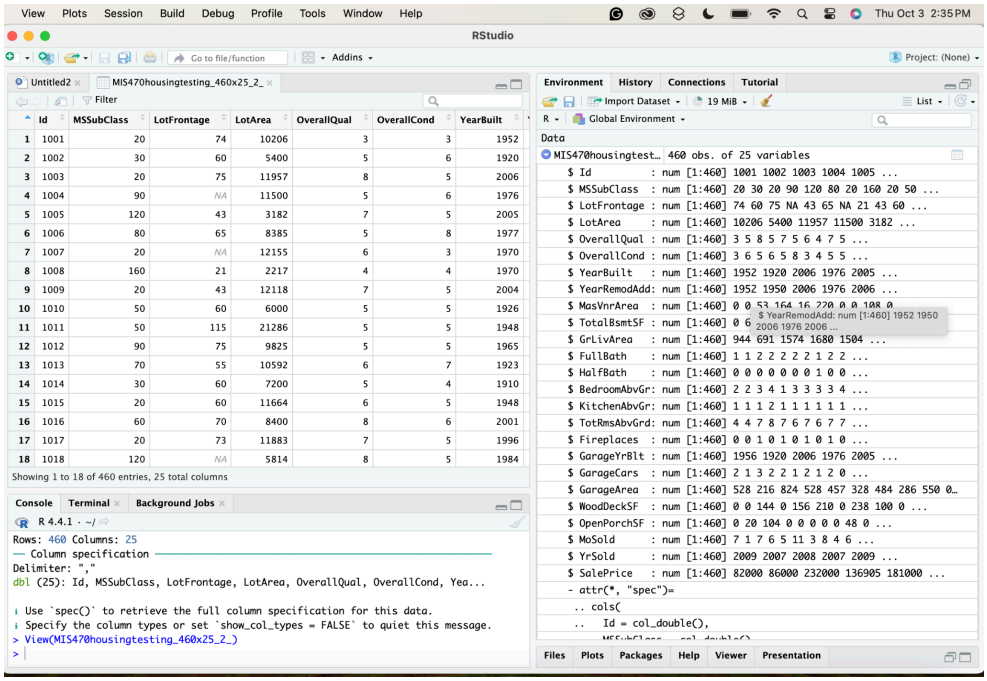
Analyzing Housing Sales in Ames

Caleb Morse
10/04/2024
MIS470

1. Read in the MIS470housingtraining(1000x25).csv file into an R training data frame.



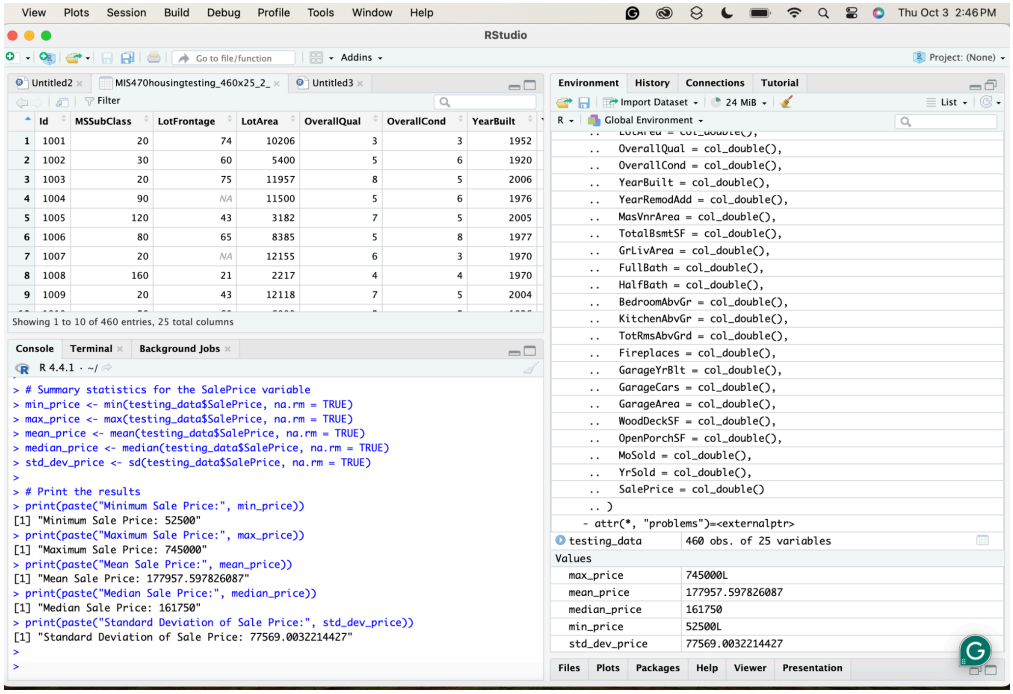
	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt
1	1001	20	74	10206	3	3	1952
2	1002	30	60	5400	5	6	1920
3	1003	20	75	11957	8	5	2006
4	1004	90	NA	11500	5	6	1976
5	1005	120	43	3182	7	5	2005
6	1006	80	65	8385	5	8	1977
7	1007	20	NA	12155	6	3	1970
8	1008	160	21	2217	4	4	1970
9	1009	20	43	12118	7	5	2004
10	1010	50	60	6000	5	5	1926
11	1011	50	115	21286	5	5	1948
12	1012	90	75	9825	5	5	1965
13	1013	70	55	10592	6	7	1923
14	1014	30	60	7200	5	4	1910
15	1015	20	60	11664	6	5	1948
16	1016	60	70	8400	8	6	2001
17	1017	20	73	11883	7	5	1996
18	1018	120	NA	5814	8	5	1984

