0.1 Question 1: Human Context and Ethics

In this part of the project, we will explore the human context of our housing dataset.

You should read the Project_CaseStudy.pdf on Canvas explaining the context and history surrounding this dataset before attempting this section.

0.1.1 Question 1a

"How much is a house worth?" Who might be interested in an answer to this question? **Please list at least three different parties (people or organizations) and state whether each one has an interest in seeing the housing price to be high or low.**

1. Homeowners - Generally speaking, the higher a home's value, the higher the property taxes are for that home. Because of this, homeowners have a mixed interest in housing prices. Some may want their properties to be appraised as high as possible to increase their equity and increase resale value, at the expense of higher property taxes. Other homeowners may want a lower appraisal to avoid property tax increases. This could be split into two parties. Typically a higher income or net worth homeowner is interested in a higher appraisal as the property tax increase is less of a financial burden, they might be more aware and capable of taking advantage of property tax tax write offs, and tend to have the priviledge of being able to focus on building long term wealth in the form of equity or investments. For lower income or net worth homeowners, a property tax increase can be a significant financial burden, and often leads to these homeowners being priced out of their homes in areas that are rapidly gentrifying. In the case of homeowners who may benefit from a higher appraisal, it is worth mentioning that the property assessment used in property tax adjustments is separate from the appraisal used for sale prices. While these are often related, they are distinct, so there could be an under assessed property with disproportionally low property taxes that still sells for a high price, which is the best of both worlds for these homeowners.
2. Cook County Assessor's Office - Ideally, this party has no interest in seeing high or low housing prices and is focused on an accurate assessment for every property. Realistically, this might not be the case, as I imagine we might see further into this project. If there was to be an influence on assessment values, I'd expect it to increase assessed property values, leading to higher property taxes collected by the local government.
3. Real Estate Professionals - Real estate agents and developers would benefit from higher property values, which would increase their commissions and investment appeals.

0.1.2 Question 1b

Which of the following scenarios strike you as unfair and why? You can choose more than one. There is no single right answer, but you must explain your reasoning. Would you consider some of these scenarios more (or less) fair than others? Why?

A. A homeowner whose home is assessed at a higher price than it would sell for.
B. A homeowner whose home is assessed at a lower price than it would sell for.
C. An assessment process that systematically overvalues inexpensive properties and undervalues expensive properties.
D. An assessment process that systematically undervalues inexpensive properties and overvalues expensive properties.

In my mind, all of these are unfair, though I'd say that C is the most unfair. In scenarios A and B, the home is over or under assessed, so the homeowner will be paying more or less property tax than they should be. In scenario A, the home is over assessed, so the homeowner is paying more than their fair share of property taxes, which is obviously unfair. Scenario B, where the homeowner is paying less than their fair share of property taxes, might not be viewed as unfair by that homeowner, though I still consider it unfair. Property taxes often fund a variety of public services and infrastructure, like schools, police or fire departments, roads, bridges, streetlights, local parks, public health initiatives, etc. The homeowner who pays less in property taxes can and likely does utilize these public services, though without supporting them proportionally. This creates an imbalance where the other taxpayers cover the shortfall. So while the individual homeowner benefits, this undermines the equity of the system as a whole. Scenarios A and B don't sound indicative of a larger problem though, they could just be singular mistakes or outliers. At the very least, scenarios A and B don't necessarily include a property assessment process that is systematically unfair.

Both scenarios C and D include a systematic valuation problem that impacts many homeowners, making them far more damaging and unfair in my eyes. In scenario C, this flawed assessment process overvalues inexpensive properties and undervalues expensive ones. To me, this seems particularly unfair because it places a disproportionate tax burden on homeowners with less expensive properties, who often have fewer resources to absorb higher tax rates. At the same time, wealthier homeowners benefit from lower taxes. This systemic unfairness would worsen existing inequality and perpetuate economic disparities, so I feel that it is the most unfair scenario.

Scenario D, where inexpensive properties are undervalued and expensive properties are overvalued shares the systemic aspect of scenario C, but doesn't seem as immediately harmful as scenario C. Wealthier homeowners would bear a larger share of the tax burden, though they are likely more equipped to do so. This would likely benefit lower income homeowners to some extent, as they would pay less in taxes, but I expect it would cause some issues as well. For example, low and high value homes aren't typically homogenous in an area, there are usually clusters or neighborhoods that are predominantly one or the other. As we mentioned above, property taxes support a variety of public services, and these are often funded by the taxes from the surrounding area. So in this situation where inexpensive properties, which are likely grouped together and funding the same local public services, are undervalued and contributing less to these public services, I think its reasonable to assume these public services would suffer. While the outcome or intent of this could be viewed as a Robin Hood-esque redistribution of wealth, shifting the tax burden to wealthier homeowners and giving relief to the less fortunate, a targeted and deliberate redistribution from intentional policy, not a systemic flaw, would be a better approach. As much as the idea of redistributing wealth is redeeming in my eyes, and does seem "fair" to some degree, I can't call this scenario fair, ideal, or sustainable.

0.1.3 Question 1d

What were the central problems with the earlier property tax system in Cook County as reported by the Chicago Tribune ? And what were the primary causes of these problems? (Note: in addition to reading the paragraph above you will need to **read the Project_CaseStudy.pdf explaining the context and history of this dataset before answering this question).**

There were several central problems in the tax system in Cook County identified by the Chicago Tribune. For one, they found regressive taxation, where lower value homes were consistently over assessed, leading to a disproportionately high property taxes for lower income homeowners. At the same time, higher value homes were under assessed, so wealthier homeowners paid less than their fair share of taxes. Alongside this, there were racial disparities where the likelihood of a property being over or under assessed was strongly correlated with the racial makeup of the neighborhoods. These correlations showed that non-white, working class homeowners were likely to be paying more in property taxes relative to their home's value than predominantly white, more affluent homeowners.

