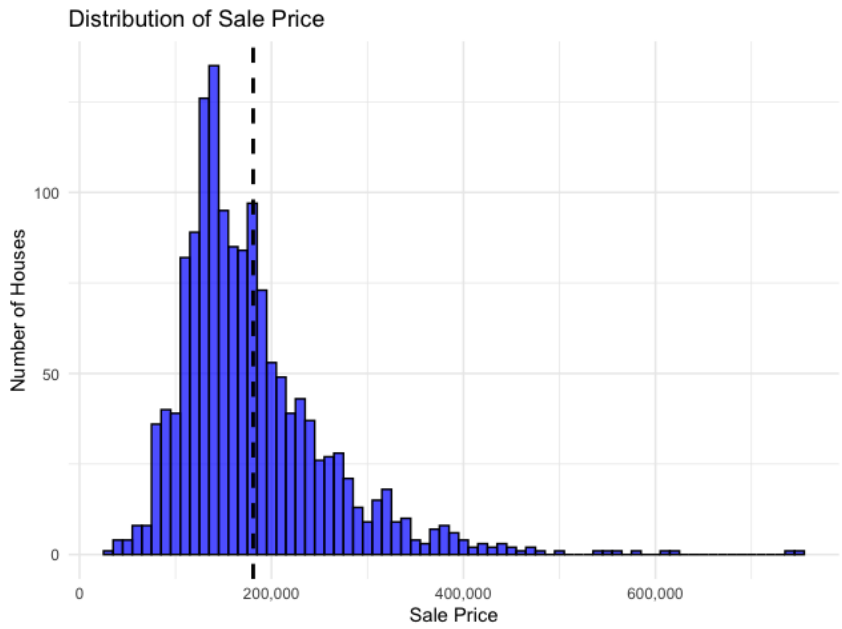
EDA Assignment

I used R to replicate the python code for the EDA Assignment

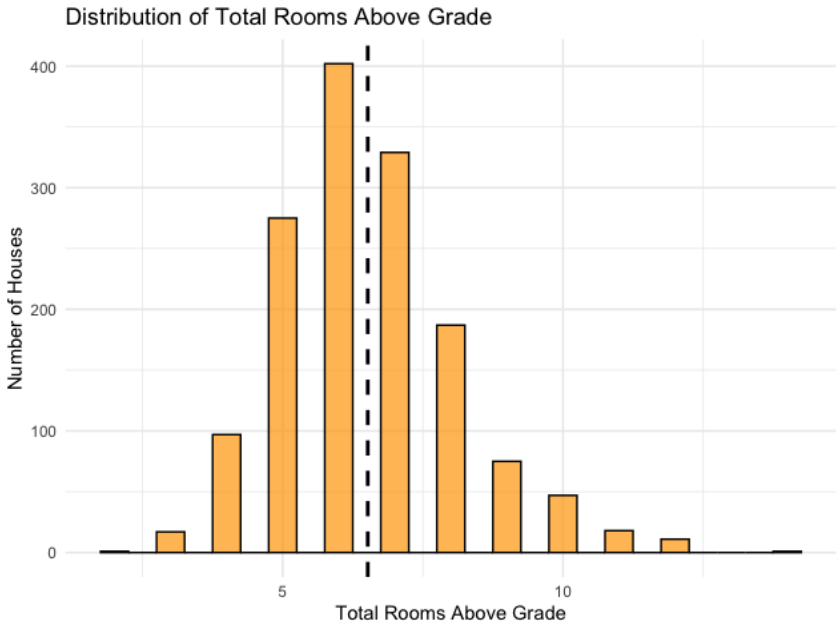
1) Sales Histogram




```
> library(ggplot2)
> library(scales)
>
>
> sale_price_mean <- mean(df$SalePrice, na.rm = TRUE)
>
>
> ggplot(df, aes(x = SalePrice)) +
+     geom_histogram(binwidth = 10000, fill = "blue", color = "black", alpha = 0.7) +
+     geom_vline(aes(xintercept = sale_price_mean), color = "black", linetype = "dashed", size = 1) +
+     labs(
+         title = "Distribution of Sale Price",
+         x = "Sale Price",
+         y = "Number of Houses"
+     ) +
+     scale_x_continuous(labels = scales::label_comma()) +
+     theme_minimal()
```

```
> print_stats <- function(df, column) {
+     cat("Mean:               ", mean(df[[column]], na.rm = TRUE), "\n")
+     cat("Median:             ", median(df[[column]], na.rm = TRUE), "\n")
+     cat("Standard Deviation:", sd(df[[column]], na.rm = TRUE), "\n")
+ }
>
>
> print_stats(df, "SalePrice")
Mean:               180921.2
Median:             163000
Standard Deviation: 79442.5
>
```

