Stat 6021: Project 2

Group 4: Camisha Belle; Brian Nolton; Amanda Betag; Heejeong Yoon

2024-12-15

Section 1: High-Level Results of Analysis

Group four (4) of the Masters of Data Science program at the University of Virginia was commissioned to study house sale prices for King County, Washington for homes sold between May 2014 and May 2015. King County includes Seattle, a diverse metropolitan with the largest population in the Pacific Northwest region of the United States.

Linear Regression - Results and Practicality

To investigate the relationship between house price and multiple house characteristics, group 4 used multiple linear regression. This method helps us figure out how varied factors, like the number of bedrooms or a waterfront view, affect one particularly important element. For this study, the essential element is house price. Using linear regression, we can understand which properties are most important as well as predict a house's price based on those features.

The final linear regression model included significant predictors such as bathrooms, waterfront presence, and region. Key predictors such as square feet of living space, waterfront, renovated, condition, and grade show strong influence on home property prices.

For instance, each additional bathroom is expected to increase the price by approximately 1.28%, assuming all other factors remain constant. While 1.28% may not seem like a lot, it equates to an increase of more than $6{,}900 in the average home price()$. Regional effects are also significant, with the South region associated with a substantial price decrease compared to the North. This model serves as a robust tool for estimating house prices within King County and offers insights for real estate stakeholders (i.e. agents, analysts, home buyers).

Logistic Regression - Results and Practicality

House price is not the only important characteristic of a home. One famous Seneca quote reads, "It is quality rather than quantity that matters." We agree, and explored whether a home was of good quality and what characteristics influence that classification by 1) creating a variable, good_quality, indicating whether a house has better-than-average condition and grade as well as 2) employing logistic regression using that newly created variable.

The logistic model identified significant predictors, including price, number of bedrooms, and whether the house was renovated. These variables provide critical insights into what contributes to a home's perceived quality.

Group 4's logistic regression model was designed to predict the probability of a property being of good quality based on several characteristics. Ultimately, the log of the house price and age are the most significant predictors. This suggests that higher price and property age are strongly associated with higher odds of a home being classified as good quality. Overall, the model highlights that the log of price, age, and region are important factors influencing the likelihood of a property being classified as of good quality, while the number of bedrooms is moderately influential, and bathrooms have little effect.

Specifically, for every unit increase in the log of price, the odds of a property being good quality are approximately 6.58 times higher. A one-unit increase in the log of price corresponds to a 558% increase in the odds of a property being good quality. Being renovated reduces the odds of a property being good quality to 17% of the odds for non-renovated properties (or an 83% decrease in odds). This model can be useful for a myriad of interested parties including home buyers and building contractors.

Section 2: Data Exploration

Description of Variables

This dataset contains the house prices from King County, Washington (Seattle and surrounding area) from May 2014 to May 2015. The dataset contains 21 variables total (including house price). Below you will find a description of each of the variables.

id: a unique ID per house sold
date: the date of the house sale
price: how much the house was sold for
bedrooms: the number of bedrooms in the house
bathrooms: the number of bathrooms in the house
sqft_living: the square footage of living space in the house
sqft_lot: the square footage of the lot the house is on
floors: the number of floors in the house
waterfront: an index as to whether or not the house is overlooking the waterfront, where 0 represents no waterfront and 1 represents a waterfront overlook
view: an index from 0 to 4 of how good the view of the property is, where 0 represents not a good view and 4 represents an excellent view
condition: an index from 0 to 5 on the condition of the house, where 0 represents a poor condition and 5 represents excellent condition
grade: an index from 1 to 13 on the grade of the house where 1 to 3 falls short of building construction and design, 7 has an average level of construction and design, and 11 to 13 have higher quality level of construction and design
sqft_above: the square footage of the house above ground level
sqft_below: the square footage of the house below ground level
yr_built: the year the house was built
yr_renovated: the year of the house's last renovation
zipcode: the postal zipcode of the house
lat: the house's latitude
long: the house's longitude
sqft_living15: the average of the square footage of the living space for the nearest 15 neighboring houses
sqft_lot15: the average square footage of the lot size for the nearest 15 neighboring houses

We also created new variables to aid us. Here are the new variables with the description of each one.

yr_sold: the year the house was sold. This taken from the date variable
quarter: indicates the quarter of the year the house was sold.
age: represents how old the house was when it was sold. It is calculated by the year the house was sold minus the year the house was built.
renovated: indicates whether or not the house was renovated, where 0 represents no renovation documented and 1 represents a documented renovation. The yr_renovated variable is helpful, but full of 0s. These 0s mean the house either wasn't renovated or the renovation wasn't recorded (for whatever reason). Using this variable as is with all the 0s would not be a good metric, but knowing whether a house was renovated could have an impact on the house price. It was then factorized and the levels changed to no for 0 and yes for 1.
region: represents the region the house is in based on zip codes. We grouped 70 unique zip codes into 4 regions (Seattle, North, East, South) based on 3 references. 1. Zip code map of King county , Washington- https://your.kingcounty.gov/GIS/web/Web/VMC/boundaries/zipcodes.pdf 2. King county geographics - https://www.communitiescount.org/king-county-geographies 3. Zip code list - https://www.ciclt.net/sn/clt/capitolimpact/gw_ziplist.aspx?FIPS=53033
living_gt: indicates whether the house's living area is greater than or less than the the nearest 15 neighbors, factorized
lot_gt: indicates whether the house and lot are greater than or less than the the nearest 15 neighbors, factorized

view1: indicates whether a house has a view of 0, where 0 represents a house with a view score of 0 and 1 represents a house with a view score greater than 0 (has some view)
condition1: categorizes the condition into "below average" (condition scores of 1 and 2), "average" (condition score of 3), and "above average" (condition scores of 4 and 5)
grade1: categorizes the grade of the house into "not built" (grade scores 1 through 3), "below average" (grade scores 4 through 6), "average" (grade score 7), "above average" (grade scores 8-10), and "excellent" (grade scores 11-13).
good_quality: is defined by a house that has a condition greater than 3 and a grade greater than 7. A house that satisfied that condition were given a value of a 1 and all others (condition less than or equal to 3 or grade less than or equal to 7) were given a 0. It was then factorized and the levels changed to no for 0 and yes for 1.

