ALY6015 Intermediate Analytics

Ames Housing Dataset:An Analysis On Property Features and Sale Prices

Module 1 – R Practice Assignment

Student Name: Godliver Alangyam
NUID: 002807730

Date: 01/11/2025

#PART 1

##Introduction:

The Ames Housing dataset is a collection of real estate properties sold in Ames, Iowa, in 2010. It includes detailed information about various attributes of each property, making it a valuable resource for real estate analysis and market trend studies. Here are the key aspects covered in the dataset; Property Identification, Lot Information, Location and Accessibility, Building Characteristics, Interior Features, Garage and Parking, Additional Features, and Sale Information.

##OVERVIEW The dataset includes properties with various architectural styles, construction years, and sale prices, highlighting the diversity in the real estate market. For instance, you might find entries for: - A single-family home built in 2005 with a high overall quality rating, a large garage, and a spacious lot. - A townhouse built in 1998 with a moderate quality rating, a small yard, and proximity to main roads.

The dataset contains 2,930 rows.Each row representing an individual property sale in Ames, Iowa, during the year 2010. Each row includes detailed information about the property, such as its physical characteristics, location, and sale details.

I then went ahead to load the libraries needed for my analysis and visualization

##PART 2 : Data Cleaning and Initial Exploratory Data Analysis

After reading the dataset into R, I used clean_names to change the names of the columns to make them simpler and easy to use.The dataset was loaded into R using the read_csv function from the readr package. The dataset consists of 2,930 rows and 81 columns.The dataset includes a mix of numeric and character variables, each representing different aspects of the properties.

```
#Reading dataset
ames <- read_csv("AmesHousing.csv")
```

```
## Rows: 2930 Columns: 82
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (45): PID, MS SubClass, MS Zoning, Street, Alley, Lot Shape, Land Contou...
## dbl (37): Order, Lot Frontage, Lot Area, Overall Qual, Overall Cond, Year Bu...
##
## 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.
```

```
ames <- clean_names(ames)
```

I then performed a summary statistics on the dataset. Initial exploration involved reviewing the data types, number of rows, and columns.

```
#Summary of the dataset
data_summary <- summary(ames)
```

Then I took a look at the structure of the data. I had to comment it out to reduce the number of pages in my report.

```
#data_structure <- str(ames)
```

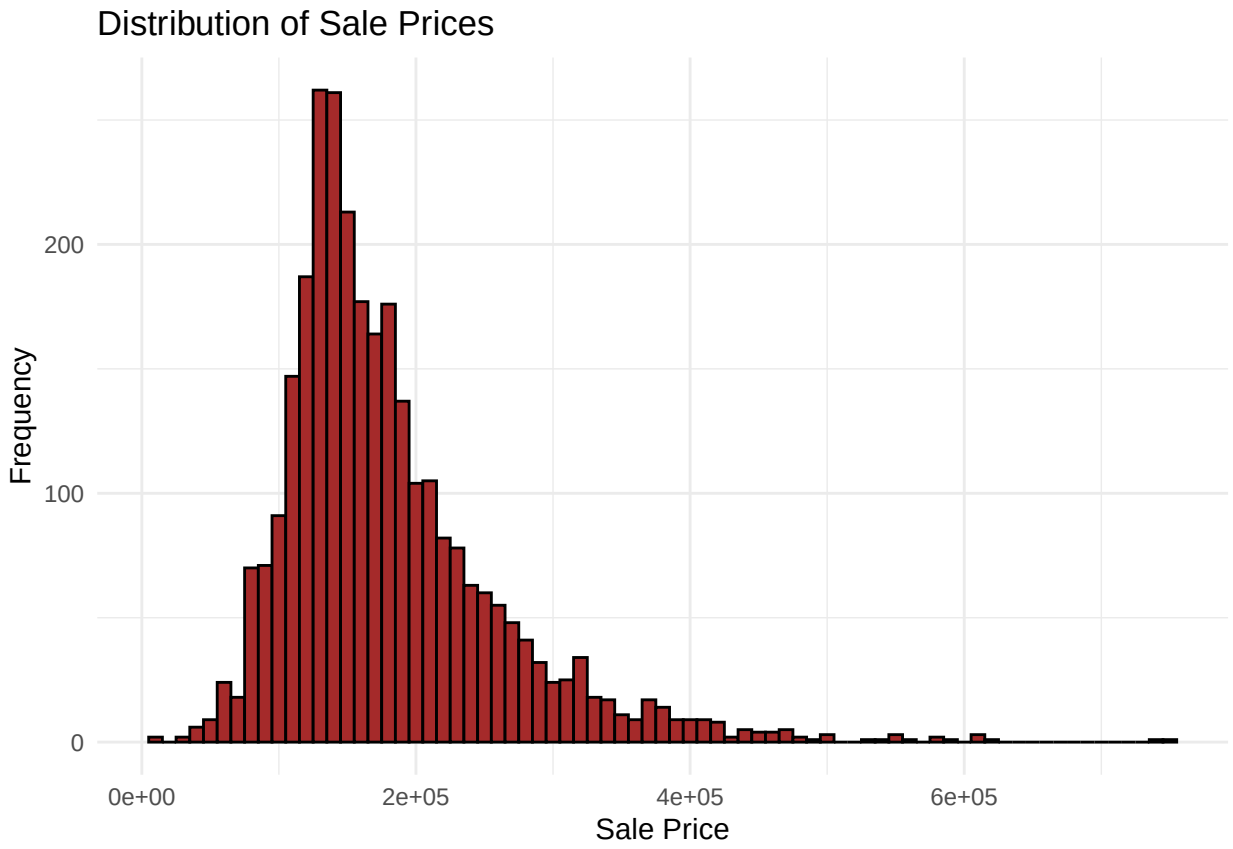
##Part 3: Visualization [1][2][3][4][5]

I was curious to know the distribution of sale prices and houses above ground living area by plotting a histogram to show the frequencies for the two variables.