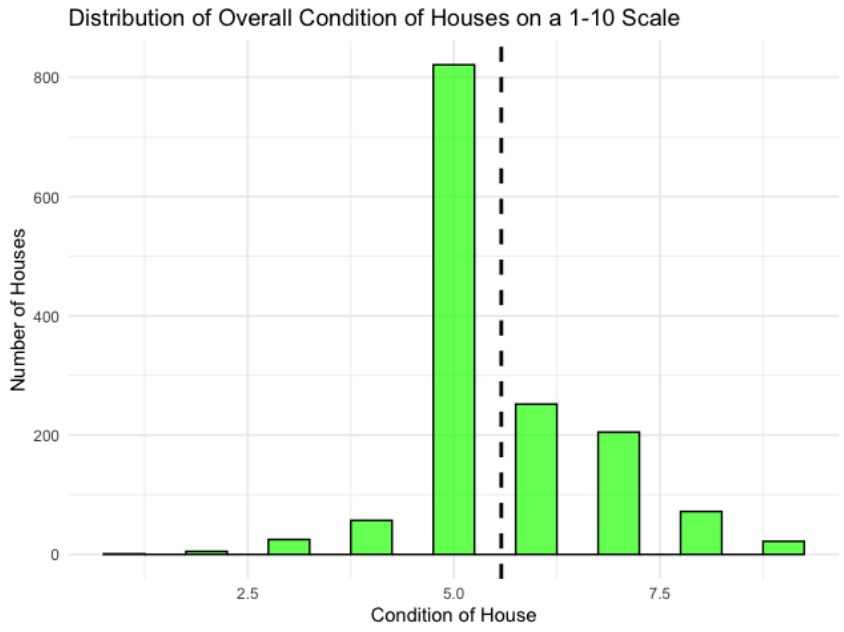
2) Total Rooms Histogram




```
> rooms_mean <- mean(df$TotRmsAbvGrd, na.rm = TRUE)
>
> ggplot(df, aes(x = TotRmsAbvGrd)) +
+     geom_histogram(binwidth = 0.5, fill = "orange", color = "black", alpha = 0.7) +
+     geom_vline(aes(xintercept = rooms_mean), color = "black", linetype = "dashed", size = 1) +
+     labs(
+         title = "Distribution of Total Rooms Above Grade",
+         x = "Total Rooms Above Grade",
+         y = "Number of Rooms"
+     ) +
+     scale_x_continuous(labels = scales::label_comma()) +
+     theme_minimal()
>
```

```
> print_stats <- function(df, column) {
+     cat("Mean:               ", mean(df[[column]], na.rm = TRUE), "\n")
+     cat("Median:             ", median(df[[column]], na.rm = TRUE), "\n")
+     cat("Standard Deviation:", sd(df[[column]], na.rm = TRUE), "\n")
+ }
>
>
> print_stats(df, "TotRmsAbvGrd")
Mean:               6.517808
Median:             6
Standard Deviation: 1.625393
```