Section 3: Data Clean Up and Manipulation

First we checked for any missing values. There are no NAs.

```
sum(is.na(house))
```

Then, we looked at the data to see what we were working with.

```
summary(house)
```

When the data was imported, all IDs starting with one or more zeros had the leading zeros removed. We fixed this by adding back the leading zeros.

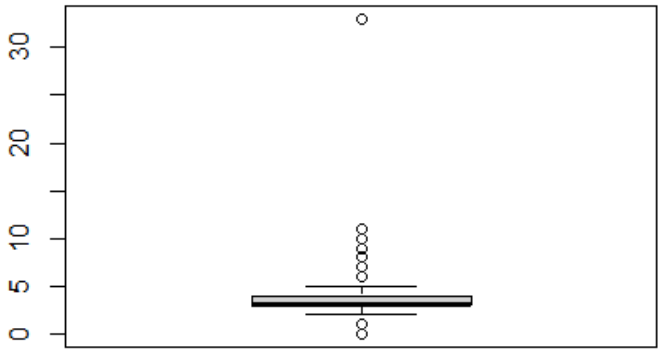
```
house$id <- as.character(house$id)
house <- house %>%
  mutate(id = ifelse(nchar(id) == 9, paste0("0", id), id))
house <- house %>%
  mutate(id = ifelse(nchar(id) == 8, paste0("00", id), id))
house <- house %>%
  mutate(id = ifelse(nchar(id) == 7, paste0("000", id), id))
```

The *date* variable appeared to include a time stamp that didn't work out. We removed the time stamp, then converted the date variable to a date format.

```
house$date <- gsub("T000000", "", house$date)
house$date <- as.Date(house$date, format = "%Y %m %d")
```

The bedrooms variable appeared off. Upon further investigation, it appeared as though the 33 is a typo and the data suggests that this house fits the profile of a 3 bedroom house.

```
boxplot(house$bedrooms)
```




```
which(house$bedrooms == 33)
house[15871,]
```

We changed the value to 3.

```
house$bedrooms[house$bedrooms==33] = 3
```

We dug into the bedrooms variable.

```
table(house$bedrooms)
which(house$bedrooms == 0)
```

Having a house with 0 bedrooms seemed odd. We understood there might be studio homes. However, after looking at all data points in the rows corresponding to these houses, the values were all over the place lessening our confidence in the data. There were only 13 houses listed with 0 bedrooms. We decided to remove these rows from the data frame.

```
house <- house[house$bedrooms != 0,]
```

Next, we looked at the bathrooms.

```
table(house$bathrooms)
```

It did not sense that a house would have 0 bathrooms. After removing the houses with 0 bedrooms there are only 3 houses left without bathrooms. We understood there are apartment buildings with communal bathrooms, but the data on these 3 houses was all over the place. We decided to remove these from the data frame as well.

```
house <- house[house$bathrooms != 0,]
```

Next we factorized the categorical variables. All variables, except the date variable, are listed as numeric. Before changing associated values, we created the $good\_quality$ variable first, then factorize waterfront, view, condition, and grade.

```
house <- house %>%
  mutate(good_quality = case_when(condition > 3 & grade > 7 ~ 1,
                                  condition <= 3 | grade <= 7 ~ 0))
house$good_quality <- factor(house$good_quality)
levels(house$good_quality) <- c("no", "yes")

house$waterfront <- factor(house$waterfront)
levels(house$waterfront) <- c("no", "yes")
house$view <- factor(house$view)
house$condition <- factor(house$condition)
house$grade <- factor(house$grade)
```

When the data was imported, all IDs starting with one or more zeros had the leading zeros removed. While, this may be a problem in certain contexts, since id will not be a variable kept for you regression models (explanation below) we decided to leave it alone. We also noticed duplicate house IDs, meaning that a house was sold more than once in the time frame (177 of them to be exact). While there was variation in some of the sales prices for these houses, ultimately, we decided not to touch any of them and let that data speak for itself. We a

Next, we created the rest of the new variables. Here is the code for them. The descriptions of each one in order (except good_quality) can be found in section 2.

Here is the code for yr_sold:

```
house$yr_sold <- as.numeric(format(house$date, "%Y"))
```

Here is the code for quarter:

```
house$quarter <- as.yearqtr(house$date)
house$quarter <- factor(house$quarter)
```

Here is the code for year_sold:

```
house$age <- house$yr_sold - house$yr_built
```

Here is the code for renovated:

```
house <- house %>%
  mutate(renovated = case_when(yr_renovated == 0 ~ 0,
                               yr_renovated > 0 ~ 1))
house$renovated <- factor(house$renovated)
levels(house$renovated) <- c("no", "yes")
```