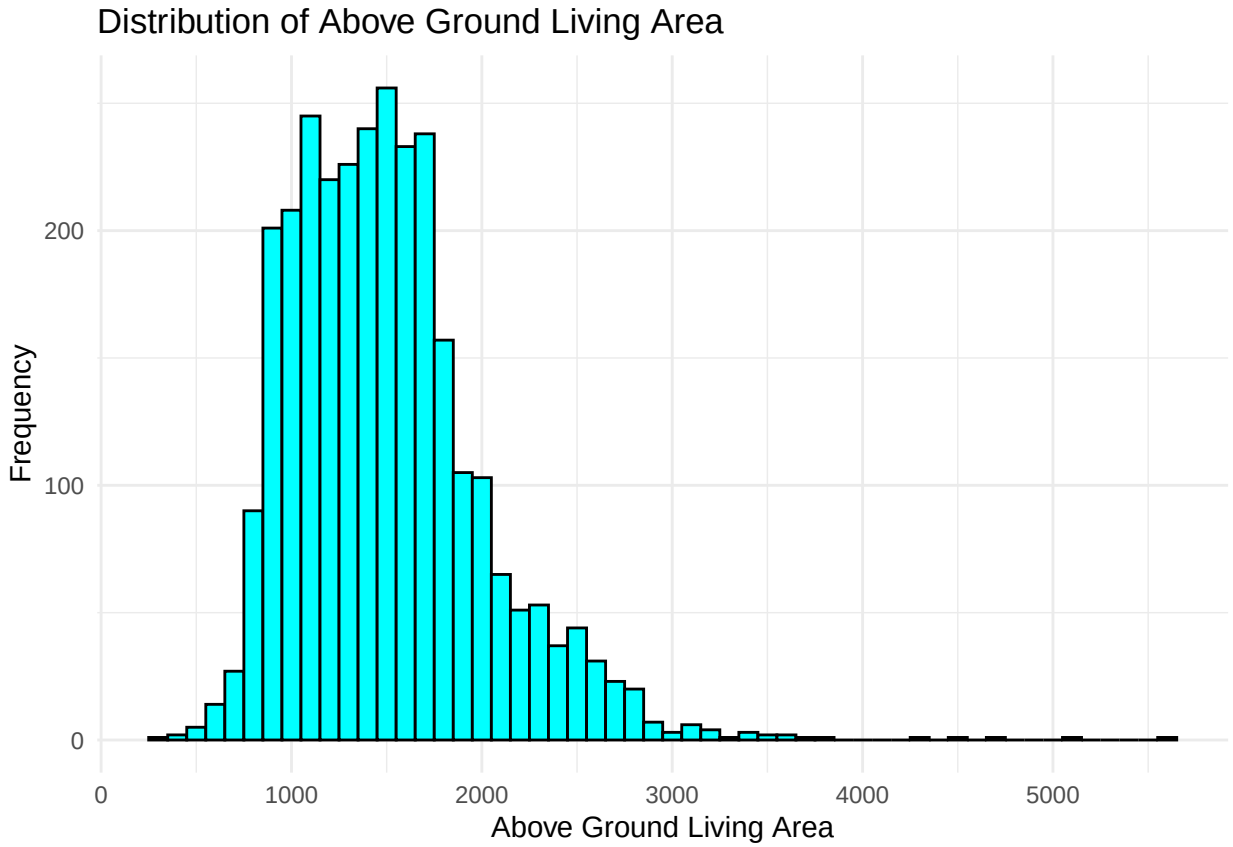
The histogram for Sale Price shows the frequency of houses sold within different price ranges.Each bar represents the number of houses sold within a $10,000 range.This plot helps identify the most common price ranges for houses in the dataset.

The histogram for Gr Liv Area shows the frequency of houses with different above ground living areas.Each bar represents the number of houses with living areas within a 100 square feet range.

```
# Plotting distributions of some key variables

ggplot(ames, aes(x = sale_price)) + geom_histogram(binwidth = 10000, fill = "brown", color = "black") +
```




```
ggplot(ames, aes(x = gr_liv_area)) + geom_histogram(binwidth = 100, fill = "cyan", color = "black") + t
```



##Imputing missing values with mean

Missing values in the dataset were handled by replacing them with the mean of their respective columns.After executing this code, any missing values (NA) in the ames dataset will be replaced with the mean value of the corresponding column. This ensures that the dataset has no missing values, which can be important for subsequent data analysis and modeling.

```
#Imputing the missing values with the mean
ames <- ames %>%
  mutate(across(everything(), ~ ifelse(is.na(.), mean(., na.rm = TRUE), .)))
```

```
## Warning: There were 16 warnings in `mutate()`.
## The first warning was:
## i In argument: `across(...)`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 15 remaining warnings.
```

##Calculating the correlation matrix

The correlation matrix reveals important relationships between various features and sale prices in the Ames Housing dataset. Key factors such as overall quality, living area, garage size, and basement area have strong positive correlations with sale prices, indicating their significant impact on property values.

```
# Calculating the correlation matrix
cor_matrix <- cor(ames %>% select_if(is.numeric), use = "pairwise.complete.obs")
```

Some key points and interpretations include:

Strong Positive Correlations with Sale Price Overall Quality (0.799): Higher overall quality ratings are strongly associated with higher sale prices.This suggests that better construction and finishing quality sig-

nificantly increase property value.

Moderate Positive Correlations with Sale Price Garage Cars (0.648): The number of cars that can be accommodated in the garage is positively correlated with sale price.Homes with larger garages tend to have higher sale prices.

Weak Positive Correlations with Sale Price Lot Frontage (0.341):The linear feet of street connected to the property has a weak positive correlation with sale price.Larger lot frontages slightly increase property values.