As in most counties, there was an assessment appeals process in Cook County, though it was found to be unfair and inaccessible to those most negatively affected by the regressive taxation. Similarly to other tax and financial areas, wealthier individuals could more easily navigate the process to reduce their assessments, often through the use of legal representation, which was a luxury often inaccessible to more low income homeowners. In fact, the model and its processes were described as rife with institutional bias and corruption, with stakeholders including tax lawyers and politicians benefiting from the inequities. Additionally, there was little accountability or oversight in the system to correct these biases, so these inequities were allowed to persist for years.

For the primary causes, Cook County had a legacy of inequitable policies. Historical practices, such as redlining and racially motivated housing policies, created lasting disparities in property values and access to public resources. Data quality was also an issue, Cook County was using outdated and inaccurate info and failed to account for neighborhood level variations and socioeconomic differences. This was largely due to poor quality data collection leading to significant gaps in the data, particularly in lower income neighborhoods. The assessment was also not designed to account for socioeconomic and racial disparities, leading to biases. This bad data quality and analysis worsened errors in assessments in these neighborhoods while the lack of transparency and opaque nature of the property assessment process made it difficult to identify and address these issues. Finally, there was some evidence of institutional bias. Policies and practices often reflect the interests of more affluent homeowners, people, and professionals, neglecting the needs of more marginalized communities.

0.2 Question 2a: More EDA

In good news you have already done a lot of EDA with this dataset in Project 1.

Before fitting any model, we should check for any missing data and/or unusual outliers.

Since we're trying to predict `Sale Price`, we'll start with that field.

Examine the `Sale Price` column in the `training_val_data` DataFrame and answer the following questions:

- 2ai). Does the `Sale Price` data have any missing, N/A, negative or 0 values for the data? If so, propose a way to handle this.
- 2aii). Does the `Sale Price` data have any unusually large outlier values? If so, propose a cutoff to use for throwing out large outliers, and justify your reasoning).
- 2aiii). Does the `Sale Price` data have any unusually small outlier values? If so, propose a cutoff to use for throwing out small outliers, and justify your reasoning.

Below are three cells. The first is a Markdown cell for you to write up your responses to all 3 parts above. The second two are code cells that are available for you to write code to explore the outliers and/or visualize the Sale Price data.

0.2.1 Question 2abc answer cell:** *Put your answers in this cell...*

2ai: There are 0 missing values and no 0 or negative values, so no action is needed for this part.

2aii: For this, we used an upper limit for outliers of 712,200 and there were 12,229 large outliers. To find this upper limit, we calculated the IQR and defined the upper limit as Q3 + 1.5 * IQR = 712,200. Because of the significant number of large outliers, this should be addressed. To do so, a IQR based cutoff of Q3 + 1.5 * IQR = 712,200 could be used to remove or treat large outliers. A transformation like log scaling would also be a good approach. Values above this threshold deviate significantly and could distort the model's predictions if this isn't addressed.

2aiii: Yes, the Sale Price data has 35,549 unusually small outlier values. These small values are likely not reflective of the true market value of the homes. To identify these outliers, we used $10 as the cutoff, which could also be used to exclude these small outliers. Any home price below this cutoff is likely an irregular transaction from a data entry error or unique sale conditions. Including these extremely small values would distort the analysis and model predictions, they are outliers that do not represent typical sale prices.

```
In [10]: # check for missing, NaN, or 0 values in sale price
         missing_values = training_val_data['Sale Price'].isna().sum()
         zero_values = (training_val_data['Sale Price'] <= 0).sum()
```

```python
print(f"Missing values: {missing_values}")
print(f"Zero or negative values: {zero_values}")

########################################################################

# calc IQR to find large outliers
Q1 = training_val_data['Sale Price'].quantile(0.25)
Q3 = training_val_data['Sale Price'].quantile(0.75)
IQR = Q3 - Q1

# define upper limit for outliers
upper_limit = Q3 + 1.5 * IQR

print(f"Upper limit for outliers: {upper_limit}")

# filter large outliers
large_outliers = training_val_data[training_val_data['Sale Price'] > upper_limit]
print(f"Number of large outliers: {large_outliers.shape[0]}")

##########################################################################

# find small outliers
small_outliers = training_val_data['Sale Price'][training_val_data['Sale Price'] < 10]
print(f"Number of small outliers: {small_outliers.shape[0]}")

# your code exploring Sale Price above this line
```

```
Missing values: 0
Zero or negative values: 0
Upper limit for outliers: 712200.0
Number of large outliers: 12229
Number of small outliers: 35549
```

```
In [ ]: ...
        # optional extra cell for exploring code
```

0.3 Question 5: Improving the Model

0.3.1 Question 5a: Choose an additional feature

It's your turn to choose another feature to add to the model. Choose one new **quantitative** (not qualitative) feature and create Model 3 incorporating this feature (along with the features we've already chosen in Model 2). Try to choose a feature that will have a large impact on reducing the RMSE and/or will improve your residual plots. This can be a raw feature available in the dataset, or a transformation of one of the features in the dataset, or a new feature that you create from the dataset (see Project 1 for ideas). In the cell below, explain what additional feature you have chosen and why. Justify your reasoning. There are optional code cells provided below for you to use when exploring the dataset to determine which feature to add.

