Train-Test Set Split Question

In practical 2 we were introduced to a simple prediction algorithm based on linear regression using the Boston housing data set. We were given the following code:

```
y = df["median-value"]
X = df.loc[:, df.columns != "median-value"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In the third line we define the size of the test set to be 0.20 and the random state to be 42. With these parameters we achieved an RMSE of around 4.93. I was curious which results I could achieve with different values for the test set size and the random state.

Figure 1 shows how the values of RMSE change as we increase the test set size (random state = 42). The black dot represents the RMSE of 4.93 from the practical. This graph was achieved by running the linear regression 100 times (incrementing by 0.01).

The line didn't look like what I had expected so I ran the 100 linear regression over 100 iterations of the random state (from 1 to 100). The result is shown in figure 2. To make this graph more interpretable I graphed the mean ± one standard deviation per test set size. This is seen in figure 3.

What I found was counter to what I had predicted. I believed the lines would form a convex shape with the minimum somewhere around 0.10 - 0.30. Instead we see an 'L' shaped line with increasing variances towards the tails. What also struck me was that the optimal test size did not appear to be 0.20 or 0.30 but rather around 0.60. The lowest standard deviation is 0.2764 at a test set size of 0.60. The difference in means of RMSE between 0.60 and 0.20 is only 0.0641.

My question is whether this result seems to just be a quirk of the data set or whether a similar shape will appear for other data and if so, why do we choose a test size with a larger variance in its RMSEs than the optimum?

Figure 1

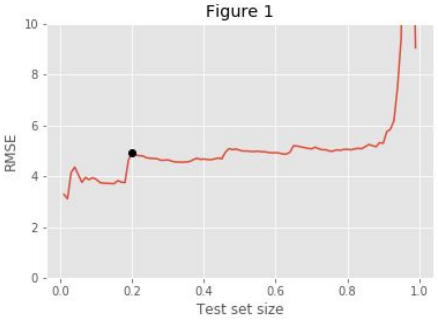

Figure 2

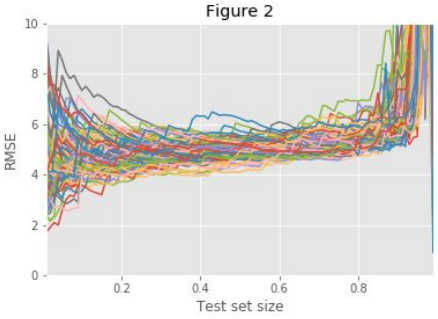

Figure 3