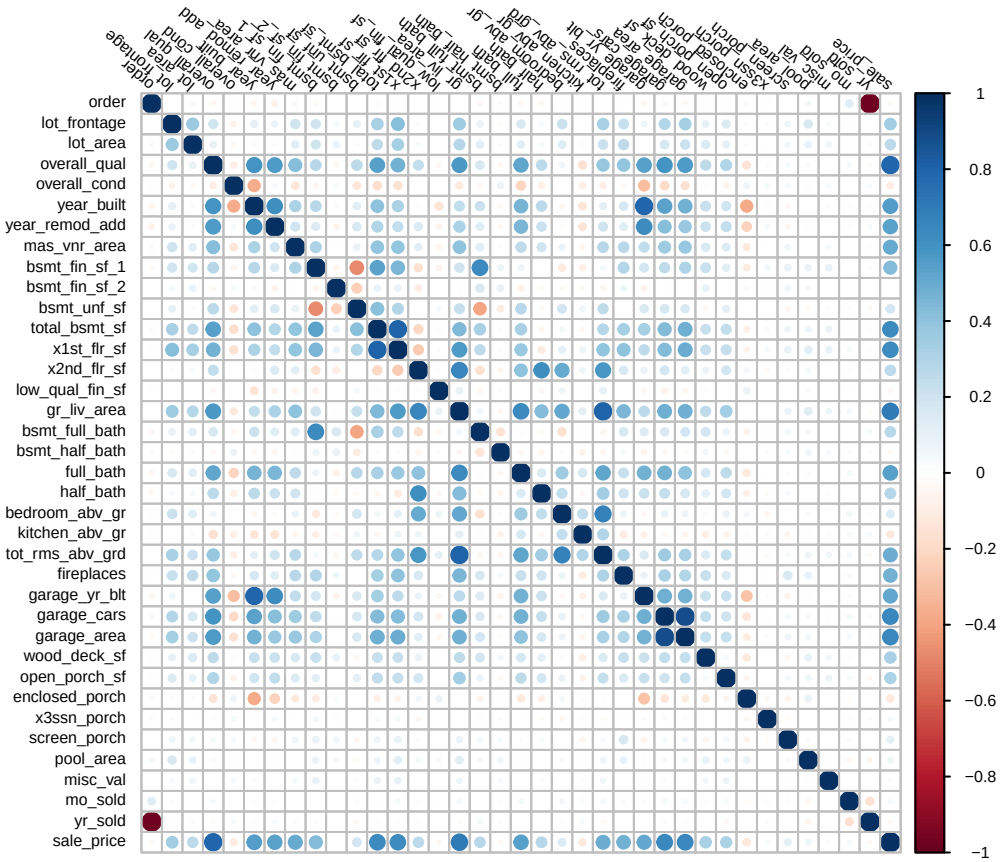
Negative Correlations with Sale Price Overall Condition (-0.102):The overall condition rating has a weak negative correlation with sale price.This suggests that homes with higher condition ratings may not necessarily have higher sale prices, possibly due to other overriding factors.

```
# Plot the correlation matrix
corrplot(cor_matrix, method = "circle", type = "full", tl.col = "black", tl.srt = 500,tl.cex = 0.5,cl.c
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt =
## tl.srt, : "use" is not a graphical parameter
```

```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col =
## tl.col, : "use" is not a graphical parameter
```

```
## Warning in title(title, ...): "use" is not a graphical parameter
```



##Interpretation of Results The provided code performs the following analysis: Finding the variable with the highest correlation with Sale Price:The code sorts the correlation values between all variables and Sale Price in descending order.The variable name associated with the second highest correlation value (after Sale Price itself) is stored in the highest_corr_var variable.

Finding the variable with the lowest correlation with Sale Price:The code sorts the correlation values between all variables and Sale Price in ascending order.The variable name associated with the lowest correlation value

is stored in the lowest_corr_var variable.

Finding the variable with correlation closest to 0.5 with SalePrice: The code calculates the absolute difference between each variable's correlation and 0.5.It then sorts these absolute differences in ascending order and selects the variable name associated with the smallest difference.

Scatter Plot Interpretation:

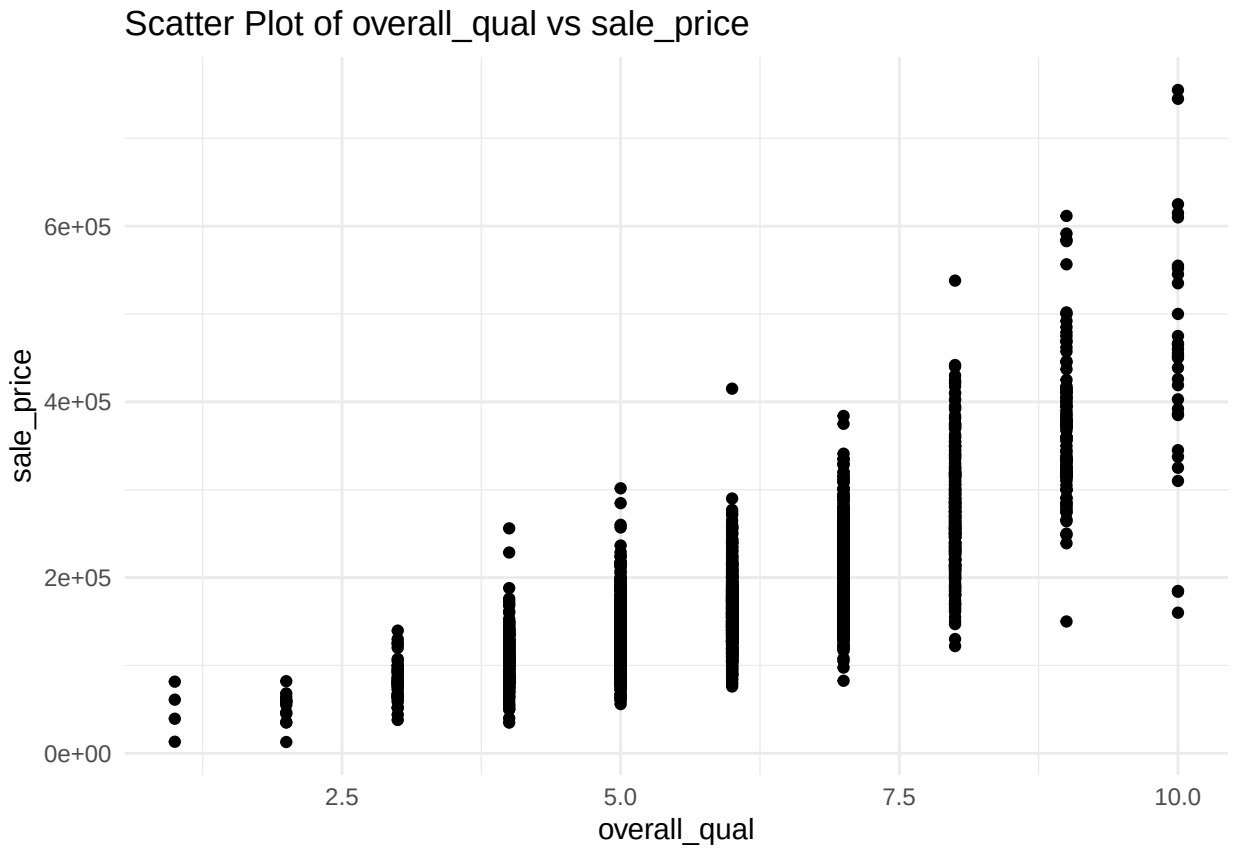
Scatter Plot for the Variable with the Highest Correlation: This plot shows a strong upward trend, indicating that as the value of this variable increases, SalePrice also tends to increase significantly.The points cluster closely around a line, suggesting a linear relationship.A high correlation implying that the variable is a good predictor of Sale Price, capturing a substantial amount of variance in home prices.

Scatter Plot for the Variable with the Lowest Correlation: This plot displays a scattered pattern with no discernible trend, indicating a weak or no relationship between this variable and Sale Price.The points are spread across the plot without forming any clear line or pattern.A low correlation suggests that changes in this variable do not significantly affect Sale Price, making it less relevant for predicting home prices.