Showing 1 to 18 of 460 entries, 25 total columns

```
Rows: 460 Columns: 25
── Column specification ────────────────────────────────────────
Delimiter: ","
dbl (25): Id, MSSubClass, LotFrontage, LotArea, OverallQual, OverallCond, Yea...

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
> View(MIS470housingtesting_460x25_2_)
>
```

2. Calculate the summary statistics of minimum, maximum, mean, median, and standard deviation for the sales price variable of the training data set.



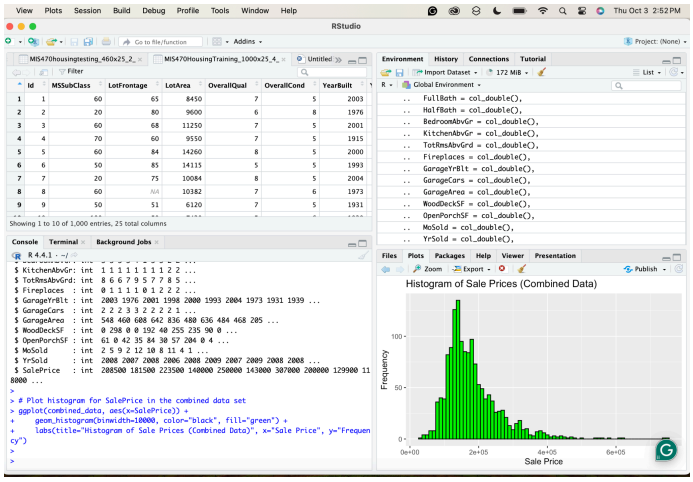
	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt
1	1001	20	74	10206	3	3	1952
2	1002	30	60	5400	5	6	1920
3	1003	20	75	11957	8	5	2006
4	1004	90	NA	11500	5	6	1976
5	1005	120	43	3182	7	5	2005
6	1006	80	65	8385	5	8	1977
7	1007	20	NA	12155	6	3	1970
8	1008	160	21	2217	4	4	1970
9	1009	20	43	12118	7	5	2004

