1 Part 1: Contextualizing the Data

Let's try to understand the background of our dataset before diving into a full-scale analysis.

1.1 Question 1a

Based on the columns present in this data set and the values that they take, what do you think each row represents? That is, what is the granularity of this data set?

Looking at the column names that are present in this data set, some of the values are going to be quantitative (i.e., what is square footage of the house) and others are going to be qualitative. For instance, `repair condition` is something that is subjective and cannot be easily described using a quantitative metrics. What some consider a good repair condition may be perceived as terrible. Something like `latitude` and `longitude` are going to be quantitative and not subjective, but rather it is subjective.

The data in this set is predominantly quantitative, looking at the `codebook.txt` file only a few entries are non numbers. Some values are in square footage, while some others are boolean values that indicate if something is present or not.

1.2 Question 1b

Why do you think this data was collected? For what purposes? By whom?

This question calls for your speculation and is looking for thoughtfulness, not correctness.

I have a strong suspicion that this data was collected firstly as a mass survey, to possibly make inferences about how much of the populations have `x` feature present in them. Consequently, after this data was collected it could be used to set housing prices in out skirts of this community. More than likely, it was collected to set house prices of other future houses and what those around others could potentially sell their homes for if they wished to.

1.3 Question 1c

Craft at least two questions about housing in Cook County that can be answered with this data set and provide the type of analytical tool you would use to answer it (e.g. "I would create a _____ plot of _____ and ***" or "I would calculate the*** [summary statistic] for _____ and _____"). Be sure to reference the columns that you would use and any additional data sets you would need to answer that question.

Question 1 What is the correlation between the square footage of the land and the sale price of said property? To answer this question, I would create a correlation plot referencing `Land Square Feet` and `Sale Price` where `Land Square Feet` is on the x axis and the `Sale Price` is on the y axis.

Question 2 What is the most common type of basement that is present in this home? To answer this question, I would create a frequency histogram of the types of basements referencing `Basement` as this is a quantifiable way to describe what type of basement is present. As usual, the variable that we are examining, in this case `Basement` would be on the x axis and the frequency of this type of basement would be shown on the y axis.

1.4 Question 1d

Suppose now, in addition to the information already contained in the dataset, you also have access to several new columns containing demographic data about the owner, including race/ethnicity, gender, age, annual income, and occupation. Provide one new question about housing in Cook County that can be answered using at least one column of demographic data and at least one column of existing data and provide the type of analytical tool you would use to answer it.