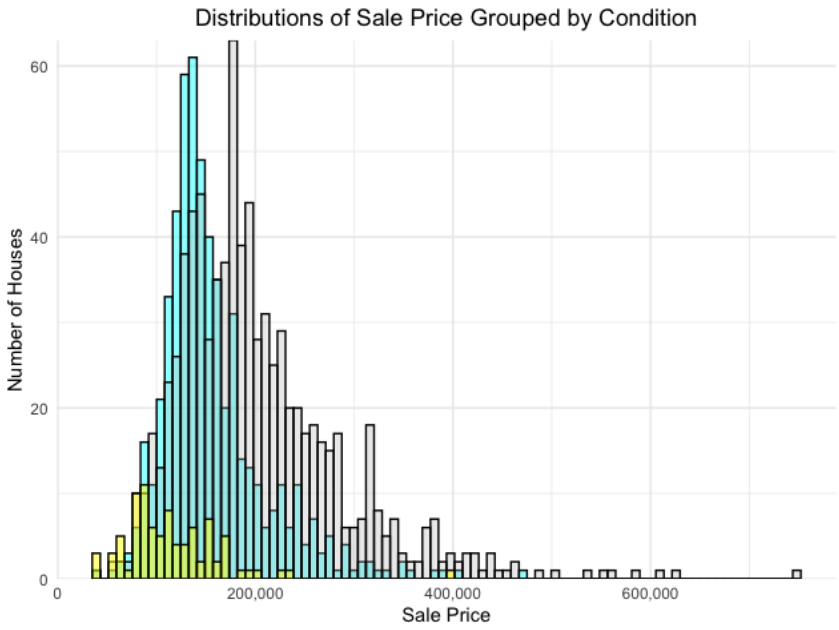
3) Histogram for Overall Condition




```
> overall_cond_mean <- mean(df$OverallCond, na.rm = TRUE)
>
>
> ggplot(df, aes(x = OverallCond)) +
+     geom_histogram(binwidth = 0.5, fill = "green", color = "black", alpha = 0.7) +
+     geom_vline(aes(xintercept = overall_cond_mean), color = "black", linetype = "dashed", size = 1) +
+     labs(
+         title = "Distribution of Overall Condition",
+         x = "Overall Condition",
+         y = "Number of Rooms"
+     ) +
+     scale_x_continuous(labels = scales::label_comma()) +
+     theme_minimal()
>
>
```

```
> print_stats <- function(df, column) {
+     cat("Mean:               ", mean(df[[column]], na.rm = TRUE), "\n")
+     cat("Median:             ", median(df[[column]], na.rm = TRUE), "\n")
+     cat("Standard Deviation:", sd(df[[column]], na.rm = TRUE), "\n")
+ }
>
> print_stats(df, "OverallCond")
Mean:               5.575342
Median:             5
Standard Deviation: 1.112799
```

4) Distributions of Sale Price Grouped by Conditions



```
> below_average_condition <- df[df$OverallCond < 5, ]
> average_condition <- df[df$OverallCond == 5, ]
> above_average_condition <- df[df$OverallCond > 5, ]
>
>
> bin_width <- floor(median(df$SalePrice, na.rm = TRUE) / 20)
> breaks <- seq(min(df$SalePrice, na.rm = TRUE), max(df$SalePrice, na.rm = TRUE), by = bin_width)
>
>
> library(ggplot2)
>
> ggplot() +
+     geom_histogram(data = above_average_condition, aes(x = SalePrice),
+                    breaks = breaks, fill = "cyan", alpha = 0.5, color = "black") +
+     geom_histogram(data = average_condition, aes(x = SalePrice),
+                    breaks = breaks, fill = "gray", alpha = 0.3, color = "black") +
+     geom_histogram(data = below_average_condition, aes(x = SalePrice),
+                    breaks = breaks, fill = "yellow", alpha = 0.5, color = "black") +
+     labs(
+         title = "Distributions of Sale Price Grouped by Condition",
+         x = "Sale Price",
+         y = "Number of Houses"
+     ) +
+     theme_minimal() +
+     theme(plot.title = element_text(hjust = 0.5)) +
+     scale_y_continuous(expand = c(0, 0)) +
+     scale_x_continuous(labels = scales::label_comma()) +
+     guides(fill = guide_legend(title = "Condition"))
>
```

5) Explore Correlations

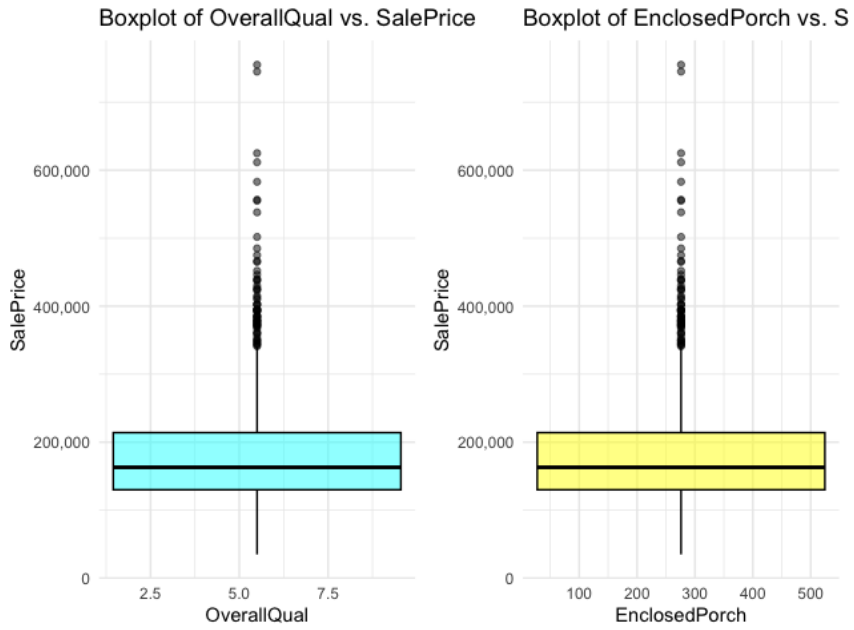
Positive Correlation:

```
> numeric_cols <- sapply(df, is.numeric)
> correlations <- cor(df[, numeric_cols], use = "complete.obs")
>
> cor_with_saleprice <- correlations["SalePrice", ]
> cor_with_saleprice <- cor_with_saleprice[!names(cor_with_saleprice) %in% "SalePrice"]
>
> most_correlated <- names(which.max(cor_with_saleprice))
> highest_correlation <- max(cor_with_saleprice)
>
> cat("The column most positively correlated with SalePrice is:", most_correlated,
+     "with a Pearson correlation of:", highest_correlation, "\n")
The column most positively correlated with SalePrice is: OverallQual with a Pearson correlation of: 0.7978807
>
```

