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 watch Lecture 13 before attempting this question.**

0.1.1 Question 1a

Consider the following question: *"How much is a house worth?"*

Who might be interested in an answer to this question? Be sure to list at least three different parties (people or organizations) and state whether each one has an interest in seeing a low or high housing price.

Your response should be approximately 3 to 6 sentences.

Homebuyers, homeowners, and local governments each have different interests in a house's value. Homebuyers prefer lower prices so they can afford to purchase property and enter desirable neighborhoods. Homeowners and real estate agents benefit from higher prices since they gain equity or earn higher commissions. Meanwhile, local governments often favor higher prices because they generate more property tax revenue to fund community services.

0.1.2 Question 1b

Which of the following scenarios strikes 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.

Your response for each chosen scenario should be approximately 2 to 3 sentences.

Scenario A and Scenario C strike me as unfair. For Scenario A, homeowners would be overtaxed and potentially discouraged from improving or selling their homes, even though their assessed value does not reflect reality. For Scenario C, it is also unjust because systematically overvaluing inexpensive properties places a heavier tax burden on lower-income households while benefiting wealthier ones. This kind of bias worsens inequality and undermines trust in the fairness of property assessments.

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? What were the primary causes of these problems?

Your response should be approximately 2 to 4 sentences.

Note: Along with reading the paragraph above, you will need to watch Lecture 13 to answer this question.

The main problem with the earlier property tax system in Cook County was that it was regressive and racially discriminatory. The Chicago Tribune discovered that the Cook County Assessor's Office systemically overvalued low-priced homes and undervalued high-priced ones, shifting the tax burden onto working-class and non-white homeowners. These inequities arose from outdated, opaque assessment models, an appeals process favoring wealthier homeowners, and unequal data quality across neighborhoods. The result was a loss of public trust and a system that reinforced existing inequalities instead of ensuring fair taxation.

0.1.4 Question 1e

In addition to being regressive, how did the property tax system in Cook County place a disproportionate tax burden on non-white property owners?

Your response should be approximately 3 to 4 sentences.

The property tax system in Cook County placed a disproportionate burden on non-white property owners because their homes were consistently overvalued relative to their actual market prices, leading them to pay higher taxes. Many predominantly Black and Latino neighborhoods had less accurate or outdated property data, which caused the assessment model to make larger errors. Additionally, wealthier and predominantly white homeowners were more likely to appeal and reduce their assessments, while residents in marginalized communities had fewer resources or access to the appeals process. This combination reinforced racial inequalities and deepened economic disparities across Chicago's neighborhoods.

0.2 Question 4a

We can assess a model's performance and quality of fit with a plot of the residuals $(y - \hat{y})$ versus the observed outcomes (y).