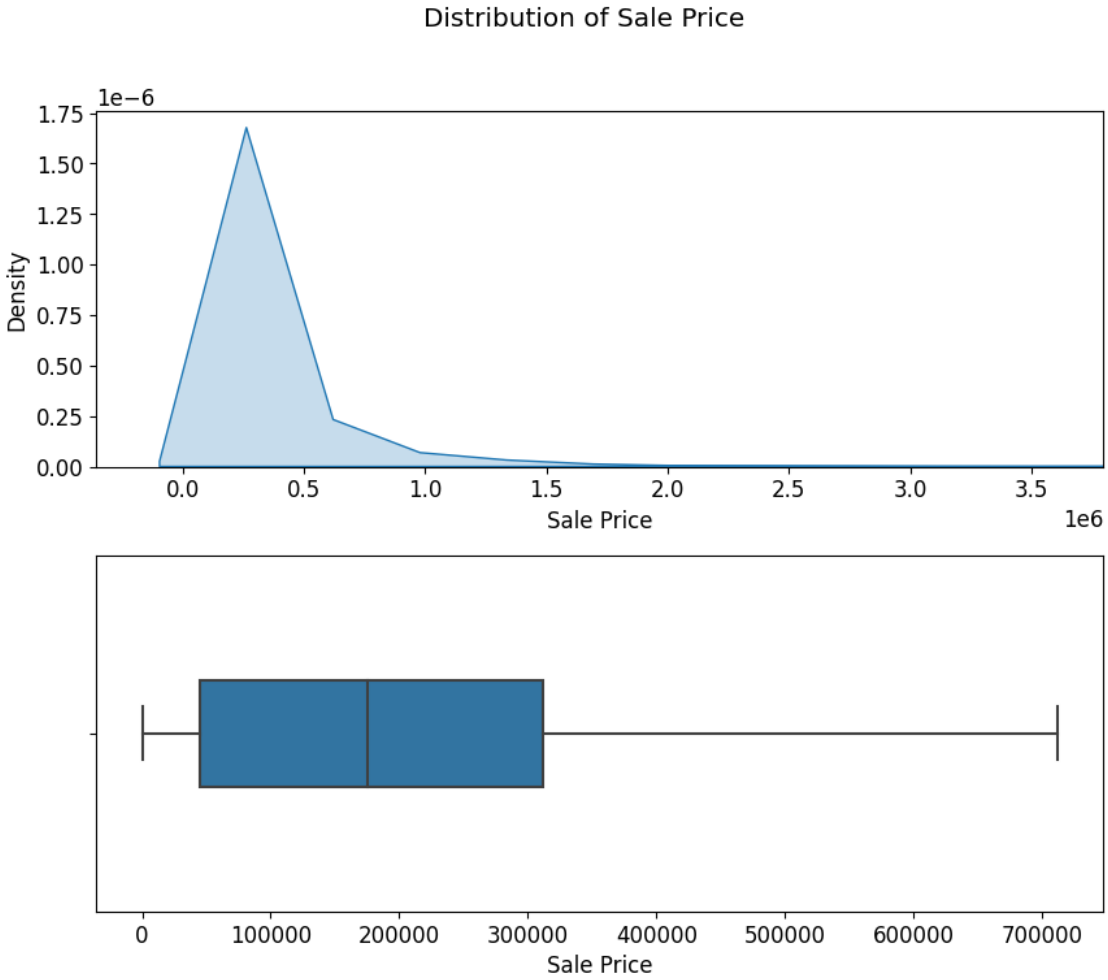
Question What is the correlation between the estimate for the building that the owner possesses and the annual income of the owner? To answer this, I would create a correlation plot of `Estimate (Building)` on the x axis and the `annual income` of the owner on the y axis.

1.5 Question 2a

Using the plots above and the descriptive statistics from `training_data['Sale Price'].describe()` in the cells above, identify one issue with the visualization above and briefly describe one way to overcome it.

It is very hard to observe the plots in the graph. The outliers that are present in the data set are skewing both the density distribution and the box plot. To make this more readable, we could change the limits on the x axis for the distribution plot so that it would be easier to see. This won't exactly give us the full scope of the data (its cutting out the most expensive home) but it will still give us a good view of what is going on. See the modifications that I made below.

```
In [11]: # optional cell for scratch work
         def plot_distribution_mod(data, label):
             fig, axs = plt.subplots(nrows=2, figsize=(10, 8))

             sns.kdeplot(
                 data[label],
                 ax=axs[0],
                 fill=True
             )

             sns.boxplot(
                 x=data[label],
                 ax=axs[1],
                 width=0.3,
                 showfliers=False
             )

             spacer = np.max(data[label]) * 0.05
             xmin = 0 - 0.1 * spacer
             xmax = np.mean(data[label]) + spacer
             axs[0].set_xlim((xmin, xmax))

             axs[0].xaxis.label.set_visible(True)
             axs[1].yaxis.label.set_visible(True)

             plt.subplots_adjust(hspace=0.25)
             fig.suptitle("Distribution of " + label)

             plt.show()

         plot_distribution_mod(training_data, label='Sale Price')
```



1.6 Question 2b

To zoom in on the visualization of most households, we will focus only on a subset of `Sale Price` for this assignment. In addition, it may be a good idea to apply log transformation to `Sale Price`. In the cell below, reassign `training_data` to a new dataframe that is the same as the original one **except with the following changes**:

- `training_data` should contain only households whose price is at least $500.
- `training_data` should contain a new `Log Sale Price` column that contains the log-transformed sale prices.