Note: There is not one single right answer as to which feature to add, however you should make sure the feature decreases the Cross Validation RMSE compared to Model 2 (i.e. we want to improve the model, not make it worse!)
This problem will be graded based on your reasoning and explanation of the feature you choose, and then on your implementation of incorporating the feature.

NOTE Please don't add additional coding cells below or the Autograder will have issues. You do not need to use all the coding cells provided.

0.3.2 Question 5a Answer Cell:

In this cell, explain what feature you chose to add and why. Then give the equation for your new model (use Model 2 from above and then add an additional term).

I am choosing to add Land Square feet to the model. I started my search by looking through other available quantitative features, but Land Square Feet and Age stood out as potentially logical choices. For Land Square Feet, it directly relates to property size, which is intuitively associated with Sale Price. I felt Age had a similar logical or intuitive appeal. I also did a series of analyses, including examining the features distribution, correlations, and residuals pattern, to further investigate if the feature would be a good addition to the model.

The histogram below of Land Square Feet shows that the distribution is highly skewed, with most values concentrated near zero but with some significantly larger outliers. I was initially concerned with this skewness, but it could also suggest variability, particularly for larger properties, which the model could potentially capture to explain Sale Price better.

Then I looked at the correlations. The correlation between Land Square Feet and Log Sale Price is 0.176, indicating a weak positive relationship. This relationship suggests that as Land Square Feet increases, the Sale price tends to increase slightly. While this was not strong by any means, the correlation still shows that this feature could have some predictive information. The correlation between Land Square Feet and Residuals from model 2 is approximately -0.016, which is very close to zero. This weak correlation indicates that Land Square Feet is not strongly related to the existing residuals, meaning its predictive information is not already captured by the current model.

Finally, the residuals analysis shows that while there is still some scatter, particularly among smaller land sizes, there appears to be potential for improvement at the higher end of the scale where residuals are more variable. Including Land Square Feet may help capture this relationship and reduce residual errors.

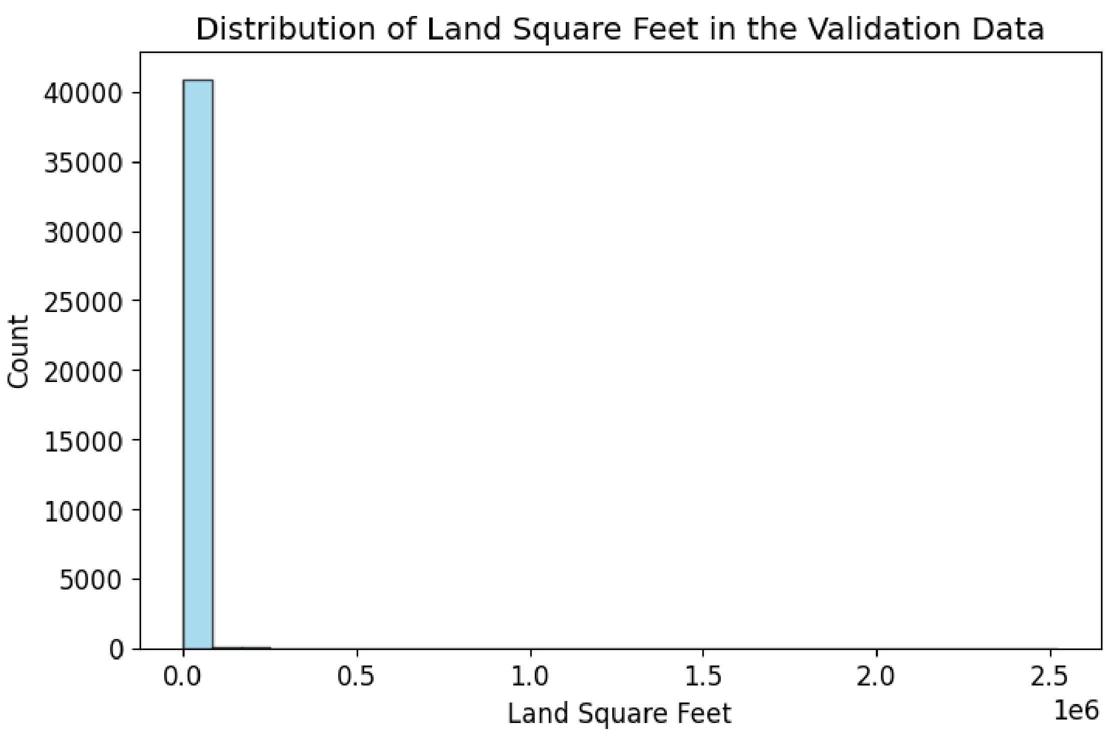
So I am adding Land Square Feet to the model because it offers some predictive power for explaining Sale Price, does not appear to be redundant with the features already included, and could contribute to improving the model's performance because of this. In all honesty though, I first added Age to the model as it showed a much higher correlation with log(Sale Price). Surprisingly, when I calculated the cross validation RMSE after adding Age, it was actually slightly higher than the cross validation RMSE in model 2. This caused me to reconsider Land Square Feet, which ended up providing a lower cross validation RMSE than Age and model 2.