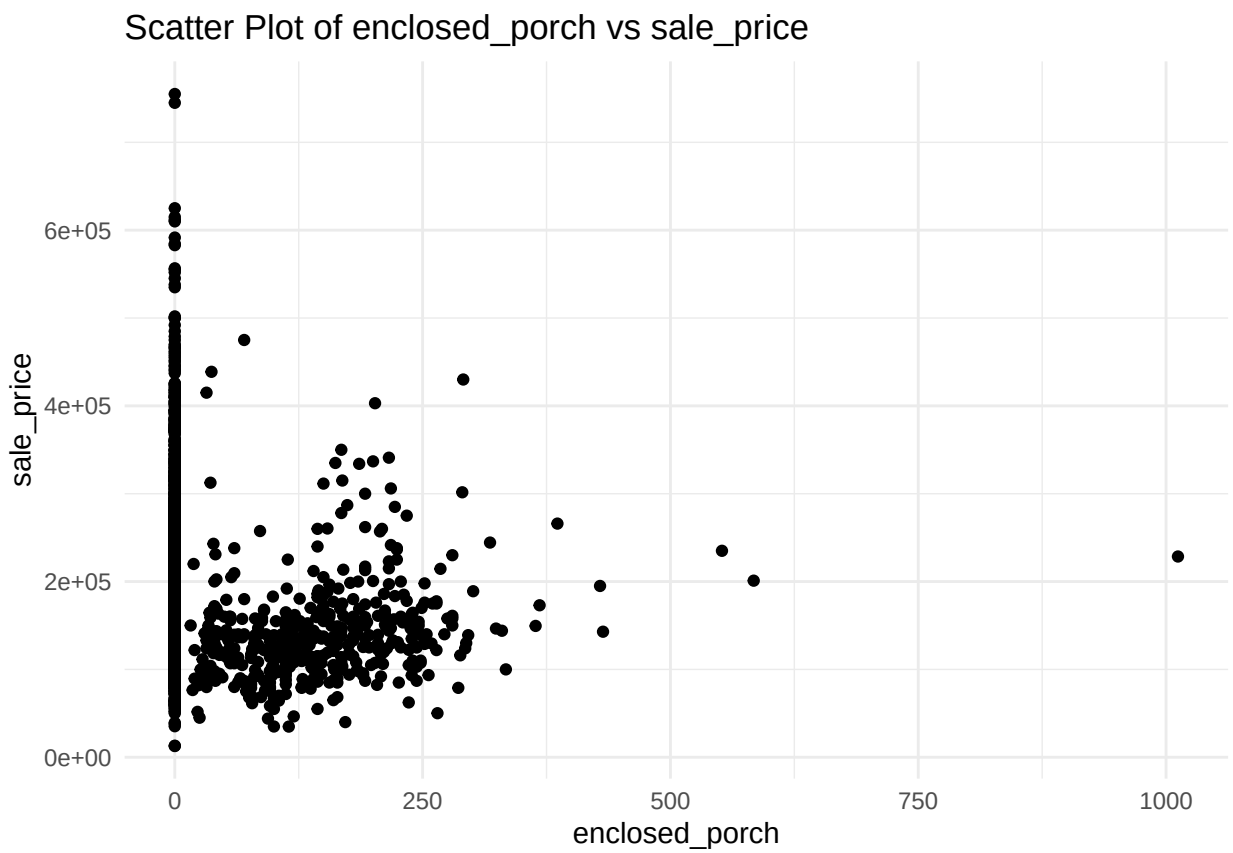
Scatter Plot for the Variable with Correlation Closest to 0.5: This plot reveals a moderate upward trend, suggesting that there is some positive relationship between this variable and Sale Price.The points show more variability compared to the first plot, indicating that while the variable influences Sale Price, it does not do so as strongly or consistently.A correlation close to 0.5 suggests that while there is a relationship, it is not as strong as with the highest correlated variable, indicating that other factors may also play a significant role in determining Sale Price.

```
# Finding the variable with the highest correlation with SalePrice
highest_corr_var <- names(sort(cor_matrix[,"sale_price"], decreasing = TRUE))[2]

# Finding the variable with the lowest correlation with SalePrice
lowest_corr_var <- names(sort(cor_matrix[,"sale_price"], decreasing = FALSE))[1]

# Finding the variable with correlation closest to 0.5 with SalePrice
mid_corr_var <- names(sort(abs(cor_matrix[,"sale_price"] - 0.5)))[1]

# Scatter plots
ggplot(ames, aes_string(x = highest_corr_var, y = "sale_price")) + geom_point() + theme_minimal() + labs
```

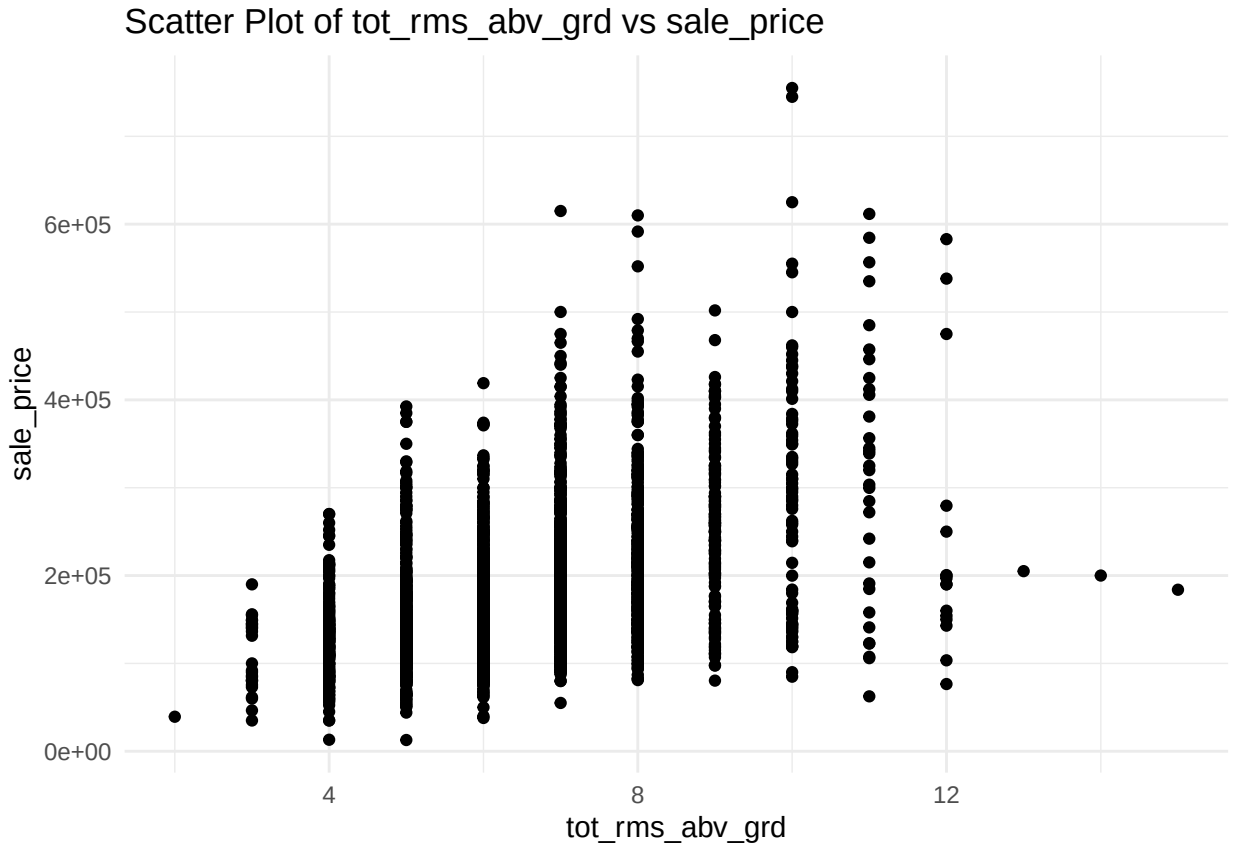
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
ggplot(ames, aes_string(x = lowest_corr_var, y = "sale_price")) + geom_point() + theme_minimal() + labs
```