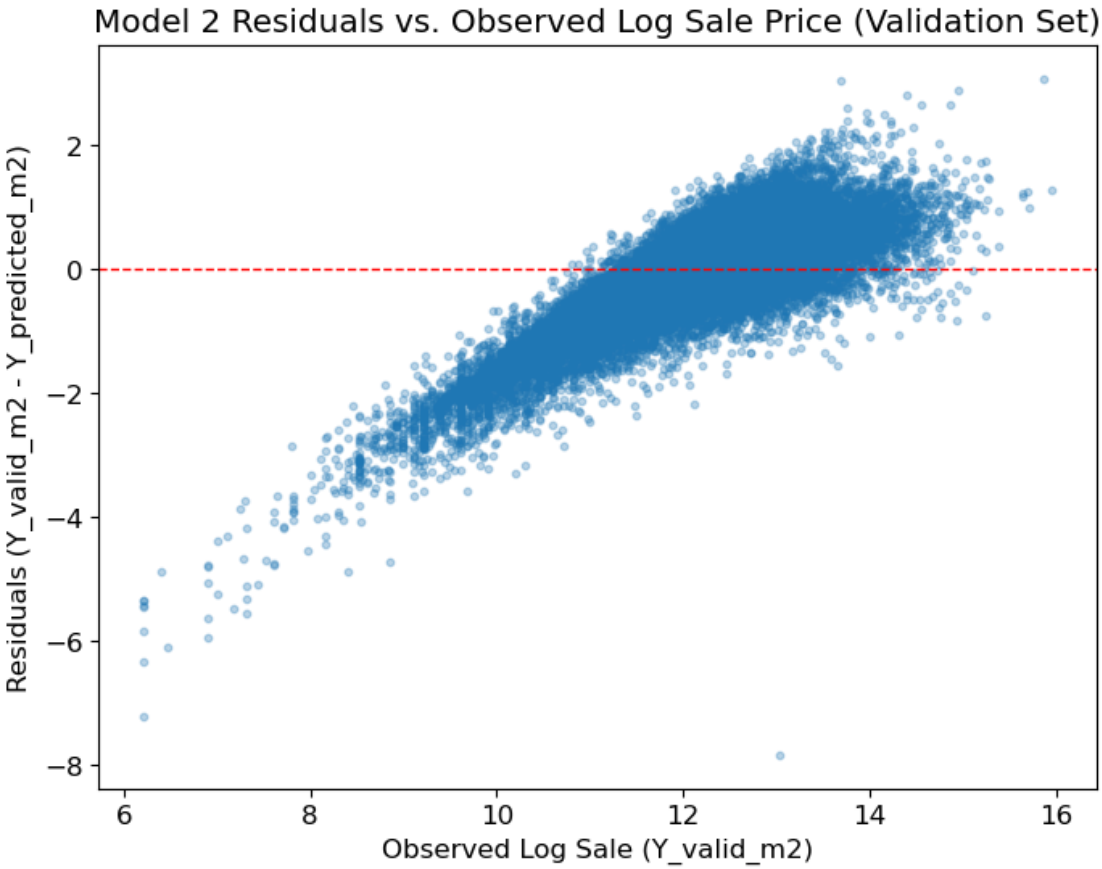
In the cell below, use `plt.scatter` (documentation) to plot the **model 2** residuals of `Log Sale Price` versus the original `Log Sale Price` values. For this part, you only need to plot the residuals and outcomes for the **validation data**.

- You should also **ensure that the dot size and opacity in the scatter plot are set appropriately** to reduce the impact of overplotting as much as possible. However, with such a large dataset, it is difficult to avoid overplotting entirely.

```
In [25]: residuals_m2 = Y_valid_m2 - Y_predicted_m2

         plt.figure(figsize=(8,6))
         plt.scatter(Y_valid_m2, residuals_m2, alpha = 0.3, s=10)

         plt.axhline(y=0, color='r', linestyle='--', linewidth=1)
         plt.xlabel("Observed Log Sale (Y_valid_m2)")
         plt.ylabel("Residuals (Y_valid_m2 - Y_predicted_m2)")
         plt.title("Model 2 Residuals vs. Observed Log Sale Price (Validation Set)")
         plt.show()
```


Model 2 Residuals vs. Observed Log Sale Price (Validation Set)
Residuals (Y_valid_m2 - Y_predicted_m2)
Observed Log Sale (Y_valid_m2)
2
0
−2
−4
−6
−8
6
8
10
12
14
16