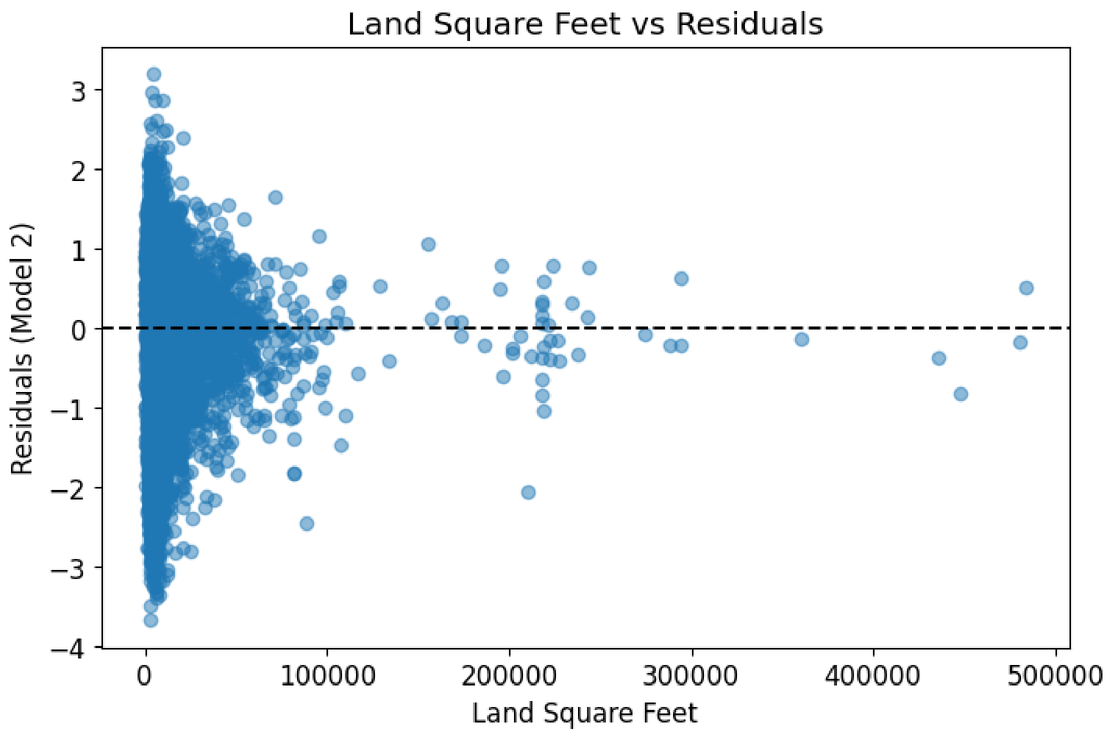
Model 3 Equation:

Log Sale Price = θ_1(Log Building Square Feet)+θ_2(Shingle/Asphalt)+θ_3(Tar&Gravel)+θ_4(Tile)+θ_5(Shake)+θ_6(Other)+θ_7(S

In [150]:
```
# Step 1: Define the feature to analyze
feature_name = "Land Square Feet"  # <-- Change this to explore other features

print(valid.columns)
print(valid_comp.columns)

# Step 2: Visualize the distribution of the chosen feature in the dataset
plt.figure(figsize=(8, 5))
plt.hist(valid[feature_name].dropna(), bins=30, alpha=0.7, color='skyblue', edgecolor='black')
plt.xlabel(feature_name)
plt.ylabel("Count")
plt.title(f"Distribution of {feature_name} in the Validation Data")
plt.show()

# Step 3: Analyze the feature vs Residuals (using valid residuals from Model 2)
valid_with_feature = valid.copy()
valid_with_feature['Residuals'] = Y_valid_m2 - Y_predict_valid_m2  # Add Model 2 residuals

plt.figure(figsize=(8, 5))
plt.scatter(valid_with_feature[feature_name], valid_with_feature['Residuals'], alpha=0.5)
plt.axhline(0, color='black', linestyle='--')
plt.xlabel(feature_name)
plt.ylabel("Residuals (Model 2)")
plt.title(f"{feature_name} vs Residuals")
plt.show()

# Step 4: Correlation analysis
feature_correlation_log_price = valid[feature_name].corr(valid_comp['Log Sale Price'])
feature_correlation_residuals = valid_with_feature[feature_name].corr(valid_with_feature['Resi

print(f"Correlation between {feature_name} and Log Sale Price:", feature_correlation_log_pric
print(f"Correlation between {feature_name} and Residuals:", feature_correlation_residuals)


# Show work in this cell exploring data to determine which feature to add
```

```
Index(['PIN', 'Property Class', 'Neighborhood Code', 'Land Square Feet',
       'Town Code', 'Apartments', 'Wall Material', 'Roof Material', 'Basement',
       'Basement Finish', 'Central Heating', 'Other Heating', 'Central Air',
       'Fireplaces', 'Attic Type', 'Attic Finish', 'Design Plan',
       'Cathedral Ceiling', 'Construction Quality', 'Site Desirability',
       'Garage 1 Size', 'Garage 1 Material', 'Garage 1 Attachment',
       'Garage 1 Area', 'Garage 2 Size', 'Garage 2 Material',
       'Garage 2 Attachment', 'Garage 2 Area', 'Porch', 'Other Improvements',
       'Building Square Feet', 'Repair Condition', 'Multi Code',
       'Number of Commercial Units', 'Estimate (Land)', 'Estimate (Building)',
       'Deed No.', 'Sale Price', 'Longitude', 'Latitude', 'Census Tract',
       'Multi Property Indicator', 'Modeling Group', 'Age', 'Use',
       'O'Hare Noise', 'Floodplain', 'Road Proximity', 'Sale Year',
       'Sale Quarter', 'Sale Half-Year', 'Sale Quarter of Year',
       'Sale Month of Year', 'Sale Half of Year', 'Most Recent Sale',
       'Age Decade', 'Pure Market Filter', 'Garage Indicator',
       'Neigborhood Code (mapping)', 'Town and Neighborhood', 'Description',
       'Lot Size'],
      dtype='object')
Index(['Log Building Square Feet', 'Roof Material', 'Bedrooms',
       'Log Sale Price', 'M1residuals_log'],
      dtype='object')
```






```
Correlation between Land Square Feet and Log Sale Price: 0.1760909955365289
Correlation between Land Square Feet and Residuals: -0.015880769688252517
```

```
In [ ]: ...

        # Optional code cell for additional work exploring data/ explaining which feature you chose.
```

```
In [ ]: ...

        # Optional code cell for additional work exploring data/ explaining which feature you chose.
```

```
In [ ]: ...

        # Optional code cell for additional work exploring data/ explaining which feature you chose.
```