```
ggplot(ames, aes_string(x = mid_corr_var, y = "sale_price")) + geom_point() + theme_minimal() + labs(ti
```



##Coefficients

(Intercept): Estimate: -29593.6437 Std. Error: 10544.4550 t value: -2.8066 Pr(>|t|): 0.00504 (**) Interpretation: The intercept represents the expected value of the dependent variable (SalePrice) when all predictors are zero. In this case, the intercept is -29593.6437, which is not meaningful in a real-world context since a negative sale price is not possible. The p-value (0.00504) indicates that the intercept is statistically significant at the 0.01 level. gr_liv_area (Above Ground Living Area):

Estimate: 68.8623 Std. Error: 4.4087 t value: 15.6196 Pr(>|t|): < 2.2e-16 (***) Interpretation: For every additional square foot of above ground living area, the sale price is expected to increase by 68.8623 units, holding other variables constant. The very low p-value (< 2.2e-16) indicates that this coefficient is highly statistically significant. total_bsmt_sf (Total Basement Area):

Estimate: 54.5858 Std. Error: 7.7093 t value: 7.0806 Pr(>|t|): 1.79e-12 (***) Interpretation: For every additional square foot of total basement area, the sale price is expected to increase by 54.5858 units, holding other variables constant. The very low p-value (1.79e-12) indicates that this coefficient is highly statistically significant. garage_area (Garage Area):

Estimate: 105.1446 Std. Error: 7.5506 t value: 13.9253 Pr(>|t|): < 2.2e-16 (***) Interpretation: For every additional square foot of garage area, the sale price is expected to increase by 105.1446 units, holding other variables constant. The very low p-value (< 2.2e-16) indicates that this coefficient is highly statistically significant. Significance Codes Signif. codes: 0 ’ ’ 0.001 ’ ’ 0.01 ’ ’ 0.05 ‘.’ 0.1 ‘ ’ 1 These codes indicate the significance levels of the p-values: ‘***’ ***indicates p < 0.001 (highly significant)*** ‘**’ *indicates p < 0.01 (very significant)* ‘*’ indicates p < 0.05 (significant) ‘.’ indicates p < 0.1 (marginally significant) ‘ ’ indicates p 0.1 (not significant) Summary Intercept: The intercept is statistically significant but not meaningful in a real-world context. gr_liv_area: The above ground living area has a strong positive and highly significant impact on sale price. total_bsmt_sf: The total basement area also has a strong positive and highly significant impact on sale price. garage_area: The garage area has the strongest positive and highly significant impact on sale price among the predictors. These results indicate that increasing the above ground living area, total

basement area, and garage area are all associated with higher sale prices, with garage area having the largest impact. The statistical significance of these coefficients suggests that they are reliable predictors of sale price in the Ames Housing dataset.

```
# Fitting the regression model
# Predicting sale price based on ground above living area, garage area and total square feet of basemen

reg_model <- lm(sale_price ~ gr_liv_area + total_bsmt_sf + garage_area, data = ames)


summary(reg_model)$adj.r.squared
```

```
## [1] 0.6791461
```

```
AIC(reg_model)
```

```
## [1] 71140.06
```

```
BIC(reg_model)
```

```
## [1] 71169.97
```

```
stargazer(reg_model, type = "text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                              sale_price
## ------------------------------------------------
## gr_liv_area                   68.862***
##                                (1.966)
##
## total_bsmt_sf                 54.586***
##                                (2.257)
##
## garage_area                  105.145***
##                                (4.736)
##
## Constant                   -29,593.640***
##                              (2,830.734)
##
## ------------------------------------------------
## Observations                    2,930
## R2                              0.679
## Adjusted R2                     0.679
## Residual Std. Error   45,251.000 (df = 2926)
## F Statistic         2,067.588*** (df = 3; 2926)
## ================================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
# robust standard errors

coeftest (reg_model, vcov = vcovHC (reg_model, type ="HC3"))
```

```
##
## t test of coefficients:
##
```

```
##                  Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)   -29593.6437  10544.4550 -2.8066   0.00504 **
## gr_liv_area       68.8623      4.4087 15.6196 < 2.2e-16 ***
## total_bsmt_sf     54.5858      7.7093  7.0806  1.79e-12 ***
## garage_area      105.1446      7.5506 13.9253 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#Model Report (beta_0): Intercept, the expected SalePrice when all predictors are zero. (beta_1): Coefficient for Gr_Liv_Area, the expected change in SalePrice for a one-unit increase in Gr_Liv_Area. (beta_2): Coefficient for Garage_Area, the expected change in SalePrice for a one-unit increase in Garage_Area. (beta_3): Coefficient for Total_Bsmt_SF, the expected change in SalePrice for a one-unit increase in Total_Bsmt_SF.

```
cat("Regression Equation: SalePrice =", round(coef(reg_model)[1], 2), "+", round(coef(reg_model)[2], 2)
```