Here is the code for region:

```
region_group <- list("Seattle" = c(98102, 98103, 98105, 98106, 98107, 98108,
                98109, 98112, 98115, 98116, 98117, 98118, 98119, 98122,
                98125, 98126, 98133, 98136, 98144, 98146, 98155,
                98168, 98177, 98178, 98188, 98198, 98199),
  "North" = c(98028, 98011, 98021, 98072),
  "East" = c(98004, 98005, 98006, 98007, 98008, 98014, 98019, 98024, 98027, 98029, 98033, 98034,
98039, 98040, 98045, 98052, 98053, 98065, 98074, 98075, 98077),
  "South" = c(98001, 98002, 98003, 98010, 98022, 98023, 98030, 98031, 98032,
              98038, 98042, 98055, 98056, 98058, 98059,98070,
              98092, 98148, 98166)
)
# Function to map zip codes to regions
group_to_region <- function(zipcode) {
  for (region in names(region_group)) {
    if (zipcode %in% region_group[[region]]) {
      return(region)
    }
  }
}
# Apply the function to assign regions
house$region <- sapply(house$zipcode, group_to_region)
#factor region
house$region<-factor(house$region)
```

Here is the code for living_gt and lot_gt:

```
house$living_gt <- ifelse(house$sqft_living > house$sqft_living15, "yes", "no")
house$lot_gt <- ifelse(house$sqft_lot > house$sqft_lot15, "yes", "no")
house$living_gt <- factor(house$living_gt)
house$lot_gt <- factor(house$lot_gt)
```

Here is the code for view1, condition1, and grade1:

```
house$view1 <- ifelse(house$view == "0", 0, 1)
house$view1 <- factor(house$view1)

#Here is the code for condition1:
house$condition1 <- ifelse(house$condition == "1" | house$condition == "2", "below average",
ifelse(house$condition == "3", "average", "above average"))
house$condition1 <- factor(house$condition1)

house$grade1 <- ifelse(house$grade == "1" | house$grade == "2" | house$grade == "3", "not built",
ifelse(house$grade == "4" | house$grade == "5" | house$grade == "6", "below average",
ifelse(house$grade == "7", "average", ifelse(house$grade == "8" | house$grade == "9" | house$grade
== "10", "above average", "excellent"))))
house$grade1 <- factor(house$grade1)
```

Here is the code for logprice:

```
house <- house %>%
  mutate(logprice = log(price))
```

When looking at the variables, some of them did not make sense to include, so we eliminated them. Below are the variables we eliminated and the rationale: **id** is an arbitrary number so should have no bearing on price. **date** rather than a specific date, we will be using the more generalized variable quarter. **zipcode, lat, long** variables describing location of the house that may not be useful have been replaced with the more encompassing region variable. **view, condition, grade** have been replaced by view1, condition1, and grade1 respectively giving more clarity and less variation to these variables. **sqft_above, sqft_basement** these two variables added together make the variable sqft_living. The total square footage makes more sense to use and we sought to avoid collinearity. **year_built, year_sold** year sold was created solely to establish age, which, while the telling the same story as year_built, is easier to understand. **yr_renovated** has been replaced by the simpler renovated. **sqft_living15, sqft_lot15** are both very specific comparisons to the neighboring house. People are more interested in the house they are looking for than the neighbors, since neighborhoods are offer similar housing. They have been replaced with the more general comparisons living_gt and lot_gt respectively.

```
house <- house[,!names(house) %in% c("id", "date", "yr_built", "yr_renovated", "zipcode",
"sqft_living15", "sqft_lot15", "view", "condition", "grade", "lat", "long", "sqft_above",
"sqft_basement", "year_sold")]
```

Section 4: Data Visualizations - Price

The following code splits the dataset to training and test datasets in a 50:50 ratio. Before splitting, the team completed R version check and set.seed(1)/sample.int(100,5) test as instructed by Dr.Woo.

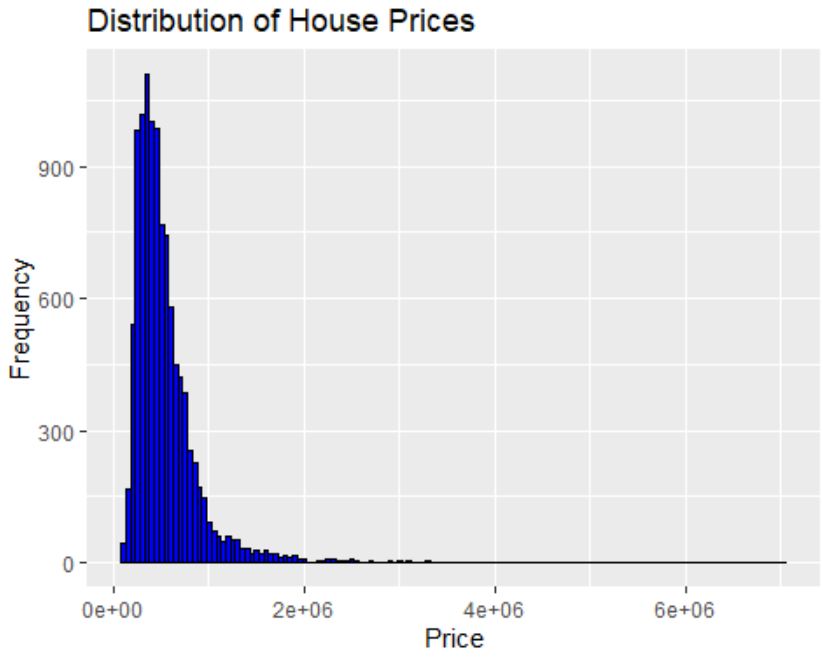
```
set.seed(6021) #initializes the random number generator with 6021
sample.data<-sample.int(nrow(house), floor(.50*nrow(house)), replace = F)
train<-house[sample.data, ]
test<-house[-sample.data, ]
```

The first part of this project was to try and predict the price of a house in King County. To start this process, we will first use visualizations of the variables around the house price.