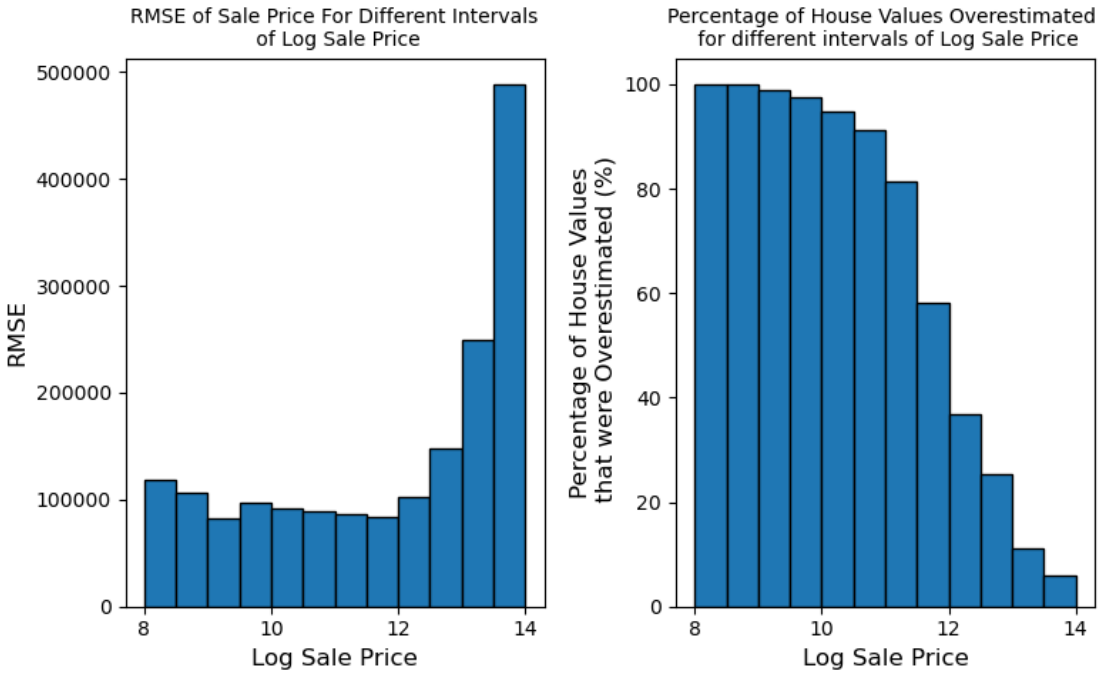
0.2.1 Question 6c

Using the functions above, we can generate visualizations of how the RMSE of sale price and proportion of overestimated houses vary for different intervals:

In [48]:
```
# RMSE plot
plt.figure(figsize = (8,5))
plt.subplot(1, 2, 1)
rmses = []
for i in np.arange(8, 14, 0.5):
    rmses.append(rmse_interval(preds_df, i, i + 0.5))
plt.bar(x = np.arange(8.25, 14.25, 0.5), height = rmses, edgecolor = 'black', width = 0.5)
plt.title('RMSE of Sale Price For Different Intervals\n of Log Sale Price', fontsize = 10)
plt.xlabel('Log Sale Price')
plt.yticks(fontsize = 10)
plt.xticks(fontsize = 10)
plt.ylabel('RMSE')

# Overestimation plot
plt.subplot(1, 2, 2)
props = []
for i in np.arange(8, 14, 0.5):
    props.append(prop_overest_interval(preds_df, i, i + 0.5) * 100)
plt.bar(x = np.arange(8.25, 14.25, 0.5), height = props, edgecolor = 'black', width = 0.5)
plt.title('Percentage of House Values Overestimated \n for different intervals of Log Sale Pri
plt.xlabel('Log Sale Price')
plt.yticks(fontsize = 10)
plt.xticks(fontsize = 10)
plt.ylabel('Percentage of House Values\n that were Overestimated (%)')

plt.tight_layout()
plt.show()
```



Which of the two plots above would be more useful in ascertaining whether the assessments tended to result in progressive or regressive taxation? Provide a brief explanation to support your choice of plot.

Then, explain whether your chosen plot aligns more closely aligns with scenario C or scenario D from `q1b`:

```
C. An assessment process that systematically overvalues inexpensive properties and undervalues expensive
D. An assessment process that systematically undervalues inexpensive properties and overvalues expensive
```

Your response should be approximately 3 to 4 sentences.