0.3.3 Question 5b: Create Model 3

In the cells below fill in the code to create and analyze Model 3 (follow the Modeling steps outlined in Questions 3 and 4).

PLEASE DO NOT ADD ANY ADDITIONAL CELLS IN THIS PROBLEM OR IT MIGHT MAKE THE AUTOGRADER FAIL

In [157]:
```python
# Modeling Step 1:  Process the Data

# Hint: You can either use your implementation of the One Hot Encoding Function from Project 

from feature_func import *

...
# Optional:  Define any helper functions you need for one-hot encoding above this line


def process_data_m3(data):

    # You should start by only keeping values with Pure Market Filter = 1

    data = data[data["Pure Market Filter"] == 1]

    # add log transformed  columns
    data = data.copy()
    data["Log Sale Price"] = np.log(data["Sale Price"])
    data["Log Building Square Feet"] = np.log(data["Building Square Feet"])

    # add the Age feature
    data["Land Square Feet"] = data["Land Square Feet"]

    # one hot encode roof material
    data = ohe_roof_material(data)

    # Select columns for model 3 while ensuring order
    columns_for_model = ["Log Building Square Feet", "Land Square Feet", "Log Sale Price"] + 
                        [col for col in data.columns if "Roof Material_" in col]

    data = data[columns_for_model]

    return data


# Process the data for Model 3
processed_train_m3 = process_data_m3(train)

processed_val_m3 = process_data_m3(valid)

# Create X (Dataframe) and Y (series) to use to train the model
X_train_m3 = processed_train_m3.drop(columns="Log Sale Price")
```

```python
Y_train_m3 = processed_train_m3["Log Sale Price"]

X_valid_m3 = processed_val_m3.drop(columns="Log Sale Price")
Y_valid_m3 = processed_val_m3["Log Sale Price"]


# Take a look at the result
display(X_train_m3.head())
display(Y_train_m3.head())

display(X_valid_m3.head())
display(Y_valid_m3.head())
```

```
        Log Building Square Feet  Land Square Feet  Roof Material_1.0  \
130829                  7.870166            9632.0                1.0
193890                  7.002156            4183.0                1.0
30507                   6.851185            6632.0                1.0
91308                   7.228388            2256.0                1.0
131132                  7.990915           10480.0                1.0

        Roof Material_2.0  Roof Material_3.0  Roof Material_4.0  \
130829                0.0                0.0                0.0
193890                0.0                0.0                0.0
30507                 0.0                0.0                0.0
91308                 0.0                0.0                0.0
131132                0.0                0.0                0.0

        Roof Material_5.0  Roof Material_6.0
130829                0.0                0.0
193890                0.0                0.0
30507                 0.0                0.0
91308                 0.0                0.0
131132                0.0                0.0
```

```
130829    12.994530
193890    11.848683
30507     11.813030
91308     13.060488
131132    12.516861
Name: Log Sale Price, dtype: float64
```

```
        Log Building Square Feet  Land Square Feet  Roof Material_1.0  \
50636                   7.310550            3125.0                1.0
82485                   7.325808            4960.0                1.0
193966                  7.090077            7488.0                1.0
160612                  7.281386           12864.0                1.0
7028                    7.118016            4158.0                1.0

        Roof Material_2.0  Roof Material_3.0  Roof Material_4.0  \
50636                 0.0                0.0                0.0
```

```
82485                0.0                0.0                0.0
193966               0.0                0.0                0.0
160612               0.0                0.0                0.0
7028                 0.0                0.0                0.0

        Roof Material_5.0  Roof Material_6.0
50636                0.0                0.0
82485                0.0                0.0
193966               0.0                0.0
160612               0.0                0.0
7028                 0.0                0.0


50636     11.682668
82485     12.820655
193966     9.825526
160612    12.468437
7028      12.254863
Name: Log Sale Price, dtype: float64
```

In [158]:
```python
# Modeling STEP 2:  Create a Multiple Linear Regression Model

# Be sure to set fit_intercept to False, since we are incorporating one-hot-encoded data
linear_model_m3 = lm.LinearRegression(fit_intercept=False)

linear_model_m3.fit(X_train_m3, Y_train_m3)

# your code above this line to create regression model for Model 3

Y_predict_train_m3 = linear_model_m3.predict(X_train_m3)

Y_predict_valid_m3 = linear_model_m3.predict(X_valid_m3)
```