You should NOT remove the original column `Sale Price` as it will be helpful for later questions.

If you accidentally remove it, just restart your kernel and run the cells again.

Note: This also implies from now on, our target variable in the model will be the log-transformed sale prices from the column `Log Sale Price`.

To ensure that any error from this part does not propagate to later questions, there will be no hidden tests for this question.

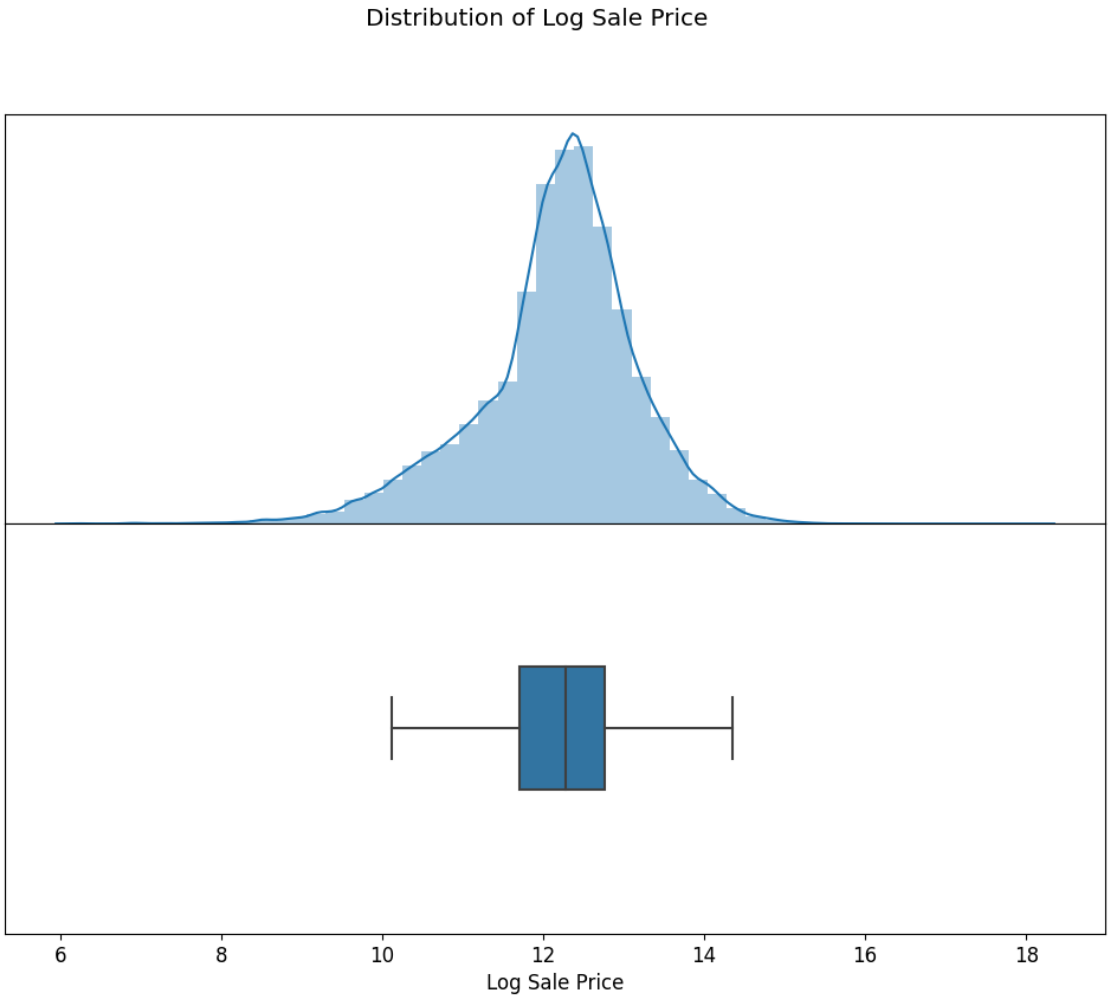
```
In [12]: log_sale_price = np.log(training_data['Sale Price'])
         training_data = training_data[training_data['Sale Price'] >= 500]
         sale_price_idx = training_data.columns.get_loc('Sale Price') + 1
         training_data.insert(loc=sale_price_idx, column='Log Sale Price', value=log_sale_price)
```

```
In [13]: grader.check("q2b")
```

```
Out[13]: q2b results: All test cases passed!
```