The plot on the right would be more useful because the left plot only shows the size of the errors, which tends to grow with price, but it can't tell whether cheap homes are systematically over- or under-assessed, so it would not be useful for judging progressive vs. regressive patterns. The overestimation plot shows the direction where low log-price bins have very high overestimation rates near 100%, and it steadily falls for high-price bins, meaning that high-price homes are often underestimated. This pattern matches Scenario C, where inexpensive properties are overvalued and expensive properties are undervalued, which is a regressive assessment pattern.

0.3 Question 7: Evaluating the Model in Context

0.4 Question 7a

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 a positive or negative residual affect them in terms of property taxes? Discuss the cases where the residual is positive and negative separately.

Your response should be approximately 2 to 4 sentences.

A residual represents the difference between a home's true value and the value predicted by the model. For an individual homeowner, a positive residual means their property's actual sale price is higher than the predicted value, which means the model undervalued their home, and they may end up paying lower property taxes than they should. Conversely, a negative residual means the model overvalued their home, leading to higher property taxes. In this context, residuals reflect how fair or unfair assessments may be, like how systemically positive residuals in wealthier areas and negative residuals in lower-income or minority neighborhoods could indicate regressive or biased taxation patterns.

0.5 Question 7b

Reflecting back on your exploration in Questions 6 and 7a, 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 on 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?

Your response should be approximate 1 to 2 paragraphs. Feel free to answer the questions in the hint to structure your answer.

A model's predictions are fair when they not only minimize overall error (low RMSE) but also avoid systematic biases that disadvantage certain groups of homeowners. While RMSES measures accuracy on average, fairness asks who the errors affect beyond their impact on accuracy. A model could have a low RMSE yet still be unfair if it consistently overvalues lower-priced homes and undervalues higher-priced ones, leading to regressive tax burdens. A fair property assessment model would therefore ensure that residuals are them. In short, fairness requires both technical accuracy and equitable outcomes, ensuring that no community bears a disproportionate share of adverse effects.

0.6 Question 8: Finding Better Metrics For Our Model

0.7 Question 8a

As discussed in Project A1, RMSE—while a widely used and powerful error metric introduced in earlier lectures—is not always the most appropriate choice when evaluating the *fairness* of a property appraisal system. In Question 7, you already encountered some of RMSE's limitations, particularly its tendency to disproportionately emphasize errors in high-priced properties due to the squaring of residuals.

In this question, rather than relying on RMSE, we will train and evaluate our model using another custom fairness metric. Specifically, we will examine the **Mean Absolute Percentage Error (MAPE)**, which measures the average error as a percentage of the true value. This allows us to better assess whether the model is making relatively fair predictions across different segments of the housing market.

In the code cells that follow, we'll explore how MAPE varies across price ranges—comparing the model's relative performance on inexpensive versus expensive housing. This helps identify whether the model systematically favors or disadvantages certain groups of properties based on their price.

Note: You'll notice that we're no longer using `lm.LinearRegression()`, but instead have started using `minimize`. This is because `scikit-learn` does not allow customization of the loss function in their standard linear regression model. The approach shown below provides an equivalent way to train a linear regression model using a custom error metric.

Warning: These cells take quite a long time to run. Please be patient, wait, and avoid restarting the kernel or rerunning these cells more than necessary.

In [49]:
```
data = pd.read_csv("cook_county_train.csv", index_col='Unnamed: 0')
trainX, trainY = feature_engine_final(data)
```

In [50]:
```
# X is the design matrix (including bias column), y is the vector of true outputs, theta is th
def mape(theta, X, y):
    y_pred = X @ theta  # compute predicted values using linear combination of features
    percentage_error = np.abs((y - y_pred) / y)  # calculate element-wise percentage errors
    return np.mean(percentage_error)  # return the MAPE
```