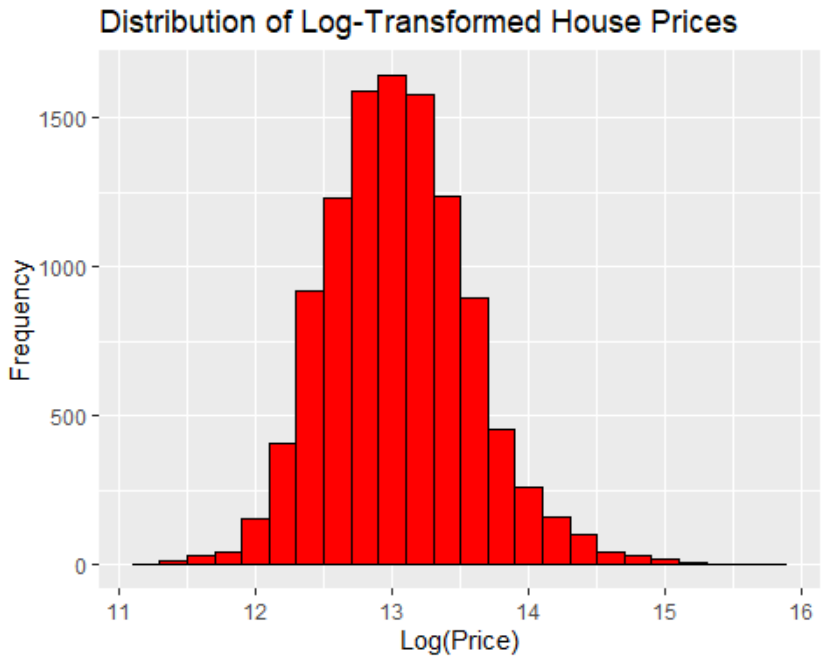
Presenting univariate visualizations

The histogram of house prices confirms the skewed pattern observed in the box plot and density plot. After applying a log transformation, the histogram of the log-transformed prices shows a more balanced distribution. This suggests that transforming the price variable to a log scale could be beneficial for improving the performance and interpretability.

```
ggplot(train, aes(x = price)) +
  geom_histogram(binwidth = 50000, fill = "blue", color = "black") +
  labs(title = "Distribution of House Prices",x = "Price",y = "Frequency")
```




```r
ggplot(train, aes(x = logprice)) +
  geom_histogram(binwidth = 0.2, fill = "red", color = "black") +
  labs(title = "Distribution of Log-Transformed House Prices",x = "Log(Price)",y = "Frequency")
```

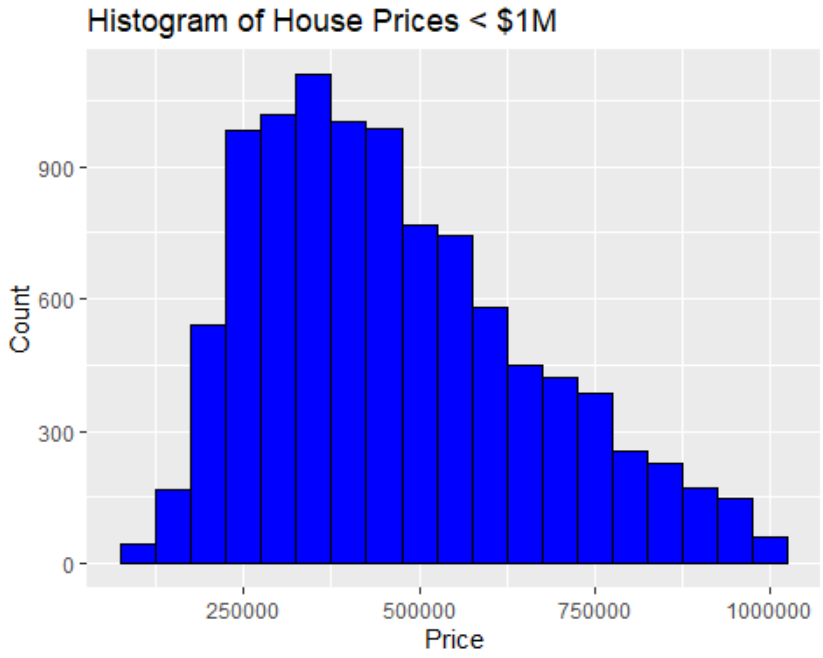


Since the original price distribution is right-skewed, extreme outliers can obscure the underlying pattern in the histogram. To address this, we inspected the range of prices and found a substantial difference between the minimum price of $80K and the maximum price of $7.062M. Additionally, the mean price is $540K, while the median is $450K, further highlighting the influence of outliers on the distribution.