Let's create a new distribution plot on the log-transformed sale price.

```
In [14]: plot_distribution(training_data, label='Log Sale Price');
```



1.7 Question 3a

Is the following statement correct? Assign your answer to `q3statement`.

```
"At least 25% of the properties in the training set sold for more than $200,000.00."
```

Note: The provided test for this question does not confirm that you have answered correctly; only that you have assigned each variable to `True` or `False`.

```
In [15]: # These should be True or False
         q3statement = True
```

```
In [16]: grader.check("q3a")
```

```
Out[16]: q3a results: All test cases passed!
```

1.8 Question 3b

Next, we want to explore if there is any correlation between `Log Sale Price` and the total area occupied by the property. The `codebook.txt` file tells us the column `Building Square Feet` should do the trick – it measures "(from the exterior) the total area, in square feet, occupied by the building".

Let's also apply a log transformation to the `Building Square Feet` column.

In the following cell, create a new column `Log Building Square Feet` in our `training_data` that contains the log-transformed area occupied by each property.

You should NOT remove the original `Building Square Feet` column this time, as it will be used for later questions. If you accidentally remove it, just restart your kernel and run the cells again.

To ensure that any errors from this part do not propagate to later questions, there will be no hidden tests for this question.

```
In [17]: log_bldg_sqft = np.log(training_data['Building Square Feet'])
         log_bldg_sqft_idx = training_data.columns.get_loc('Building Square Feet') + 1
         training_data.insert(loc=log_bldg_sqft_idx, column='Log Building Square Feet', value=log_bldg_s
```

```
In [18]: grader.check("q3b")
```