In [51]:
```
from scipy.optimize import minimize

# Add bias (intercept) column to the design matrix
trainX_with_bias = np.column_stack([np.ones(trainX.shape[0]), trainX])

# Initialize parameter vector with zeros
```

```
theta_0 = np.zeros(trainX_with_bias.shape[1])

# Use scipy's minimize to find weights that minimize MAPE
res = minimize(mape, theta_0, args=(trainX_with_bias, trainY), method='BFGS')

# Optimal weights after training
theta_opt = res.x

theta_opt
```

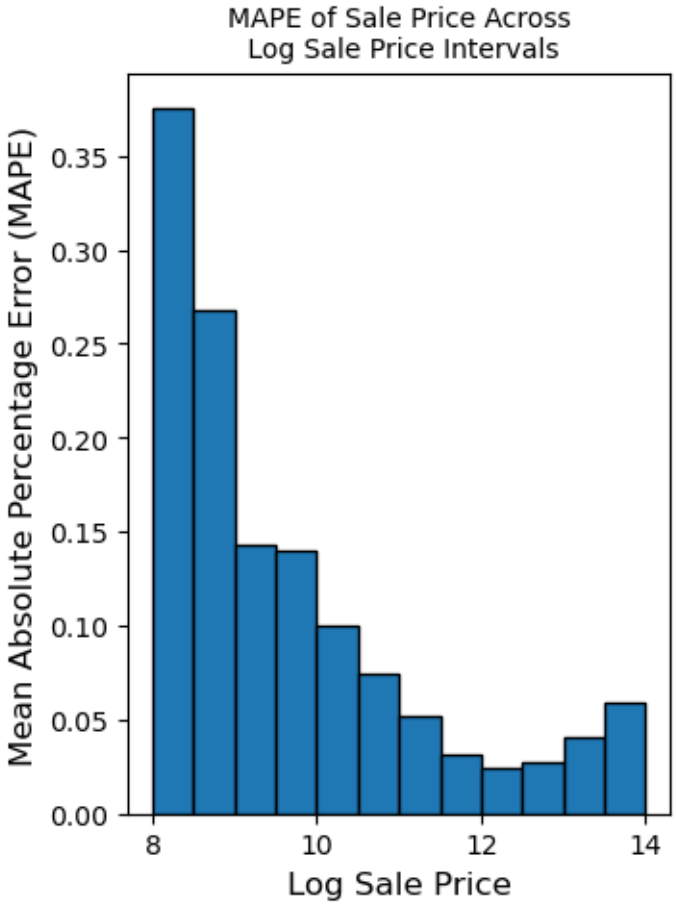
Out[51]:
```
array([ 5.73771915e+00,  8.63954545e-01,  1.82931753e-01,  1.78870639e-01,
        3.76014940e-01,  3.24532494e-01,  2.28688371e-01,  1.64247336e-01,
        5.50117761e-01, -5.30448248e-01,  3.56521952e-01,  1.76776019e-01,
        3.38993920e-01,  3.07327407e-01,  4.77785742e-01,  5.09118303e-01,
        3.70982383e-01,  5.56311152e-01,  3.26738627e-01,  3.24079332e-01,
        3.43154694e-01, -5.06349359e-01,  3.82999211e-01,  1.21247263e-01,
        4.92310769e-01,  6.42589764e-02,  2.78571113e-01,  4.41224199e-01,
       -3.15500630e-02, -4.00920113e-01, -9.38296258e-01,  3.50566036e-01,
        2.49941725e-01,  1.70318632e-01, -3.08261159e-01,  1.16940426e-02,
       -4.39263428e-02, -2.10427272e+00,  6.56722323e-01,  3.47624282e-01,
       -4.56311404e-01,  4.63207436e-01,  6.23118702e-01, -3.51159434e-02,
        3.12371409e-01, -5.27149840e-01, -1.66790952e+00,  5.29026144e-01,
       -1.96627196e-01, -1.16091794e-01,  3.48411637e-01,  5.65375717e-01,
        3.49401399e-03, -5.20128578e-01, -1.23113607e+00,  3.14714415e-01,
        7.80417898e-01, -1.43323079e+00, -1.41431327e-01, -6.50922243e-01,
        3.37138142e-01, -5.19738878e-01,  5.20565470e-01, -1.84597497e-01,
        1.86883686e-01,  4.11740093e-01,  5.50721701e-01,  3.81648426e-01,
        6.90732930e-01, -8.63961882e-01, -1.40916889e+00,  4.47790054e-01,
        1.17835677e+00, -1.17622035e+00,  7.25472837e-01, -1.23100687e-01,
        4.55540504e-01, -3.59386453e-01,  3.10773745e-01, -9.48570219e-01,
       -2.11288834e-01, -6.44467483e-01, -3.77730719e-01,  3.33095009e-01,
       -3.55761771e-02,  3.68348597e-01,  8.38058244e-01, -8.98430254e-01,
        7.62503588e-01,  2.39097941e-01, -5.22735952e-01, -1.28418486e+00,
        7.20342601e-01,  3.64108822e-01,  4.12441563e-02,  2.60595919e-01,
       -8.39915472e-02,  1.35308069e-01,  1.24969427e+00, -4.49434469e-01,
        4.42239746e-01,  1.87749777e-01, -3.26499307e-02,  2.05195015e-01,
        2.40750243e-02,  1.34768740e-01,  3.09494526e-01, -1.09801496e+00,
        1.89431459e-01,  1.16847690e+00,  6.60975618e-01,  1.11217484e-01,
       -3.40199107e-01,  6.29190589e-01,  1.53346153e-01,  2.04030514e-01,
        3.22639057e-01,  5.08956679e-01,  6.14686582e-01, -6.44502132e-01,
        3.64163613e-03, -1.21467105e+00,  1.98356290e-02,  0.00000000e+00])
```