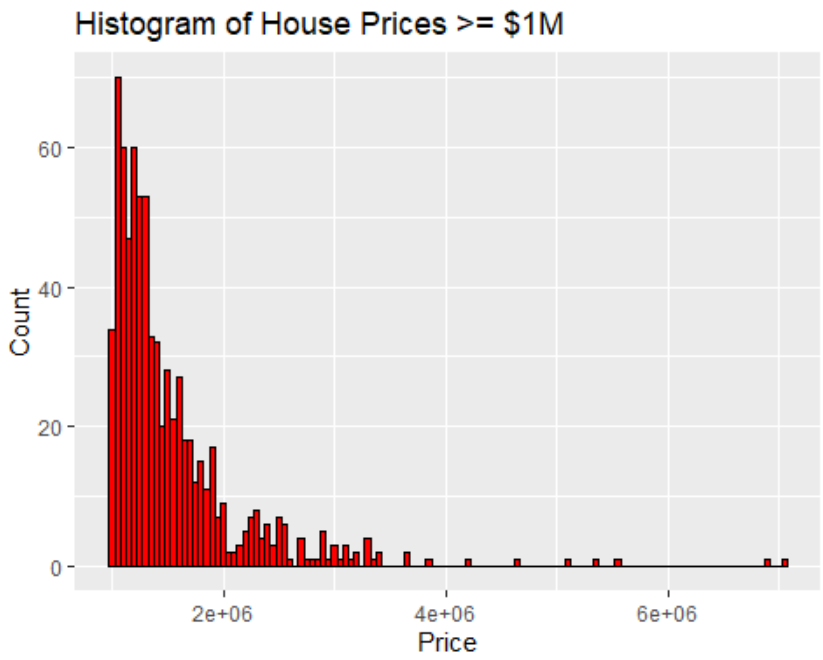
```r
summary(train$price)
```

To investigate further, we created separate histograms for prices below $1M and above $1M. The histogram for prices below $1M shows a fairly balanced distribution around the median, with slight left skewness in the range of $250K to $500K. In contrast, the histogram for prices above $1M is highly skewed to the left, with most prices concentrated below $2M.

```r
ggplot(train %>% filter(price < 1000000), aes(x = price)) +
  geom_histogram(binwidth = 50000, fill = "blue", color = "black") +
  labs(title = "Histogram of House Prices < $1M",x = "Price",y = "Count")
```




```r
ggplot(train %>% filter(price >= 1000000), aes(x = price)) +
  geom_histogram(binwidth = 50000, fill = "red", color = "black") +
  labs(title = "Histogram of House Prices >= $1M",x = "Price",y = "Count")
```

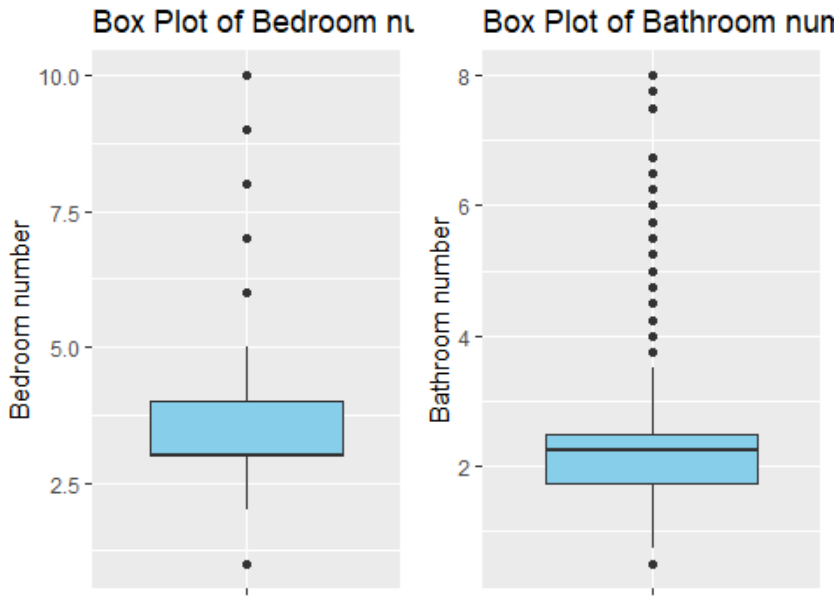


Box-plots of other variables (quantitative variables)

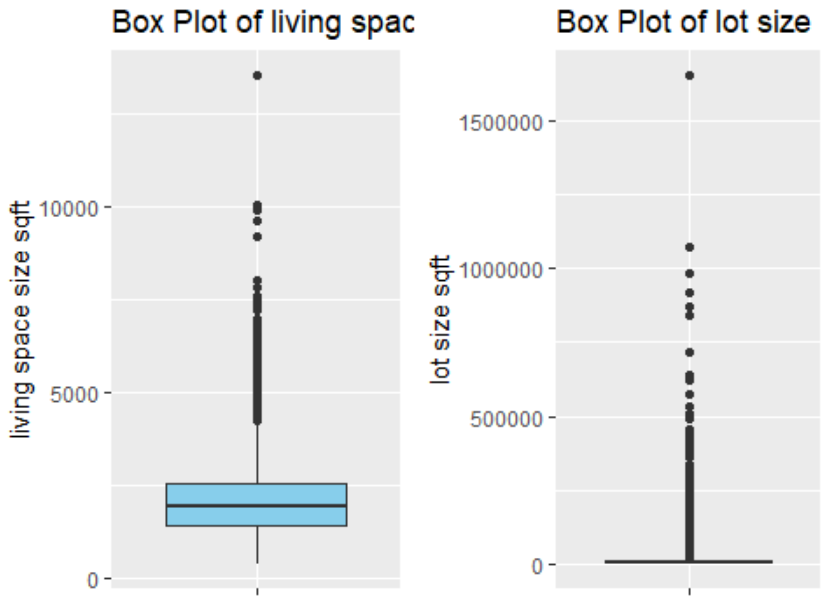
For investigative purposes, we created box plots for several quantitative variables and bar charts for categorical variables. The box plots reveal patterns similar to the price variable, with skewness in a direction that reflects characteristics typically associated with lower-priced houses, such as fewer bedrooms, bathrooms, smaller living spaces, and lot sizes. Higher values for these variables often lie outside the box, likely representing outliers

associated with a small number of high-priced houses. In contrast, house age is more evenly distributed, ranging from 15 to 60 years.

