

```
In [9]: sns.heatmap(mask_data.corr(), annot = True).set(title = 'Pairwise Correlation of Mask Usage and
```



0.0.2 Question 2d

(1) Describe the trends and takeaways visible in the visualization of pairwise correlations you plotted in Question 2c. Specifically, how does the correlation between pairs of features (i.e. mask usage) look like? How does the correlation between mask usage and cases per capita look like?

(2) If we are going to build a linear regression model (with an intercept term) using all five mask usage columns as features, what could be the problem?

1 - We observe that the correlation between mask usage and cases per capita is weak because there are no values that go above 0.3 or -0.3 correlation. From this, we can rule out mask usage for the feature to use to predict cases per capita.

2 - The problem is that using all five mask columns against one another will not give us much insight on mask usage and cases per capita. The model built from it would give us no insight and is meaningless. In addition, the correlation between mask usage columns and cases per capita is low, so they, too, may not be good features.

0.0.3 Question 3b

To visualize the model performance from part (a), let's make the following two visualizations:

(1) the predicted values vs. observed values on the test set,

(2) the residuals plot. (Note: in multiple linear regression, the residual plot has predicted values vs. residuals)

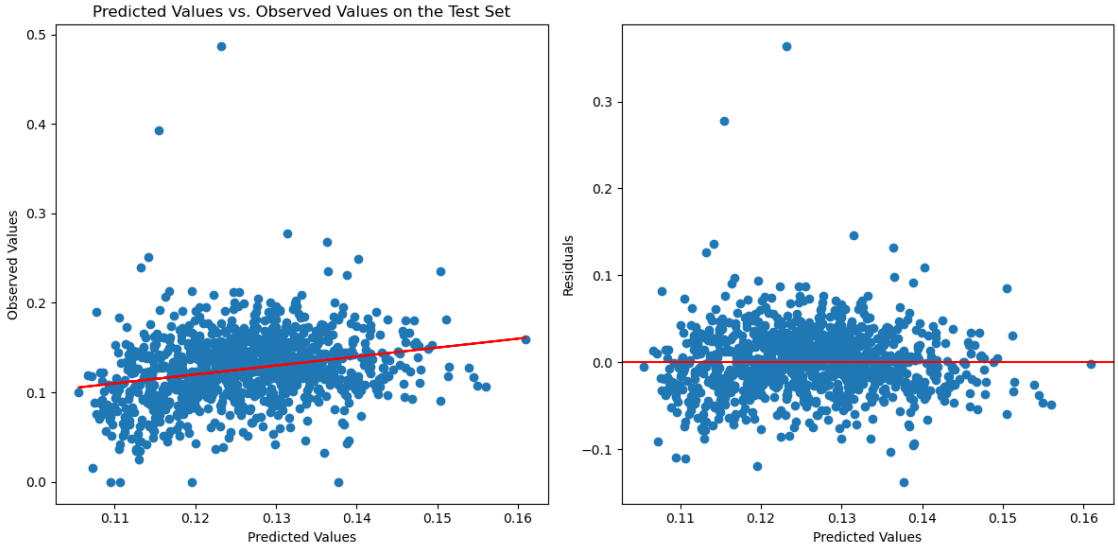
Some notes: * We've used `plt.subplot` (documentation) so that you can view both visualizations side-by-side. For example, `plt.subplot(121)` sets the plottable area to the first column of a 1x2 plot grid; you can then call Matplotlib and Seaborn functions to plot that area, before the next `plt.subplot(122)` area is set. * Remember to add a guiding line to both plot where $\hat{y} = y$, i.e., where the residual is 0. * Remember to label your axes.

```
In [12]: plt.figure(figsize=(12,6))      # do not change this line

         plt.subplot(121)                # do not change this line
         # (1) predictions vs observations
         plt.scatter(y_predict, y_test)
         plt.plot(y_predict, y_predict, color = 'red')
         plt.xlabel('Predicted Values')
         plt.ylabel('Observed Values')
         plt.title('Predicted Values vs. Observed Values on the Test Set')

         plt.subplot(122)               # do not change this line
         # (2) residual plot
         plt.scatter(y_predict, y_test - y_predict)
         plt.xlabel('Predicted Values')
         plt.ylabel('Residuals')
         plt.axhline(y = 0, color = 'red')

         plt.tight_layout()             # do not change this line
```


Predicted Values vs. Observed Values on the Test Set
Observed Values
Predicted Values
Residuals
Predicted Values

0.0.4 Question 3c

Describe what the plots in part (b) indicate about this linear model.

The plots show that our linear model is doing pretty well. The residuals appear to be centered around 0 and there aren't any distinct patterns and the values are close to y = 0.

0.0.5 Question 4d

Interpret the confidence intervals above for each of the θ_i, where θ_0 is the intercept term and the remaining θ_i's are parameters corresponding to mask usage features. What does this indicate about our data and our model?

Describe a reason why this could be happening.

Hint: Take a look at the design matrix, heatmap, and response from Question 2!

Every confidence interval has 0in it, which shows that our mask usage features aren't as effective in predicting cases per capita because they can have the weight 0 assigned to them. This could be resulted from values in our design matrix being small. This means that the weight we use for our features will also be small. In addition, the high correlated values in our heatmap indicate that the features may not be useful, which results in our model assigning small weights to them.

0.0.6 Question 5b

Comment on the ratio `prop_var`, which is the proportion of the expected square error on the data point captured by the model variance. Is the model variance the dominant term in the bias-variance decomposition? If not, what term(s) dominate the bias-variance decomposition?

Note: The Bias-Variance decomposition from lecture:

$$\text{model risk} = \sigma^2 + (\text{model bias})^2 + \text{model variance}$$

where σ^2 is the observation variance, or "irreducible error".

The prop_var has a low value, which shows that a low proportion of expected squared error was captured by the model variance. So the model variance is not the dominant term and the dominating terms may be the model bias squared or observation variance.

0.0.7 Question 5d

Propose a solution to reducing the mean square error using the insights gained from the bias-variance decomposition above.

Assume that the standard bias-variance decomposition used in lecture can be applied here.

To reduce the mean squared error using the insights gained from the bias-variance decomposition above, we can add more features to our model and help our model better fit the data because the dominant term is the model bias squared term and the model bias is an underfitting variable. So, adding more features would increase the model complexity and reduce underfitting.