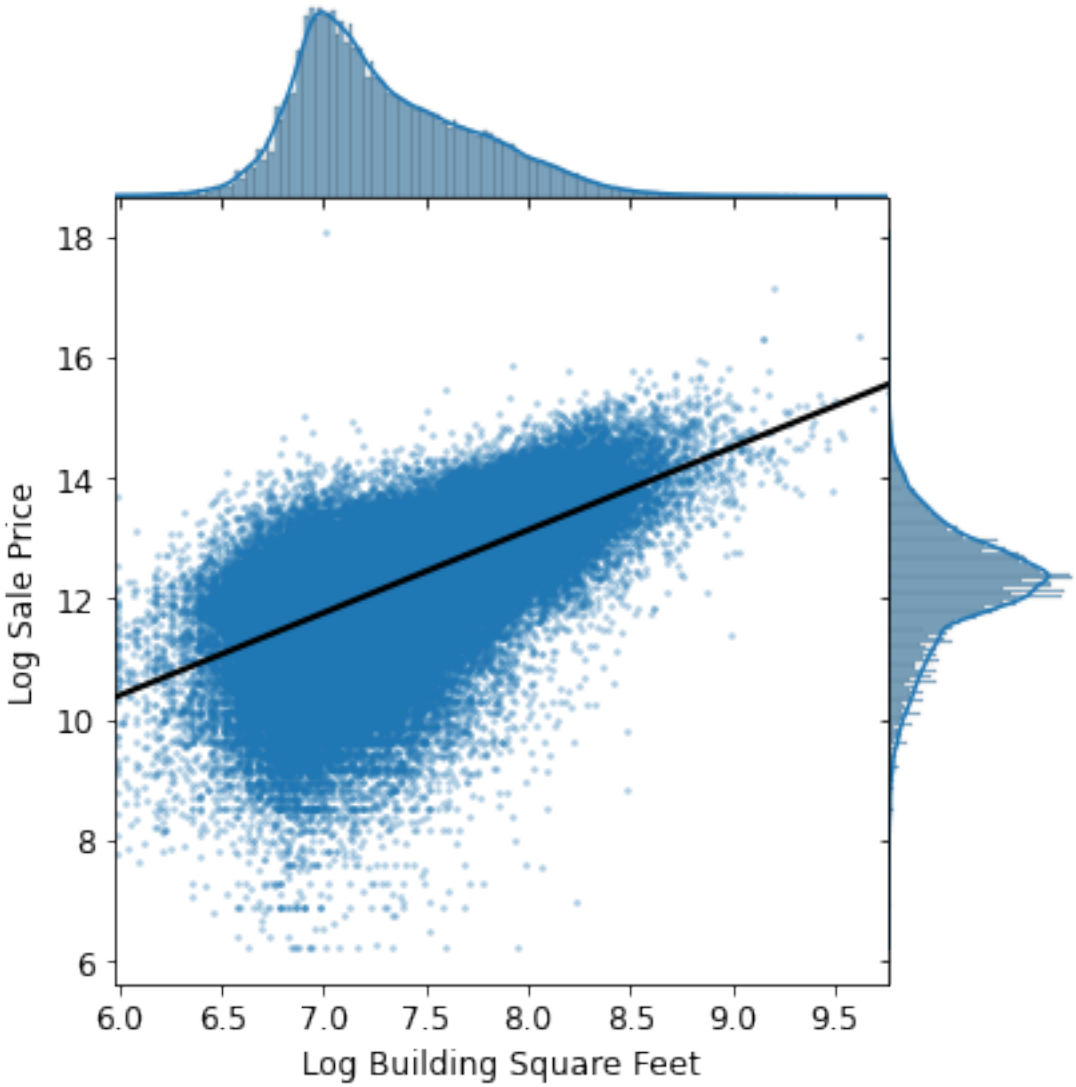
```
Out[18]: q3b results: All test cases passed!
```

1.9 Question 3c

In the visualization below, we created a `jointplot` with `Log Building Square Feet` on the x-axis, and `Log Sale Price` on the y-axis. In addition, we fit a simple linear regression line through the bivariate scatter plot in the middle.

Based on the following plot, would `Log Building Square Feet` make a good candidate as one of the features for our model? Why or why not?

Hint: To help answer this question, ask yourself: what kind of relationship does a "good" feature share with the target variable we aim to predict?



Yes, `Log Building Square Feet` does make a good candidate as one of the features of our model. This is because there is a clear linear relationship between the variables. Along with the linearity, the scatter plot

in the aforementioned plot has most points centered around the regression line, indicating that this is a good fit. And lastly, we can see that the distributions of this data is normal and not with an overbearing skew, thus showing that it would be a good feature of our model.

1.10 Question 5c

Create a visualization that clearly and succinctly shows if there exists an association between `Bedrooms` and `Log Sale Price`. A good visualization should satisfy the following requirements: - It should avoid overplotting. - It should have clearly labeled axes and a succinct title. - It should convey the strength of the correlation between `Sale Price` and the number of rooms: in other words, you should be able to look at the plot and describe the general relationship between `Log Sale Price` and `Bedrooms`

Hint: A direct scatter plot of the `Sale Price` against the number of rooms for all of the households in our training data would result in overplotting (since there are only a small discrete number of bedrooms) - so **don't use a scatter plot**.

```
In [28]: sns.boxplot(x = 'Bedrooms', y = 'Log Sale Price', data = training_data, whis=6)
         plt.xlabel("Number Of Bedrooms")
         plt.ylabel("Log Sale Price")
         plt.title("Log Sale Price vs. Number Of Bedrooms")
         plt.show()
```