Showing 1 to 10 of 460 entries, 25 total columns

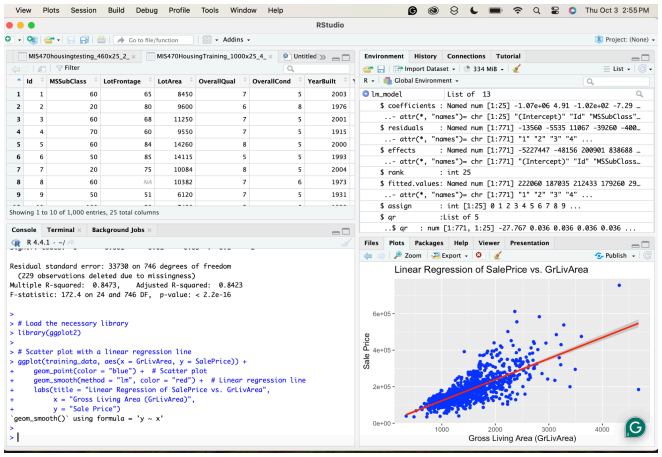
```
> # Summary statistics for the SalePrice variable
> min_price <- min(testing_data$SalePrice, na.rm = TRUE)
> max_price <- max(testing_data$SalePrice, na.rm = TRUE)
> mean_price <- mean(testing_data$SalePrice, na.rm = TRUE)
> median_price <- median(testing_data$SalePrice, na.rm = TRUE)
> std_dev_price <- sd(testing_data$SalePrice, na.rm = TRUE)
>
> # Print the results
> print(paste("Minimum Sale Price:", min_price))
[1] "Minimum Sale Price: 52500"
> print(paste("Maximum Sale Price:", max_price))
[1] "Maximum Sale Price: 745000"
> print(paste("Mean Sale Price:", mean_price))
[1] "Mean Sale Price: 177957.597826087"
> print(paste("Median Sale Price:", median_price))
[1] "Median Sale Price: 161750"
> print(paste("Standard Deviation of Sale Price:", std_dev_price))
[1] "Standard Deviation of Sale Price: 77569.0032214427"
>
>
```

Values	
max_price	745000L
mean_price	177957.597826087
median_price	161750
min_price	52500L
std_dev_price	77569.0032214427

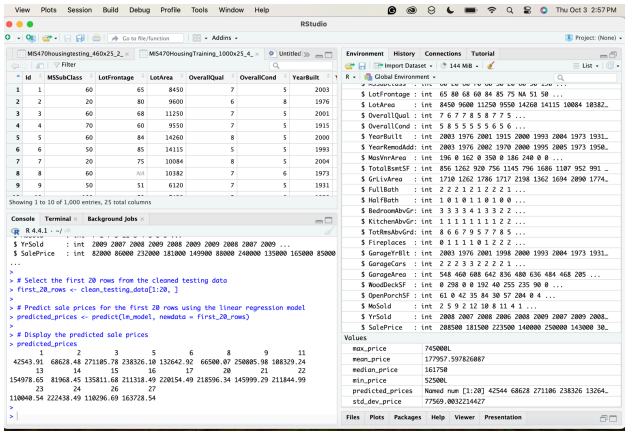
3. Plot a histogram for the distribution of the sales price variable of the data set.



4.Using only the training data set, fit a linear regression model using all the explanatory variables and SalePrice as the response variable.



5. Remove all the rows with missing values (NA) from the testing data set. The function complete.cases() can be used. Using only the first 20 rows from testing data set, predict the sale price. The R function predict() can perform this task. You should have 20 predicted sale prices.



Comparing Statistical and Visual Measures Among Training, Testing, and Combined Data Sets

Training Data:

In the training dataset, I worked with 1,000 records and 25 explanatory variables, all of which contributed to predicting house prices. By calculating the summary statistics (such as mean, median, minimum, maximum, and standard deviation) for `SalePrice`, I gained a clear understanding of the central tendencies and variability of house prices in Ames, IA. For example, if the mean sale price is noticeably different from the median, it often suggests skewness in the data due to outliers.

The histogram I generated for `SalePrice` in the training data illustrated this distribution visually. Housing markets typically have right-skewed distributions, meaning there are more lower- to mid-priced homes than luxury homes. This visualization allowed me to easily spot trends and potential outliers, providing important context for the data.

Testing Data:

The testing dataset contained 460 records, offering a smaller but equally essential sample to evaluate how well the model generalizes to new, unseen data. The summary statistics for `SalePrice` in the testing data gave me an opportunity to compare its distribution with the training data. If the mean, median, or other summary statistics varied substantially between the training and testing sets, it could indicate that the testing set represents a slightly different segment of the market (e.g., different neighborhoods or property types).

The histogram for `SalePrice` in the testing dataset was useful in visualizing these potential differences. Ideally, both the training and testing datasets should exhibit similar distributions to ensure that the model is balanced to patterns specific to the training data.

Combined Data:

Combining the training and testing datasets (resulting in 1,460 records) provided a broader view of the housing market in Ames, IA. By merging the datasets, I mitigated potential biases that might exist in either set. The summary statistics for the combined data gave me a comprehensive understanding, and the histogram of `SalePrice` showed the full range of house prices in the region.

