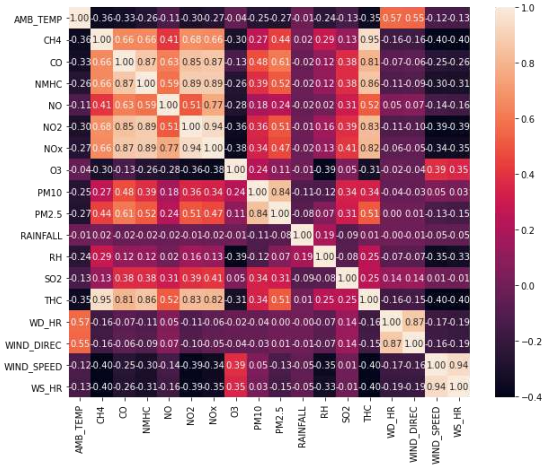
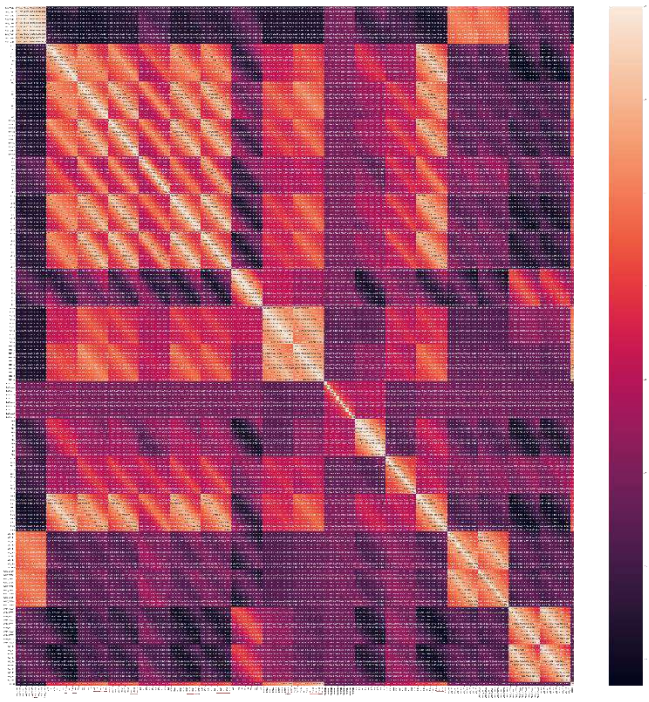
DM HW1

1. **How do you select features for your model input, and what preprocessing did you perform?**

   I compute the correlation coefficients of each column. Higher absolute values of correlation coefficients indicates that the PM2.5 is more related to those columns. Here is the coefficient correlation matrix of the columns.



   Since we are predicting the new value in the next hour given 9-hour data as input, I reformat the data to be the same as the one I will send into the model. (This image is also in /code/png)

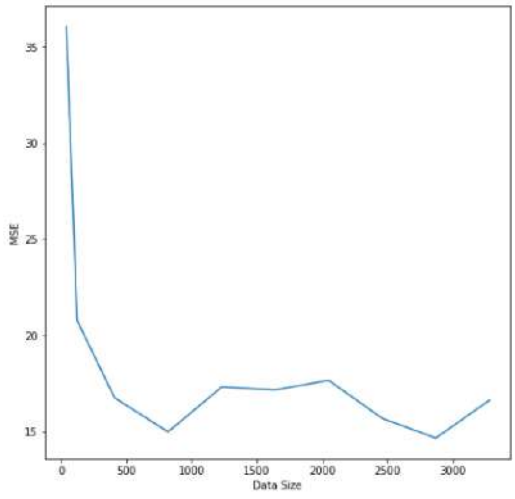


There are several potential inputs ['CO', 'PM10', 'PM2.5', 'AMB_TEMP', 'CH4', 'NMHC', 'NO2', 'NOx', 'THC'], but since there are too many missing values in both the testing set and the training set, I can not take all of them as input. Here I select three columns with not that many missing values in the testing set [‘CO’, ‘PM10’, ‘PM2.5’], more precisely, last 5 hours of PM10, PM2.5 and last 3 hours of CO.

For the training set, I fill in the nans with the average of the last and next value if they are not nan, and this helps increase the training set a lot. Then, I drop the rows that still have nan value when forming into the format of model input. For the testing set, I tried to fill in the nans with the same approach, but there are still nans, so I fill the rest with the average.

2. **Compare the impact of different amounts of training data on the PM2.5 prediction accuracy. Visualize the results and explain them.**

   Before I do this part, I used gradient descent for the weights, but it will be troublesome to check whether it converge to optimal when I must do that many time, so here I implement linear regression using the closed form solution. I append a value 1 for the input to do interpolation. I fixed the validation set to 0.2 * total dataset size, and test the training set size from 0.01, 0.03, 0.1, 0.2....0.8 * total dataset size.



I random shuffle the dataset everytime and take the average of 3 times for each datasize. This result is normal except for the datasize of 0.3 * dataset size. This might be caused by two reasons. One reason is that I filled in some values into the dataset, so it might cause the data to be inaccurate. The second is luck, maybe it will be normal if I do more trials.

3. Discuss the impact of regularization on PM2.5 prediction accuracy.
   I tried using min max scalar to nomalize the values between 0 and 1 using the following program.

```
x_max = np.max(trainingX, axis=0)
x_min = np.min(trainingX, axis=0)
trainingX = (trainingX - x_min) / (x_max - x_min)
testingX = (testingX - x_min) / (x_max - x_min)
```

   With the nomalization, we can now have L2 regularization. I add the square of the weight times 0.01 (this can be tuned) to the loss function. This stablize the accuracy between different dataset (I random shuffle the whole dataset and pick by index range, which is somehow like k-fold cross validation). The original MSE is about 14 to 18, and the average MSE after regularization is 15 to 16. The average MSE does not change much because the model is not overfit, and the MSE on training set and the validation set is similar. The model is simply linear regression, which usually does not overfit. I looked at the weights of my model, and they are already small even without regularization, so the regularization does not affect much. I tried to increase the complexity, so I increase the feature of last 3 PM2.5 numbers using the square numbers, and it did make the MSE lower. After I append the feature, L2 regularization help a little, making the MSE to drop about 0.1.