In [159]:
```python
# MODELING STEP 3:  Evaluate the RMSE for your model

# Training and test errors for the model (in its units of Log Sale Price)

training_error_log[2] = rmse(Y_predict_train_m3, Y_train_m3)
validation_error_log[2]= rmse(Y_predict_valid_m3, Y_valid_m3)

# Convert predictions and actuals back to original scale - undo log transformation
Y_train_m3_exp = np.exp(Y_train_m3)  # actual sale price for training data
Y_predict_train_m3_exp = np.exp(Y_predict_train_m3)  # predicted sale price for training data

Y_valid_m3_exp = np.exp(Y_valid_m3)  # actual sale price for validation data
Y_predict_valid_m3_exp = np.exp(Y_predict_valid_m3)  # predicted sale price for validation da

# Training and test errors for the model (in its original values before the log transform)
training_error[2] = rmse(Y_predict_train_m3_exp, Y_train_m3_exp)
validation_error[2] = rmse(Y_predict_valid_m3_exp, Y_valid_m3_exp)
```

```
print("3rd Model\nTraining RMSE (log): {}\nValidation RMSE (log): {}\n".format(training_error
print("3rd Model \nTraining RMSE: {}\nValidation RMSE: {}\n".format(training_error[2], valida
```

```
3rd Model
Training RMSE (log): 0.7483738042823385
Validation RMSE (log): 0.7479894313110084

3rd Model
Training RMSE: 242115.98698909127
Validation RMSE: 247142.56572707082
```

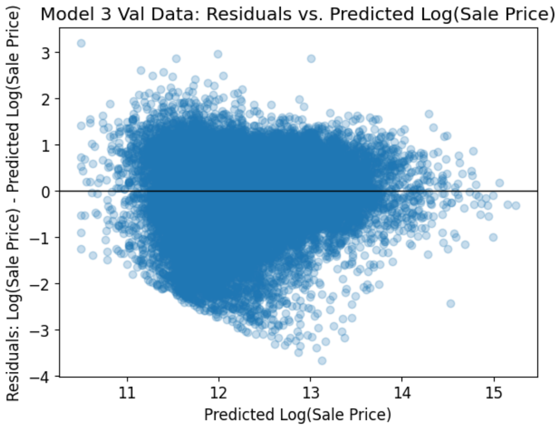
In [160]:
```
# MODELING STEP 4:  Conduct 5-fold cross validation for model and output RMSE

# create new model instance with fit_intercept=False
linear_model_m3_cv = lm.LinearRegression(fit_intercept=False)

# process entire training and validation data (train and val combined)
processed_full_m3 = process_data_m3(training_val_data)

# split into x and y for combined dataset
X_full_m3 = processed_full_m3.drop(columns="Log Sale Price")
Y_full_m3 = processed_full_m3["Log Sale Price"]

# your code above this line to use 5-fold cross-validation and output RMSE (in units of dolla

cv_error[2] = cross_validate_rmse(linear_model_m3_cv, X_full_m3, Y_full_m3)

print("3rd Model Cross Validation RMSE: {}".format(cv_error[2]))
```

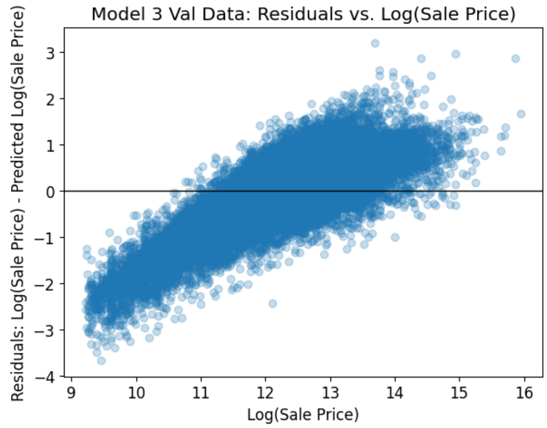
```
3rd Model Cross Validation RMSE: 242906.0742422669
```

In [161]:
```
# MODELING STEP 5:  Add a name for your 3rd model describing the features and run this cell t

model_names[2] = "M3: log(bsqft)+Roof+Land Sqft"


fig = go.Figure([
go.Bar(x = model_names, y = training_error, name="Training RMSE"),
go.Bar(x = model_names, y = validation_error, name="Validation RMSE"),
go.Bar(x = model_names, y = cv_error, name="Cross Val RMSE")
])

fig.update_yaxes(range=[180000,260000], title="RMSE")

fig
```

In [163]:
```python
# MODELING STEP 5 cont'd:  Plot 2 side-by-side residual plots (similar to Question 3, for val

fig, ax = plt.subplots(1,2, figsize=(15, 5))

# x_plt1: predicted log(Sale Price), y_plt1: residuals
x_plt1 = Y_predict_valid_m3  # predicted values for validation data (model 3)
y_plt1 = Y_valid_m3 - Y_predict_valid_m3  # residuals

# x_plt2: actual log(Sale Price), y_plt2: residuals
x_plt2 = Y_valid_m3  # actual values for validation data
y_plt2 = Y_valid_m3 - Y_predict_valid_m3  # residuals


ax[0].scatter(x_plt1, y_plt1, alpha=.25)
ax[0].axhline(0, c='black', linewidth=1)
ax[0].set_xlabel(r'Predicted Log(Sale Price)')
ax[0].set_ylabel(r'Residuals: Log(Sale Price) - Predicted Log(Sale Price)');
ax[0].set_title("Model 3 Val Data: Residuals vs. Predicted Log(Sale Price)")

ax[1].scatter(x_plt2, y_plt2, alpha=.25)
ax[1].axhline(0, c='black', linewidth=1)
ax[1].set_xlabel(r'Log(Sale Price)')
ax[1].set_ylabel(r'Residuals: Log(Sale Price) - Predicted Log(Sale Price)');
ax[1].set_title("Model 3 Val Data: Residuals vs. Log(Sale Price)")
```