Negative Correlation:

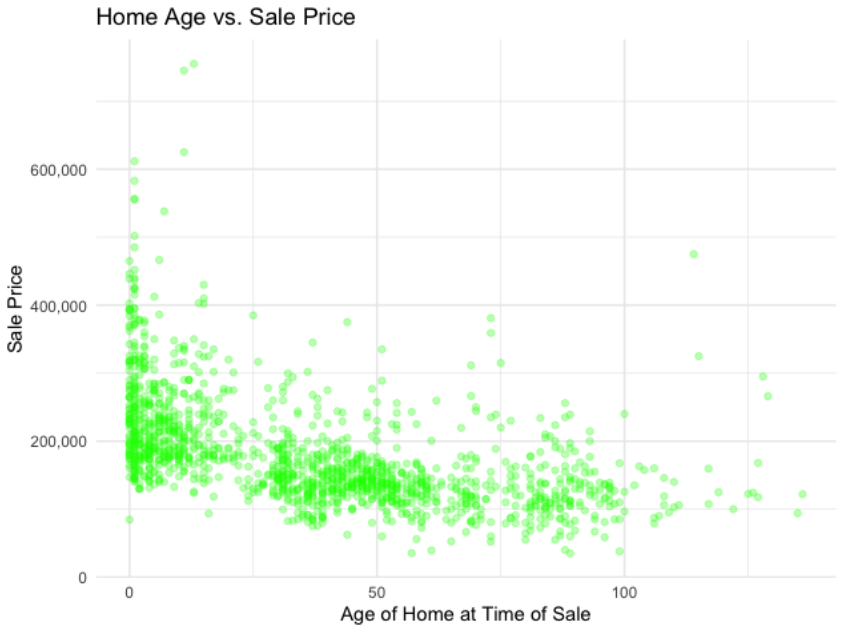
```
> most_negatively_correlated <- names(which.min(cor_with_saleprice))
> lowest_correlation <- min(cor_with_saleprice)
>
> cat("The column most negatively correlated with SalePrice is:", most_negatively_correlated,
+     "with a Pearson correlation of:", lowest_correlation, "\n")
The column most negatively correlated with SalePrice is: EnclosedPorch with a Pearson correlation of: -0.1548
432
>
```

Box Plot:



```
> plot1 <- ggplot(df, aes_string(x = max_corr_column, y = "SalePrice")) +
+     geom_boxplot(fill = "cyan", color = "black", alpha = 0.5) +
+     labs(
+         title = paste("Boxplot of", max_corr_column, "vs. SalePrice"),
+         x = max_corr_column,
+         y = "SalePrice"
+     ) +
+     theme_minimal() +
+     scale_y_continuous(labels = scales::label_comma())
>
>
> plot2 <- ggplot(df, aes_string(x = min_corr_column, y = "SalePrice")) +
+     geom_boxplot(fill = "yellow", color = "black", alpha = 0.5) +
+     labs(
+         title = paste("Boxplot of", min_corr_column, "vs. SalePrice"),
+         x = min_corr_column,
+         y = "SalePrice"
+     ) +
+     theme_minimal() +
+     scale_y_continuous(labels = scales::label_comma())
>
> grid.arrange(plot1, plot2, ncol = 2)
```

6) Engineer and Explore a New Feature



```
> df$Age <- df$YrSold - df$YearBuilt
>
>
> ggplot(df, aes(x = Age, y = SalePrice)) +
+     geom_point(alpha = 0.3, color = "green") +
+     labs(title = "Home Age vs. Sale Price",
+          x = "Age of Home at Time of Sale",
+          y = "Sale Price") +
+     scale_y_continuous(labels = scales::label_comma()) +
+     theme_minimal()
```