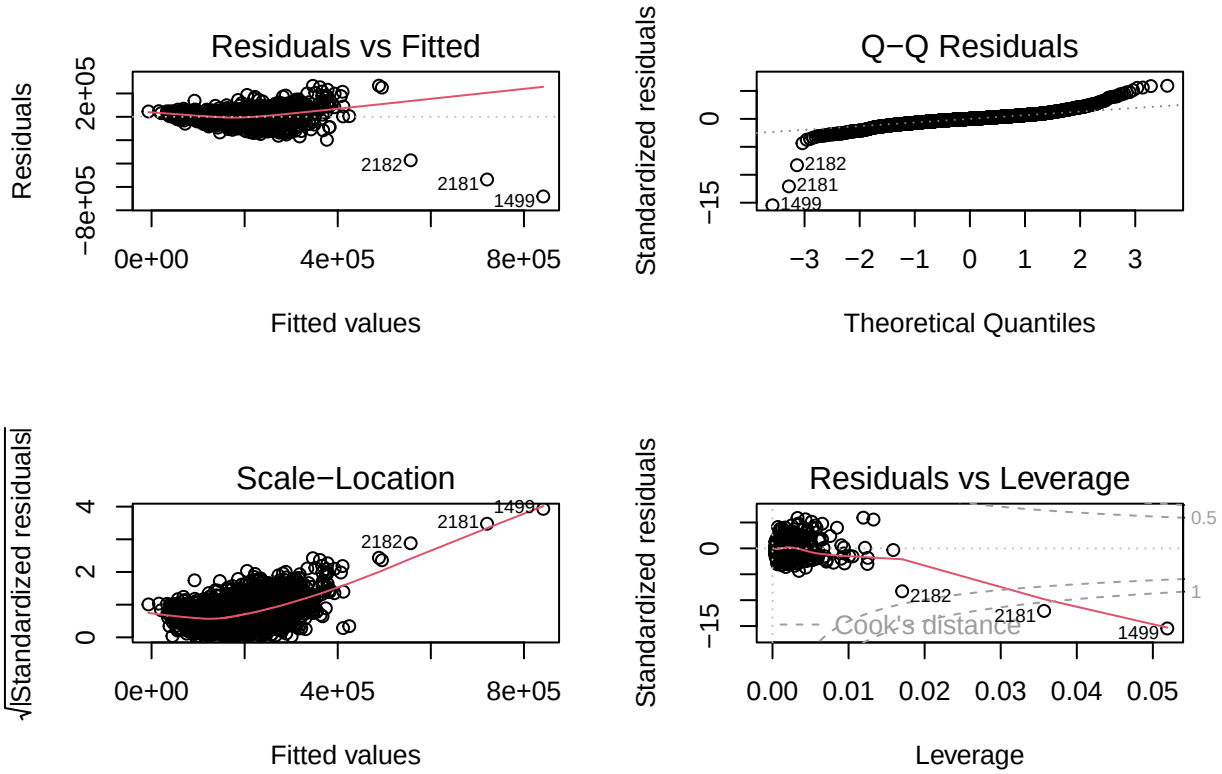
```
## Regression Equation: SalePrice = -29593.64 + 68.86 54.59 * GarageArea + 105.14 * TotalBsmtSF
```

```
## Regression Equation: SalePrice = -29536.07 + 68.86 * GrLivArea + 105.09 * GarageArea + 54.59 This eq
```

##Plotting the regression models

Residuals vs. Fitted Values:This plot shows the residuals (differences between observed and predicted values) plotted against the fitted values (predicted values). A random scatter around zero suggests that the model adequately captures the underlying relationship. Patterns or systematic shapes indicating potential issues with the model, such as non-linearity. Normal Q-Q Plot:This plot compares the distribution of the residuals to a normal distribution. If the points closely follow the diagonal line, it suggests that the residuals are normally distributed, which is an assumption of linear regression. Deviations from the line indicate non-normality. Scale-Location Plot (Spread-Location Plot):This plot shows the square root of standardized residuals against fitted values.It helps assess homoscedasticity (constant variance of residuals). A horizontal line with equally spread points suggests constant variance, while a trend (e.g., a funnel shape) indicates heteroscedasticity. Residuals vs. Leverage: This plot helps identify influential data points.Points that are far from the center (high leverage) and have large residuals may disproportionately affect the model. A Cook's distance line may also be included to highlight influential observations.

```
# Plotting the regression model
par(mfrow = c(2, 2))
plot(reg_model)
```



##Checking for multicollinearity

I then went ahead to assess various assumptions and potential issues in my regression model. The The vif() function calculates the Variance Inflation Factor (VIF) for each predictor in the regression model. The qqPlot() function creates a Quantile-Quantile (Q-Q) plot to assess the normality of the residuals from the regression model.If the residuals follow a straight line in the Q-Q plot, this indicates that they are approximately normally distributed, which is an assumption of linear regression. The crPlots() function generates Component + Residual plots, which help assess the linearity assumption of the regression model.Each plot shows the relationship between a predictor and the response variable while accounting for the effects of other predictors. The spreadLevelPlot() function assesses the homoscedasticity assumption by examining the spread of the residuals against fitted values.

```
# Loading the car library
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'
```

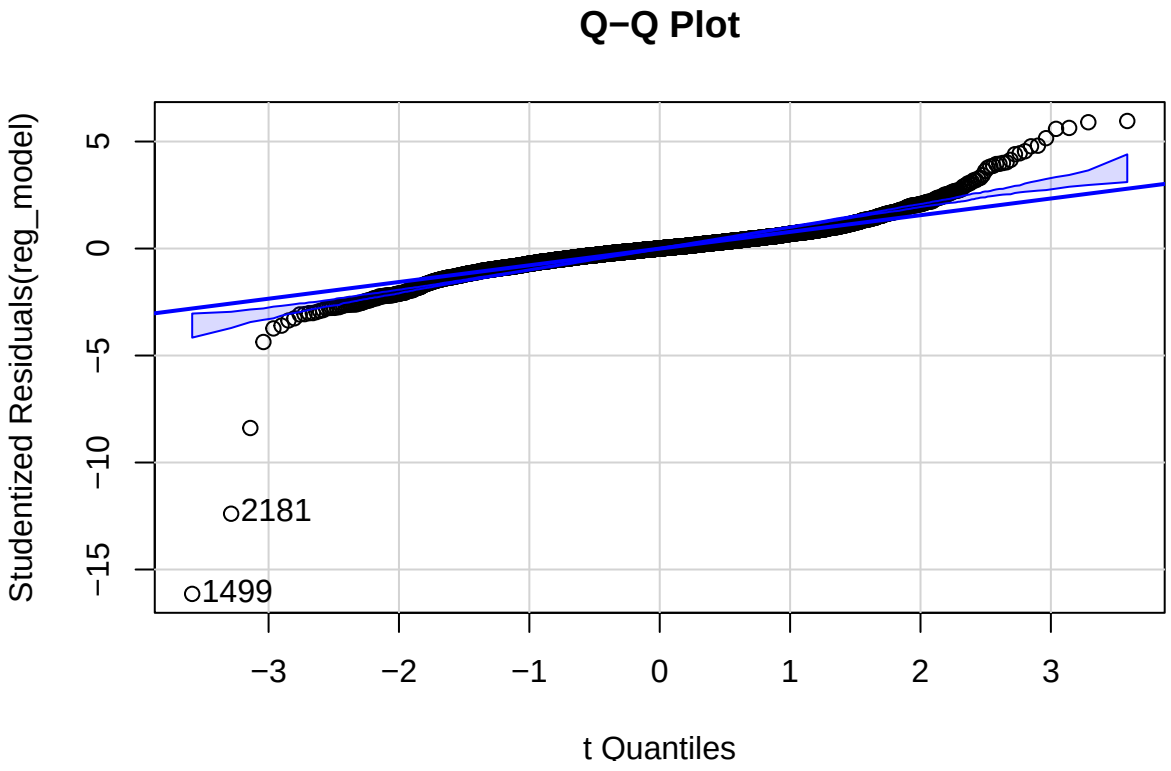
```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```
## The following object is masked from 'package:purrr':
##
##     some
```

```
# Checking for multicollinearity
vif(reg_model)
```