Out[163]:
```
Text(0.5, 1.0, 'Model 3 Val Data: Residuals vs. Log(Sale Price)')
```


Model 3 Val Data: Residuals vs. Predicted Log(Sale Price)
Residuals: Log(Sale Price) - Predicted Log(Sale Price)
Predicted Log(Sale Price)


Model 3 Val Data: Residuals vs. Log(Sale Price)
Residuals: Log(Sale Price) - Predicted Log(Sale Price)
Log(Sale Price)

0.3.4 Question 5c

i). Comment on your RMSE and residual plots from Model 3 compared to the first 2 models.

ii). Are the residuals of your model still showing a trend that overestimates lower priced houses and underestimates higher priced houses? If so, how could you try to address this in the next round of modeling?

iii). If you had more time to improve your model, what would your next steps be?

i) Model 1 had the highest Validation RMSE (255,534) and Cross Validation RMSE (250,986). Model 2 improved over Model 1, reducing the Validation RMSE to 247,382 and the Cross Validation RMSE to 243,125. Model 3 achieved the best results, with a Validation RMSE of 247,143 and a slightly lower Cross Validation RMSE of 242,906 compared to Model 2. The RMSE values indicate that Model 3 has marginally improved over Model 2, demonstrating the benefit of including Land Square Feet as an additional predictor. While I am happy with the improvement and it was hard earned, I am slightly disappointed to see that Model 3 only reduced Model 1's Cross Validation RMSE by about 8,080. As we talked about earlier in the project, this is not a fair process for a number of reasons, but I expected more of a change with what we added to the models.

Similarly, the differences in the residual plots across the three models are quite subtle. The RMSE did improve between models, but not significantly, and this is reflected in the residual plots. That said, I do think that the plots from Model 1 are noticeably more spread out than the plots from Model 3, but just barely.

ii) Yes, the residuals are still showing a trend that overestimates lower priced houses and underestimates higher priced houses. This can most noticeably be seen in the Model 3 plot on the right. Here, residuals are systematically below zero at the low end of log(Sale Price) and above zero at the high end, indicating that lower value homes are being predicted to be higher value than they are. This can also be seen in the left side plot, where there is a clustering of homes with negative residuals on the lower value end of the plot that is not reflected in the higher value end, though I feel this is less visually apparent.

To address this issue in future rounds of modeling, we could try log transforming additional features. Log Sale Price is already log transformed, but other features, like Land Square Feet, could benefit from a log transformation as well. We could also try incorporating nonlinear features. A polynomial term, like a square or cubic term, would be interesting to experiment with and could allow the model to better capture curvature in the relationship between features and the target variable. Separate from changing how we use the data we have, we could also potentially add additional data, like racial or socioeconomic data, that could benefit the model.

iii) If I had unlimited time, I would start by exploring log transformations on more features, notably Land Square Feet. This could help improve linearity and reduce the impact of outliers. Then, I'd circle back on adding Age to model 3, I was surprised that it actually increased the RMSE and am curious about why. After that, I would experiment with polynomial or nonlinear features by squaring or cubing key predictors like Log Building Square Feet or Land Square Feet to try to capture curvature in the relationships. These nonlinear terms could help address the systematic over or underestimation seen in the residuals. I would also try to source additional data, the data available does not include any racial or socioeconomic data, which might be helpful.

0.4 Question 6: Evaluating the Model in Context

0.5 Question 6a

When evaluating your model, we used RMSE. In the context of estimating the value of houses, what does the residual mean for an individual homeowner? How does it affect them in terms of property taxes? Discuss the cases where residual is positive and negative separately.

Generally speaking, a residual is the difference between the actual value and the predicted value from the model. In the context of this question, a residual is the difference between the actual sale price or value and the predicted sale price or value of a home:

$$Residual = ActualSalePrice - PredictedSalePrice$$

When the residual is positive (actual value > predicted value), then the property was undervalued by the model. For the homeowner, this can affect the amount they pay in property taxes if the predicted value is used in the assessment process. This undervaluation would cause the homeowner to pay less in property taxes than they should. While this could benefit individual homeowners in the short term, it is an inaccuracy that undermines fairness and could ultimately be detrimental to homeowners if undervaluations are concentrated in certain areas, leading to lack of funding for public services in those areas. Even if the undervaluations don't lead to a lack of funding, this is an inequitable distribution of the tax burden.

When the residual is negative (actual value < predicted value), the property was overvalued by the model. This has the opposite impact on the homeowner, they may pay higher property taxes than they should. This creates an undue financial burden for the homeowner, regardless of their socioeconomic status, but will be particularly harmful for marginalized or low income homeowners. As with a positive residual, this is still an inequitable distribution of the tax burden.

0.6 Question 6b

Reflecting back on your exploration in Questions 5 and 6a, in your own words, what makes a model's predictions of property values for tax assessment purposes "fair"?

This question is open-ended and part of your answer may depend upon your specific model; we are looking for thoughtfulness and engagement with the material, not correctness.