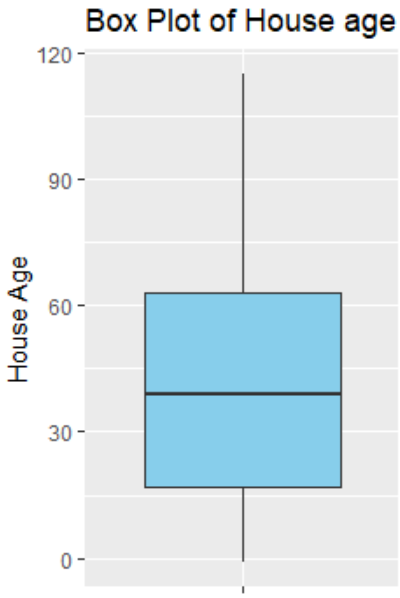
```
grid.arrange(box1, box2, ncol=2, nrow=1)
```




```
grid.arrange(box3, box4, ncol=2, nrow=1)
```



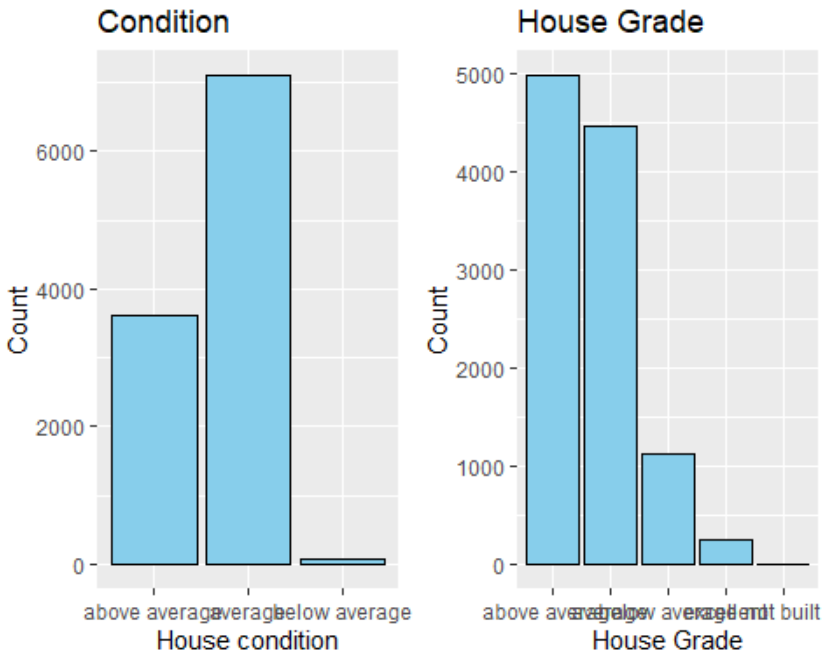

```
grid.arrange(box5, ncol=2, nrow=1)
```



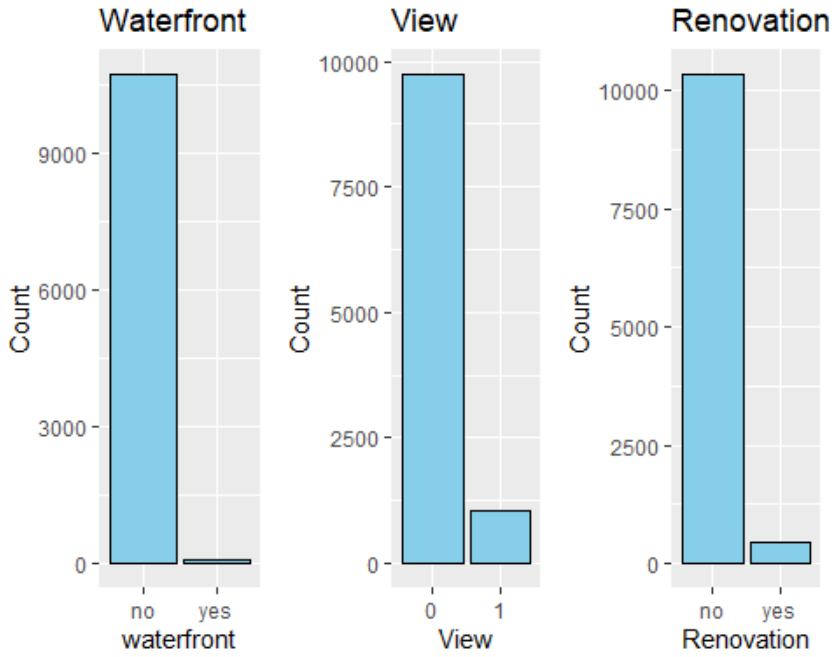
Bar-chart of other variables (categorical variables)

The bar charts for several categorical variables reveal that the majority of house sales occurred in the Seattle area. The East and South regions have a similar number of sales, while the North region shows the lowest number of house sales. The bar charts for house conditions and grade are skewed toward values greater than average.The bar charts for categorical variables that are commonly considered desirable by buyers—such as waterfront, view, and renovation—show that most houses have no waterfront, no view grades, and no renovations. The bar charts for larger spaces (both living space and lot size) show no significant difference in the number of houses with sizes larger than 15 neighbors compared to those with smaller sizes. This suggests that the dataset includes a well-distributed range of house sizes within each neighborhood.