In [52]:
```
new_preds_df = pd.DataFrame({
    'True Log Sale Price'     : trainY,
    'Predicted Log Sale Price': trainX_with_bias @ theta_opt,
    'True Sale Price'         : np.e ** trainY,
    'Predicted Sale Price'    : np.e ** (trainX_with_bias @ theta_opt)
})

plt.figure(figsize=(8, 5))
plt.subplot(1, 2, 1)
```

```
mape_values = []
for i in np.arange(8, 14, 0.5):
    mape_values.append(mape_interval(new_preds_df, i, i + 0.5))

plt.bar(x=np.arange(8.25, 14.25, 0.5), height=mape_values, edgecolor='black', width=0.5)
plt.title('MAPE of Sale Price Across\n Log Sale Price Intervals', fontsize=10)
plt.xlabel('Log Sale Price')
plt.ylabel('Mean Absolute Percentage Error (MAPE)')
plt.xticks(fontsize=10)
plt.yticks(fontsize=10);
```



What can you infer from this graph using our new custom error metric? Write your findings in the space below in three to five sentences. Consider addressing the following points:

- How does this metric differ from RMSE?

- What is the purpose of this custom error function?

- How does it relate to the idea of underpriced expensive housing mentioned in Question 6a?
- Why would this potentially be better than the RMSE in terms of the CCAO dataset?

MAPE differs from RMSE in that it measures percentage error

$$(\frac{|y - \hat{y}|}{y})$$

instead of squared dollar error, so it normalizes mistakes by a home's values with the goal of this custom loss to compare errors fairly across price tiers. For example, a 10k dollar miss on a 100k dollar home (10%) is much worse than a 10k dollar miss on a $1M home (1%). This lens shows that cheaper homes, where we often overestimate, incur larger percentage errors, while very expensive homes, though they drive big dollar RMSE, tend to have smaller percentage errors and are often underestimated. This is preferable to RMSE for the CCAO context because property taxes are levied as a percentage of value. What matters for fairness is systematic relative over- or under-assessment, which MAPE captures without letting a few luxury properties dominate the metric.