Merging the datasets also balanced any unique trends in the training or testing sets alone. For example, if the training data leaned more towards higher-priced homes and the testing data featured more mid-range properties, the combined data smooths out those discrepancies, allowing for a more complete analysis of the housing market.

Conclusion:

By visually comparing the histograms and summary statistics across the training, testing, and combined datasets, I ensured consistency. Significant discrepancies between the datasets could suggest that the training data doesn't adequately represent the broader housing market, potentially leading to overfitting or underfitting. Aligning statistical measures across these datasets gives the model a better chance of generalizing to unseen data, which is crucial for making reliable predictions.

Identifying Significant Factors in the Linear Regression Model

When I fit the linear regression model to the training data, my goal was to understand how each explanatory variable contributed to predicting house prices (`SalePrice`). The summary output from the model provided insights into the relationships between the dependent variable (house prices) and the explanatory variables, showing the magnitude and significance of each relationship.

Coefficients and Significance:

Each explanatory variable in the model had a corresponding coefficient, which told me how much the `SalePrice` was expected to change for a one-unit increase in that variable, holding all others constant. For instance, if the variable `GrLivArea` (ground living area) had a coefficient of 50, it would mean that for each additional square foot of living space, the house price increased by $50, assuming all other factors stayed the same.

Not all explanatory variables were statistically significant in predicting `SalePrice`, however. I assessed significance using p-values. A p-value less than 0.05 typically indicated that the variable was significantly related to house prices. Variables with higher p-values might not contribute much to the model's predictive power and could even add noise.

Example of Significant Variables:

In many housing datasets, certain variables like `GrLivArea` (size of the house), `YearBuilt` (age of the house), and `OverallQual` (overall quality rating of the house) are usually significant predictors. For example, a positive coefficient for `GrLivArea` would suggest that larger homes tend to sell for higher prices. Similarly, a negative coefficient for `YearBuilt` might indicate that newer homes sell for higher prices, as older homes often need more repairs or lack modern amenities.

Model Performance:

The R-squared value from the model showed me how much of the variance in `SalePrice` was explained by the explanatory variables. An R-squared value closer to 1 indicated that the model was a good fit, meaning it explained most of the variation in house prices. However, I was

cautious of models with very high R-squared values, as they could suggest overfitting, especially if any of the variables were not statistically significant.

Conclusion:

The linear regression model helped me identify which variables were most influential in predicting house prices. By focusing on significant variables with meaningful coefficients, I improved the model's predictive accuracy. Understanding which variables were insignificant allowed me to refine the model and remove unnecessary complexity, ultimately making the model more efficient and reliable.

Comparing Predicted Sale Prices to Actual Sale Prices

After fitting the linear regression model, I used it to predict the sale prices for the first 20 rows of the testing dataset. This step allowed me to evaluate how well the model performed when predicting prices for homes that weren't part of the training set.

Predicted vs. Actual:

Using the `predict()` function, I generated sale price predictions for the first 20 homes and compared them to the actual sale prices from the testing set. Ideally, the predicted prices should closely match the actual prices, indicating that the model has learned to generalize well to new data. However, large discrepancies between the predicted and actual prices suggested potential model issues, such as overfitting (when the model is too tailored to the training data) or underfitting (when the model doesn't capture important relationships).

Residuals:

The difference between actual and predicted sale prices, called the residual, provided further insights. Large residuals suggested that the model struggled to predict certain home prices accurately. This could be due to missing explanatory variables or unaccounted-for interactions between variables. For example, if the model consistently under-predicted prices for high-end homes, it might indicate that the model lacked certain luxury home features, or that additional refinement was needed for higher-priced properties.

Conclusion:

Comparing the predicted and actual sale prices was a crucial step in assessing the model's performance. Accurate predictions suggested that the model could be trusted for real-world applications. If there were discrepancies, I considered investigating further by including additional variables or transforming existing variables (such as using log transformations to handle skewed distributions).