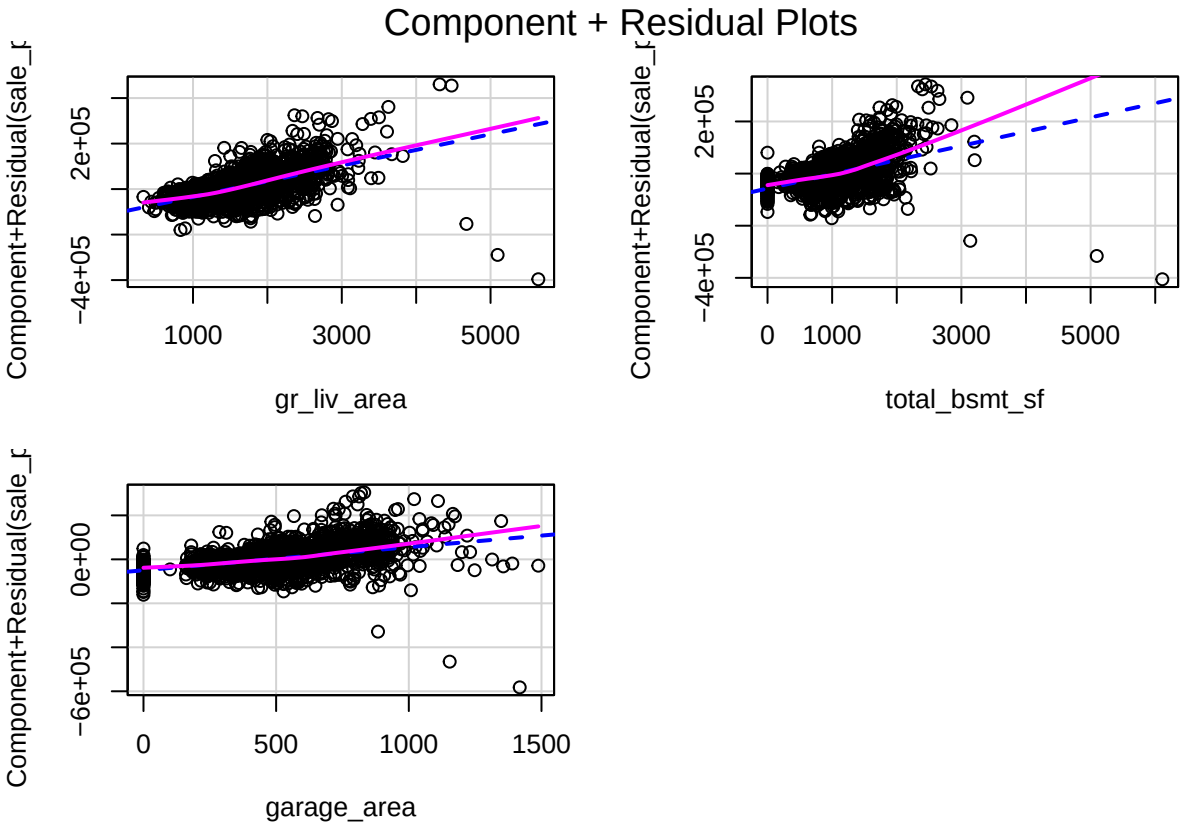
```
##   gr_liv_area total_bsmt_sf   garage_area
##      1.413121      1.414133      1.483363
```

```r
#Q-Q Plot for normality
qqPlot(reg_model,labels=row.names(ames),simulate = TRUE, main = 'Q-Q Plot')
```