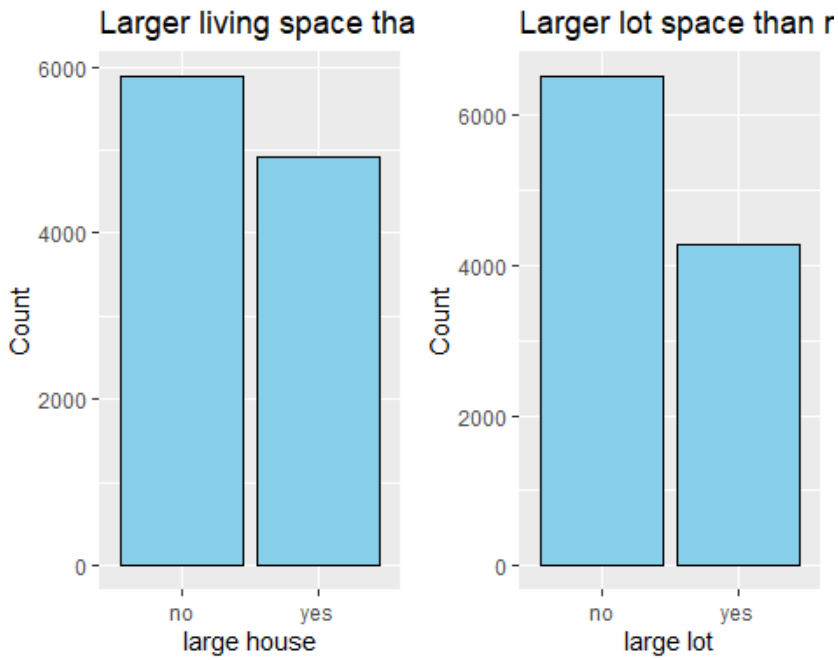
```
grid.arrange(bar4, bar5, ncol=2, nrow=1)
```



```
grid.arrange(bar2, bar3, bar6, ncol=3, nrow=1)
```




```
grid.arrange(bar7, bar8, ncol=2, nrow=1)
```

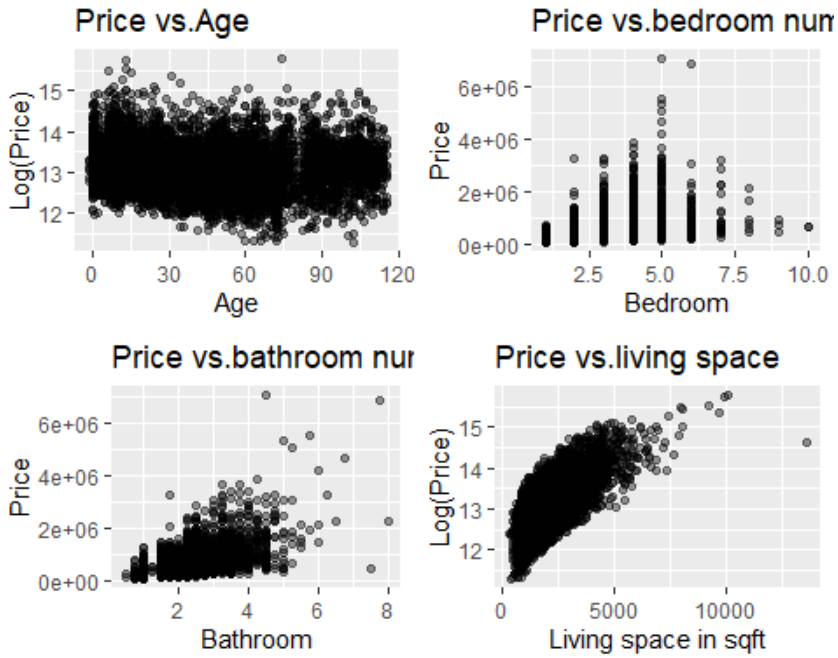


Presenting bivariate visualizations

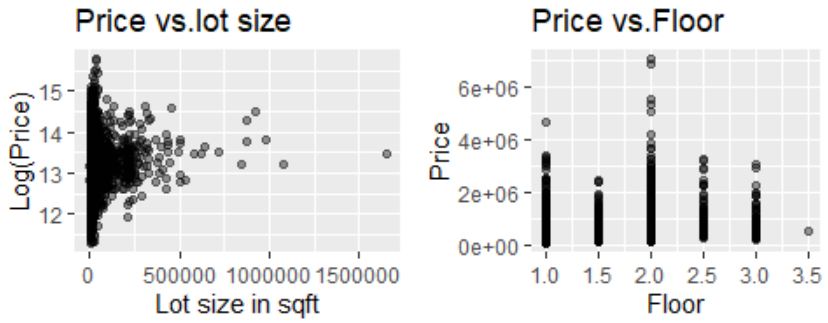
Scatter plots of Price vs. other quantitative variables

Scatter plots were used to explore associations between quantitative variables and house price. Price shows no clear pattern across house age. Positive linear associations are observed for variables like the number of bathrooms, living space size. However, no noticeable association is seen for the number of bedrooms, lot size, or number of floors.

```
grid.arrange(scatter1, scatter2, scatter3, scatter4, ncol=2, nrow=2)
```




```
grid.arrange(scatter5, scatter6, ncol=2, nrow=2)
```



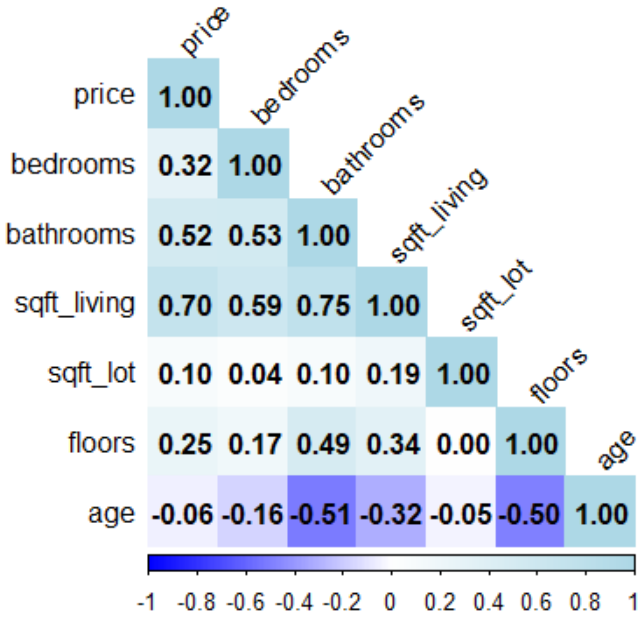
Heatmap of correlation of quantitative variables

For further investigation, we created a heat map to examine the associations between price and other quantitative predictors, as well as the relationships among the predictors themselves.The number of bathrooms and the size of the living space have the strongest positive correlations with price (>0.50), while the number of bedrooms, lot size and the number of floors show weak positive correlations with price. These findings align with the patterns observed in the scatter plots. House age shows a negative association with price, but the correlation is negligible (−0.06).

