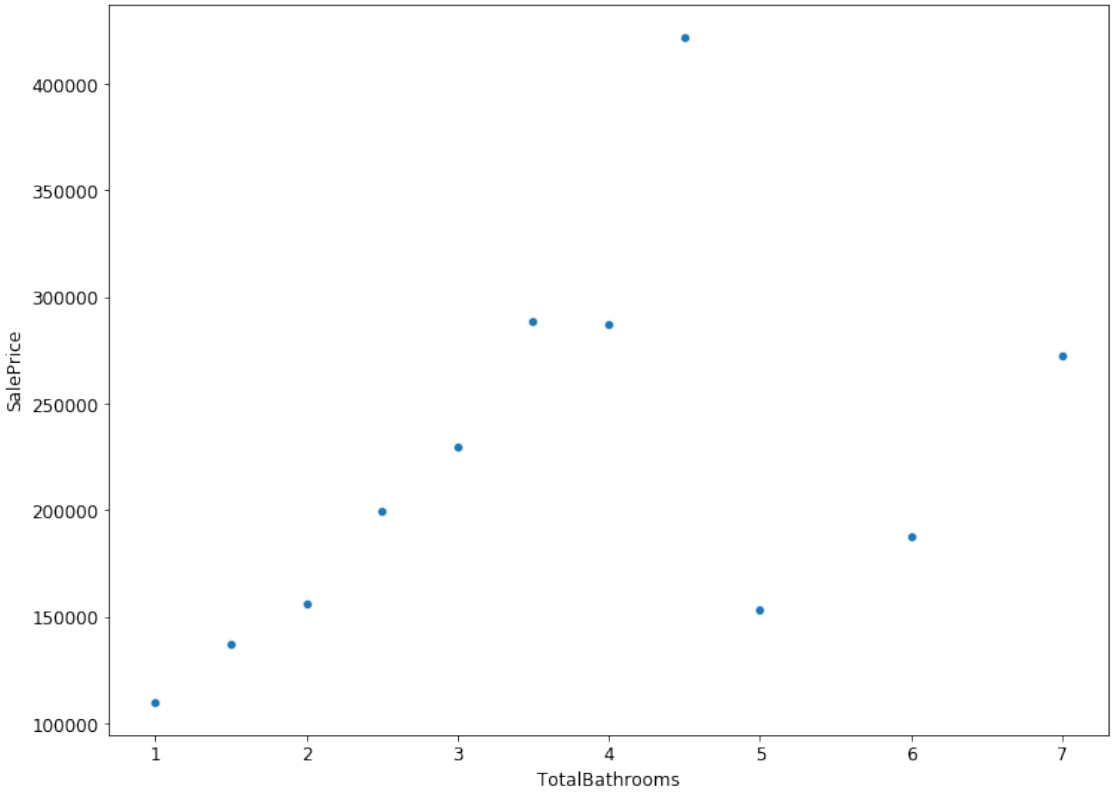
0.1 Question 2b

Create a visualization that clearly and succintly shows that `TotalBathrooms` is associated with `SalePrice`. Your visualization should avoid overplotting.

```
In [115]: price_and_bathrooms = training_data_with_bathrooms.groupby('TotalBathrooms').mean()[['SalePri
          sns.scatterplot(data=price_and_bathrooms, x='TotalBathrooms', y='SalePrice')
```

```
Out[115]: <matplotlib.axes._subplots.AxesSubplot at 0x7f1d743be490>
```



0.2 Question 5d

What changes could you make to your linear model to improve its accuracy and lower the validation error? Suggest at least two things you could try in the cell below, and carefully explain how each change could potentially improve your model's accuracy.

We could alter missing values and outlier values in the training data by making good guesses or using other strategies e.g. using the mean of a feature to fill out NaN values. This would improve accuracy because often, having missing or outlier values tends to give a false impression of one variable's relationship with another variable. This causes our predictions to be inaccurate. Hence, dealing with these missing values and outlier values will cause our model to be more accurate, and hence, cause a lower validation error.

We could also increase the size of our training data. If we do this, we are making less assumptions and doing more work because our training data looks a lot like our whole data. E.g. using 50% of our data as our training data would cause us to be making a sufficient amount of assumptions about the remaining 50% of the data, many of which may be incorrect. However, using 80% of our data as our training data means that a large subset of our whole data is already used in our training data, so we are less likely to make any false assumptions which would make our predictions inaccurate and hence reduce the credibility of our model. Thus, dealing with the size of the training data would cause us to make less inaccurate assumptions which would bring our model to better accuracy, and hence less validation error.

0.3 Question 6a

Based on the plot above, what can be said about the relationship between the houses' sale prices and their neighborhoods?

We can see that there is a negative relationship between the median sale price of a house in a neighborhood vs the number of houses in that neighborhood. This means that the more houses there are in a particular neighborhood, the lower we expect the median price of a house in that neighborhood to be. This pattern is illustrated by extreme values of the number of houses in a neighborhood e.g. the NAmes neighborhood has the maximum number of houses (299) and the median price of a house in NAmes lies well below the dotted line (which illustrates the median price of a house across all neighborhoods). On the other hand, the GrnHill neighborhood only has 2 houses but the median price of a house in the GrnHill neighborhood lies well above the dotted line. However, in order for there to be a strong negative relationship, we need this trend to be true for all neighborhoods, which is not true. There are many outliers e.g. StoneBr has the highest median price in all neighborhoods, but it doesn't have the lowest number of houses. Thus, the relationship between houses' sale prices in a neighborhood and the number of houses in the neighborhood is weakly negative.

0.4 Question 8a

Although the fireplace quality variable that we explored in Question 2 has six categories, only five of these categories' indicator variables are included in our model. Is this a mistake, or is it done intentionally? Why?

This is done intentionally. The category that was left out was "average" and the reason for this was to avoid redundancy, perhaps because it is too similar to the "fair" category. If there is redundancy in our design matrix, this will cause the columns of the design matrix to be linearly dependent. If they are linearly dependent, the matrix is not full-rank and is hence non-invertible. Also, we know that in order for there to be a least squares solution to determine optimal parameters, our design matrix must be invertible. Thus, we leave out the "average" feature because it would cause our design matrix to have linearly dependent columns which would not get us our desired optimal thetas.