Hint: Some guiding questions to reflect on as you answer the question above: What is the relationship between RMSE, accuracy, and fairness as you have defined it? Is a model with a low RMSE necessarily accurate? Is a model with a low RMSE necessarily "fair"? Is there any difference between your answers to the previous two questions? And if so, why?

This is a great question about the pragmatics and/or semantics of the word "fair." In my opinion, there are multiple meanings of "fair" and some of them do mean "accurate," but not necessarily all of them. In the relationship between RMSE, accuracy, and fairness, RMSE measures the average difference between the predicted and actual values. A lower RMSE indicates that the model's predictions are closer to the actual values, so it improves accuracy, but that doesn't necessarily equate fairness in my eyes. For example, a model could have a low RMSE overall, but systematically overvalue or undervalue certain properties based on hidden biases in the data, like properties in marginalized communities. To me, fairness requires that errors be evenly distributed across different subsets of the population or property types, not disproportionately affecting any group. So RMSE is an indicator or measure of accuracy, but not inherently a measure of fairness, fairness depends on how errors are distributed across the population.

So is a model with a low RMSE necessarily accurate and/or fair? I believe it is at least more accurate, a low RMSE suggests that the model is accurate **on average** because it reduces the error between predicted and actual values. This does not mean that a low RMSE model is accurate for everyone or every group, though. If the model systematically underestimates higher value homes and overestimates lower value homes, the overall RMSE could still appear low, but the model would not be accurate for every individual group. RMSE does not differentiate between systematic biases and random errors, it only measures overall error magnitude. Due to this, a low RMSE does not ensure that all groups or individuals are treated equitably or with fairness. For me, fairness is more about how evenly the residuals are distributed, not about the more global RMSE.

So in my opinion, there is a key difference in my answers. RMSE primarily measures accuracy, which is an aggregate error metric. Fairness considers the distribution of errors, and in this context, addresses whether specific groups are consistently over or under predicted. A model can minimize RMSE while still producing unfair predictions because RMSE alone isn't well suited for identifying systemic inequity.

0.7 Extra Credit Step 1: Creating Your Model

Complete the modeling steps (you can skip the cross validation step to save memory) in the cells below.

DO NOT ADD ANY EXTRA CELLS BELOW (for this part of the problem)

In [ ]:
```
# Modeling Step 1:  Process the Data



# Hint: You can either use your implementation of the One Hot Encoding Function from Project Pa
#from feature_func import *


...
# Optional:  Define any helper functions you need for one-hot encoding above this line


def process_data_ec(data):

    # You should start by only keeping values with Pure Market Filter = 1
    ...

    return data


# Process the data
processed_train_ec = ...

processed_val_ec = ...


X_train_ec = ...
Y_train_ec = ...

X_valid_ec = ...
Y_valid_ec = ...


# Take a look at the result
#display(X_train_ec.head())
#display(Y_train_ec.head())

#display(X_valid_m3.head())
#display(Y_valid_m3.head())
```

In [ ]:
```
# Modeling STEP 2:  Create a Multiple Linear Regression Model

# If you are are incorporating one-hot-encoded data, set the fit_intercept to False
```

```
...
# your code above this line to create regression model for Model 2

Y_predict_train_ec = ...

Y_predict_valid_ec = ...
```

In [ ]:
```
# MODELING STEP 3:  Evaluate the RMSE for your model


# Training and test errors for the model (in its original values before the log transform)
training_error_ec = ...
validation_error_ec = ...


print("Extra Credit Model\nTraining RMSE (log): {}\nValidation RMSE (log): {}\n".format(trainin
print("Extra Credit \nTraining RMSE: {}\nValidation RMSE: {}\n".format(training_error_ec, valida
```

In [ ]:
```
# Optional: Run this cell to visualize

fig = go.Figure([
go.Bar(x = ["Extra Credit Model"], y = [training_error_ec], name="Training RMSE"),
go.Bar(x = ["Extra Credit Model"], y = [validation_error_ec], name="Validation RMSE"),

])


fig
fig.update_yaxes(range=[140000,260000], title="RMSE")
```

In [ ]:
```
# MODELING STEP 5: Plot 2 side-by-side residual plots for validation data

fig, ax = plt.subplots(1,2, figsize=(15, 5))


x_plt1 = ...
y_plt1 = ...

x_plt2 = ...
y_plt2 = ...


ax[0].scatter(x_plt1, y_plt1, alpha=.25)
ax[0].axhline(0, c='black', linewidth=1)
ax[0].set_xlabel(r'Predicted Log(Sale Price)')
ax[0].set_ylabel(r'Residuals: Log(Sale Price) - Predicted Log(Sale Price)');
ax[0].set_title("EC Val Data: Residuals vs. Predicted Log(Sale Price)")

ax[1].scatter(x_plt2, y_plt2, alpha=.25)
```

```
ax[1].axhline(0, c='black', linewidth=1)
ax[1].set_xlabel(r'Log(Sale Price)')
ax[1].set_ylabel(r'Residuals: Log(Sale Price) - Predicted Log(Sale Price)');
ax[1].set_title("EC Val Data: Residuals vs. Log(Sale Price)")
```

0.8 Extra Credit Step 2: Explanation (Required for points on model above):

Explain what you did to create your model. What versions did you try? What worked and what didn't?

Comment on the RMSE and residual plots from your model. Are the residuals of your model still showing a trend that overestimates lower priced houses and underestimates higher priced houses?

Type your answer here, replacing this text.