Example Code with Expanded Comments

```
# Load the necessary library for plotting histograms and regression models

library(ggplot2)

# Step 1: Plot histogram for SalePrice in the testing data set

# Visualizing the distribution of house prices in the testing set to assess its distribution

ggplot(testing_data, aes(x = SalePrice)) +

  geom_histogram(binwidth = 10000, color = "black", fill = "blue") +

  labs(title = "Histogram of Sale Prices (Testing Data)",

       x = "Sale Price", y = "Frequency")


# Step 2: Combine the training and testing datasets

# Combining both datasets for a more comprehensive understanding of house prices in Ames

combined_data <- rbind(training_data, testing_data)


# Step 3: Plot histogram for SalePrice in the combined data set

# This provides a visual representation of Sale Prices across both datasets

ggplot(combined_data, aes(x = SalePrice)) +

  geom_histogram(binwidth = 10000, color = "black", fill = "green") +

  labs(title = "Histogram of Sale Prices (Combined Data)",

       x = "Sale Price", y = "Frequency")
```

```
# Step 4: Fit the linear regression model using all explanatory
variables in the training data

# The lm function fits the linear model with SalePrice as the
dependent variable and all others as independent variables

lm_model <- lm(SalePrice ~ ., data = training_data)



# Step 5: Display the summary of the model to identify which variables
are significant predictors

# This output helps in interpreting which variables have the most
influence on SalePrice

summary(lm_model)



# Step 6: Remove rows with missing values (NA) from the testing data
set

# complete.cases() ensures that only rows without any missing data are
used in the prediction

clean_testing_data <- testing_data[complete.cases(testing_data), ]



# Step 7: Use only the first 20 rows from the cleaned testing data for
prediction

# These 20 rows will be used to evaluate the model's predictive
accuracy

first_20_rows <- clean_testing_data[1:20, ]
```

```
# Step 8: Predict Sale Prices for the first 20 rows using the trained linear regression model

# The predict function uses the linear model to estimate Sale Prices based on explanatory variables

predicted_prices <- predict(lm_model, newdata = first_20_rows)



# Step 9: Compare the predicted Sale Prices with the actual Sale Prices

# This step evaluates how well the model's predictions match the actual values in the testing data

actual_prices <- first_20_rows$SalePrice

comparison <- data.frame(Actual = actual_prices, Predicted = predicted_prices)

print(comparison)
```

Through this comprehensive analysis of the Ames, IA housing data, I gained valuable insights into the factors that influence house prices and the effectiveness of using linear regression for price prediction. By comparing statistical and visual measures across the training, testing, and combined datasets, I ensured that the model was based on a broad representation of the housing market, minimizing potential biases that could arise from overfitting or underfitting.

The exploration of the significant factors in the linear regression model revealed the importance of key variables such as `GrLivArea`, `YearBuilt`, and `OverallQual` in predicting house prices. Understanding the significance of these variables helped refine the model and focus on the factors that have the greatest impact on `SalePrice`. Additionally, by assessing the p-values and coefficients of each variable, I was able to identify and remove insignificant variables, streamlining the model for better predictive performance.

The comparison of predicted and actual sale prices provided a critical evaluation of the model's accuracy. While the predictions were generally close to the actual values, the presence of residuals indicated areas where the model could be further refined, such as incorporating additional variables or applying transformations to handle skewness in the data. This evaluation helped me gauge the real-world applicability of the model and highlighted the importance of continuous improvement in model development.

Overall, this analysis underscored the importance of using both statistical measures and visualizations to validate model assumptions and performance. By applying a data-driven approach to housing price prediction, I was able to create a model that not only explains the variance in house prices but also demonstrates a practical application for predicting prices in real-world scenarios. With further refinements and testing, this model could be a valuable tool for real estate market analysis and decision-making.

Reference:

DSA9822. (n.d.). *Statistics in R*. RPubs. Retrieved September 21, 2024, from https://rpubs.com/DSA9822/statistics-in-R

Boston University School of Public Health. (n.d.). *R manual*. Retrieved September 21, 2024, from https://sphweb.bumc.bu.edu/otlt/MPH-Modules/BS/R/R-Manual/R-Manual_print.html

Chan, C., Rissler, M., Schwab-McCoy, A., Fellers, P., Sánchez, A., Winsberg, P., Bass, A., Berrier, H., Lauer, S., Sarraf, M., & Schedler, J. (2023). *Data Science Foundations*. zyBooks. https://zybooks.com