Module 1 final project

Kal Lemma

Looking at the Data

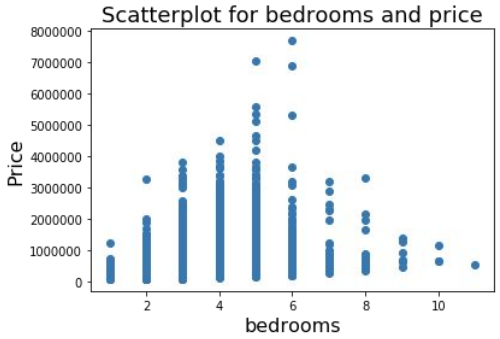
- Goal
  - Predict the sale price of the houses as accurately as possible for the data on Kings County.
- Took some assumptions about which predictors I could reasonably think would have more weight on price
  - Initial choices included,,'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'waterfront', 'grade', 'sqft_above',,, 'yr_renovated', 'zipcode', 'sqft_living15',
  - Why
- I wanted to actually first check what my multi regression would look like in the beginning, without doing any cleaning of sorts (curious on what originally had the largest influence)
  - r^2 = 0.745
    - High p-value for nearly all values

More questions, more answers

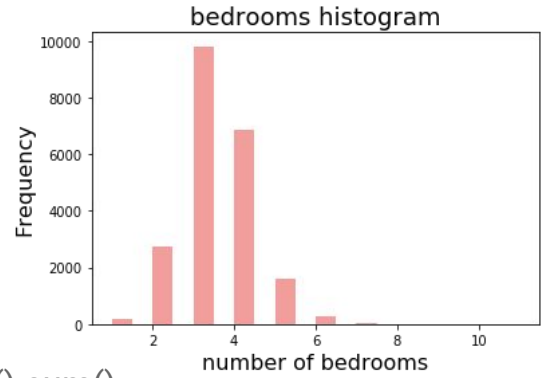
- After looking through my data and asking different questions on what would not help my regression, I clearly noticed some outliers.
  - While using df.describe(), I could notice some interesting points like 75% of homes would have four bedroom homes.
    - It's possible homes with 4 bedrooms were more desirable to build for economic purposes: having more people in housing, increase of costs for potential profits, and better area efficiency.
  - Also there were some potential max values that seem to be disproportionate to the rest of the values in their column.
    - I thought I would like to take out the outliers for bedrooms, bathrooms, sqft_living, sqft_above, , sqft_lot15, sqft_living15
      - But I needed to be sure

Visualizing for better understanding

- Sorting through the values of each of the columns I thought contained outliers proved to be beneficial, by finding out only bedrooms, bathrooms, sqft_living, _above, and _lot15 needed some dropping.
    - Sqft_living15 looked fine to me after sorting it
- Taking scatter plot between bedrooms and price (checking out the relationship)
    - We can see that the 33 bedroom house wouldn't be essential in our data frame since it's so far away from any of the other points. You can also observe clearly that the most expensive homes had either 5 or 6 bedrooms.



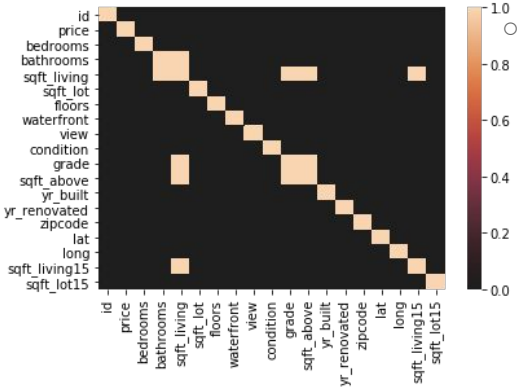
Missing NaN values, more understanding.

- Histogram of bedrooms
  - 

- df.isna().sum()
  - Clearly shows the sum of how many NaN values there are in our dataframe
    - Waterfront, View, and yr_renovated
- Created dummy variables for zipcodes to see if there is a stronger correlation specific to each zipcode

Multicollinearity?

- df.corr()
  - Checking out if any of my original assumptions about the data set could be supported by the correlation. Realized that lat is far more correlated with price than I assumed.
  - abs(df.corr()) > 0.75
    - Shows that none of the columns, except price itself, is highly correlated with price.
    - Also see that some collinearity occurring with (sft_living with bathrooms, grade, sqft_living15, and sft_above) and (sqft_above with grade)
      - Visualized through HeatMap
        - Orange blocks represent .75 correlation or greater, which our heatmap clearly shows which predictors will cause multicollinearity and should be removed.



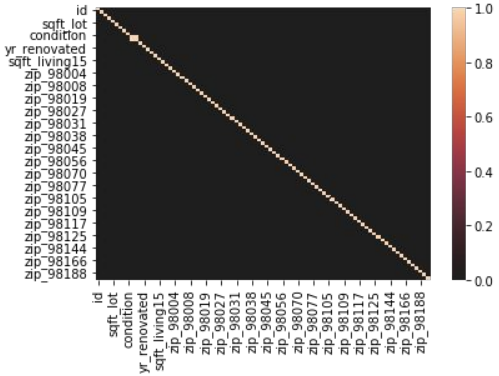
Cleaning all the dirty values I found

- Taking care of the Outliers
  - Removed outliers from bedrooms, bathrooms, sqft_living, _above, and _lot15
    - Took out the 33 bedroom home, removed the 4 homes that had more than 7.5 bathrooms, dropped some rare high & lows for sqft_living and sqft_lot15, and removed the two homes that were more than 8000 ft
- Removing our NaNs
  - Waterfront, yr_renovated, view
    - df= df[pd.notna(df['waterfront'])]
      - Using this format, where I'm setting my new data frame with the all the values that don't equal NaN
- Taking out Collinearity
  - Questioned whether to remove sft_living, which was correlated with 4 other variables or vice versa.

Fitting

I began taking out collinear variables and then regressing without them, while the outliers that were taken out.

Taking out 'sqft_living', and including our new dummy zip code columns, has removed nearly all collinearity in our data frame.



OLS after changes

1. Took the same OLS from the beginning with all columns except 'sqft_living', checking the differences from our cleaning
    a. r^2 = 0.690
2. Took the previous OLS, but now removing ‘sqft_lot’, ‘condition’, and “floors’
    a. r^2 = 0.686
3. Finally, I included the dummy variables for zip codes, dropped my collinear column of 'sqft_living', with all outliers and NaN values taken out, and dropping columns that may not be beneficial
    a. r^2 =0.692
        i. Two highest pvalues,
            1. 0.646 for sqft_lot15
            2. 0.053 for zip_98188