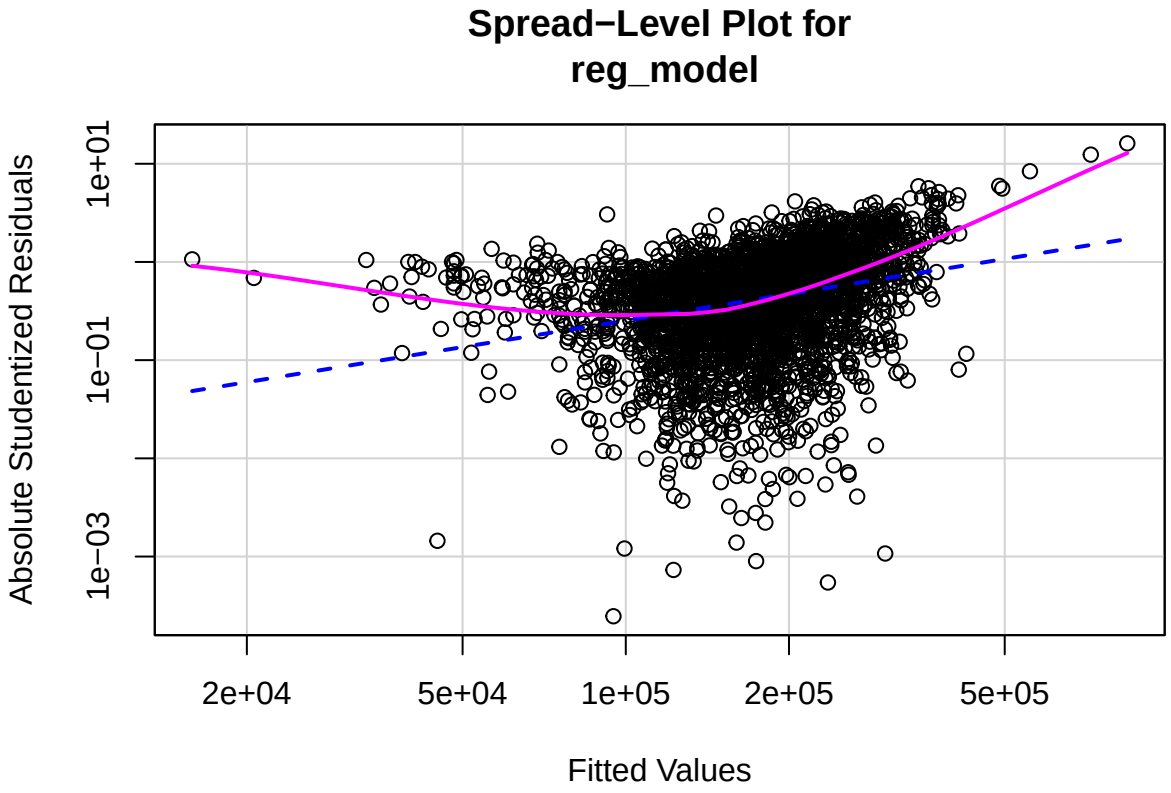

```
## [1] 1499 2181
```

```r
#Component + Residuals - Linearity
crPlots(model = reg_model)
```




```
#Spread-Level Plot for fit - Homoscedasticity
spreadLevelPlot(reg_model)
```

```
## Warning in spreadLevelPlot.lm(reg_model):
## 1 negative fitted value removed
```




```
##
## Suggested power transformation:  0.1026493
```

##Checking for Outliers in my Regression Model

I then went on to identify any potential outliers in my regression model. Observations with studentized residuals greater than 3 are considered outliers. Decisions on removing them depend on their impact on the model.

```
#Testing for outliers
outliers <- which(abs(rstudent(reg_model)) > 3)

# Check the identified outliers
print(outliers)
```

```
##   16   45  424  432  433  434  457  605 1064 1183 1259 1499 1560 1638 1641 1643
##   16   45  424  432  433  434  457  605 1064 1183 1259 1499 1560 1638 1641 1643
## 1694 1761 1768 1783 1994 2181 2182 2246 2257 2330 2331 2333 2335 2342 2383 2446
## 1694 1761 1768 1783 1994 2181 2182 2246 2257 2330 2331 2333 2335 2342 2383 2446
## 2451 2562 2593 2603 2659 2884 2893
## 2451 2562 2593 2603 2659 2884 2893
```

##Removing Outliers

After identifying all outliers in the dataset and seeing how they impacted the regression model, I went head to remove all outliers and rebuilt a new linear regression model using the cleaned dataset.By using the cleaned data, the model aims to provide a more accurate representation of the relationships between the predictors and the sale price, free from the influence of outliers that may have distorted the previous analysis. These results indicate that increasing the above ground living area, garage area, and total basement area are all associated with higher sale prices, with garage area having the largest impact. The statistical significance of these coefficients suggests that they are reliable predictors of sale price in the Ames Housing dataset.

```r
# Remove outliers from the dataset
ames_clean <- ames[-outliers, ]


#Regression model 2 with clean data void of all outliers
model_clean <- lm(sale_price ~ gr_liv_area + garage_area + total_bsmt_sf, data = ames_clean)
model_clean
```

```
##
## Call:
## lm(formula = sale_price ~ gr_liv_area + garage_area + total_bsmt_sf,
##     data = ames_clean)
##
## Coefficients:
##   (Intercept)    gr_liv_area    garage_area  total_bsmt_sf
##     -34930.15          69.46         102.03          59.58
```

##Subset selection for regression model

Formula: sale_price ~ gr_liv_area + garage_area + total_bsmt_sf This specifies the dependent variable (sale_price) and the independent variables (gr_liv_area, garage_area, total_bsmt_sf). Forced in/Forced out: Indicates whether any variables were forced into or out of the model. In this case, no variables were forced in or out.

These results indicate that gr_liv_area is the most important predictor of sale_price, followed by total_bsmt_sf and garage_area. The exhaustive search algorithm ensures that the best subset of predictors is selected for each subset size.Best 3-Predictor Model: gr_liv_area, garage_area, and total_bsmt_sf These results indicate that gr_liv_area is the most important predictor of sale_price, followed by total_bsmt_sf and garage_area. The exhaustive search algorithm ensures that the best subset of predictors is selected for each subset size.

```r
# Performing all subset regressions
best_model <- regsubsets(sale_price ~ gr_liv_area + garage_area + total_bsmt_sf,data =ames_clean ,nbest

# Summary of the best model
summary(best_model)
```

```
## Subset selection object
## Call: regsubsets.formula(sale_price ~ gr_liv_area + garage_area + total_bsmt_sf,
##     data = ames_clean, nbest = 1, nvmax = 3)
## 3 Variables  (and intercept)
##               Forced in Forced out
## gr_liv_area       FALSE      FALSE
## garage_area       FALSE      FALSE
## total_bsmt_sf     FALSE      FALSE
## 1 subsets of each size up to 3
## Selection Algorithm: exhaustive
##          gr_liv_area garage_area total_bsmt_sf
## 1  ( 1 ) "*"         " "         " "
## 2  ( 1 ) "*"         " "         "*"
## 3  ( 1 ) "*"         "*"         "*"
```

##Models Comparison Comparing the two models built in this analysis, the Model from step 12 seemed simpler and had fewer variables but the Preferred Modelcould potentially be more accurate since it includes more variables.

```
##References:

#Housing Dataset. (n.d.).

#James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An Introduction to Statistical Learning:
#With Applications in R. New York: Springer.

#Kabacoff, R. I. (2015). R in Action: Data Analysis and Graphics with R (2nd ed.). Shelter Island, NY:
#Manning Publications.

#Kuhn, M., & Johnson, K. (2013). Applied Predictive Modeling. New York: Springer.

#Wickham, H., & Grolemund, G. (2017). R for Data Science: Import, Tidy,
#Transform, Visualize, and Model Data. Sebastopol, CA: O'Reilly Media.

#Fox, J., & Weisberg, S. (2019). An R Companion to Applied Regression (3rd ed.).
#Thousand Oaks, CA: Sage Publications.

#Harrell, F. E. (2015). Regression Modeling Strategies: With Applications to Linear Models,
#Logistic and Ordinal Regression, and Survival Analysis (2nd ed.). New York: Springer.
```