The heat map also reveals that the number of bedrooms, bathrooms and the size of the living space are highly correlated with one another (0.53−0.75). Also the number of bathrooms and floors have the strong negative correlations with age (<-0.50). This suggests that we should investigate potential multicollinearity among these predictors in a later section.

```
quantitative_predictors <- train[, c("price", "bedrooms", "bathrooms",
"sqft_living","sqft_lot","floors","age")]
# Create a correlation matrix
correlation_matrix <-round(cor(quantitative_predictors, use = "complete.obs"),3)
# Visualize the correlation matrix
corrplot(correlation_matrix, method = "color", type = "lower",
         tl.col = "black", tl.srt = 45,
```

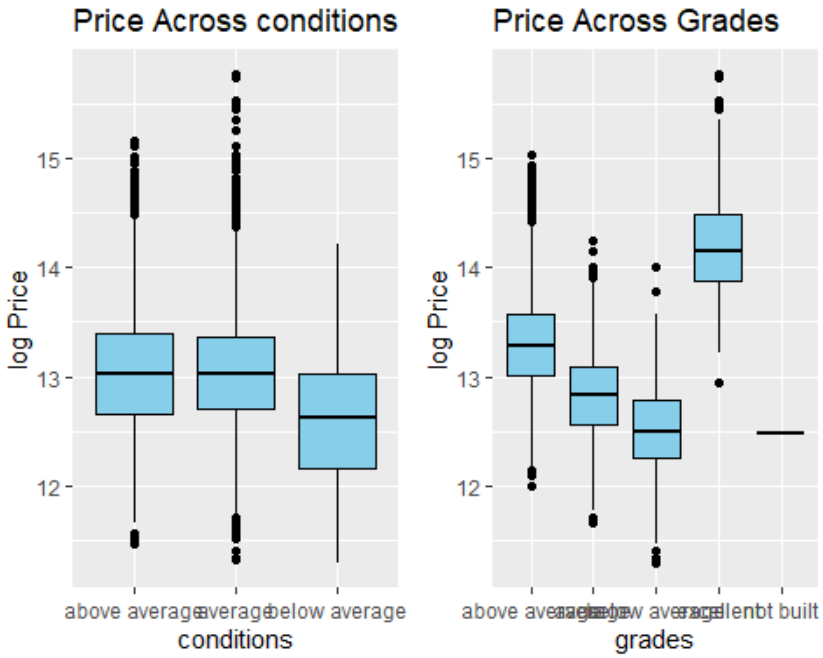
```
        addCoef.col = "black",
        col = colorRampPalette(c("blue", "white", "lightblue"))(200))
```



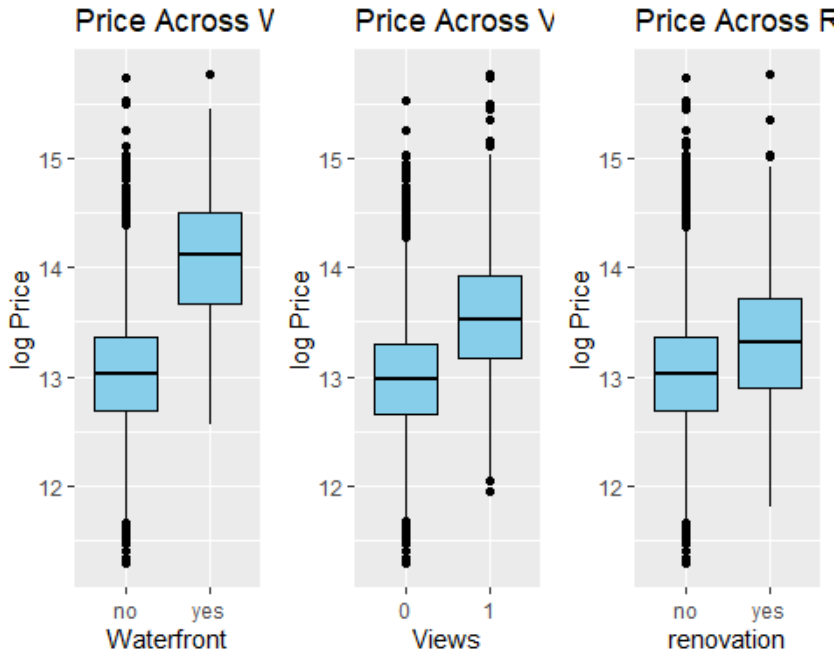
Side by side box-plots of Price across categorical variables

We created side-by-side box plots to examine the price patterns across the levels of each categorical predictor. The box plots for price by condition and grade levels show that higher condition and grade levels are associated with higher prices. We observed a similar pattern in the box plots for price by waterfront, view, renovation, and larger houses in the neighborhood, with higher levels of these factors associated with higher prices.

```
grid.arrange(ca4, ca5, ncol=2, nrow=1)
```



```
grid.arrange(ca2, ca3, ca6, ncol=3, nrow=1)
```




```
grid.arrange(ca7, ca8, ncol=2, nrow=1)
```

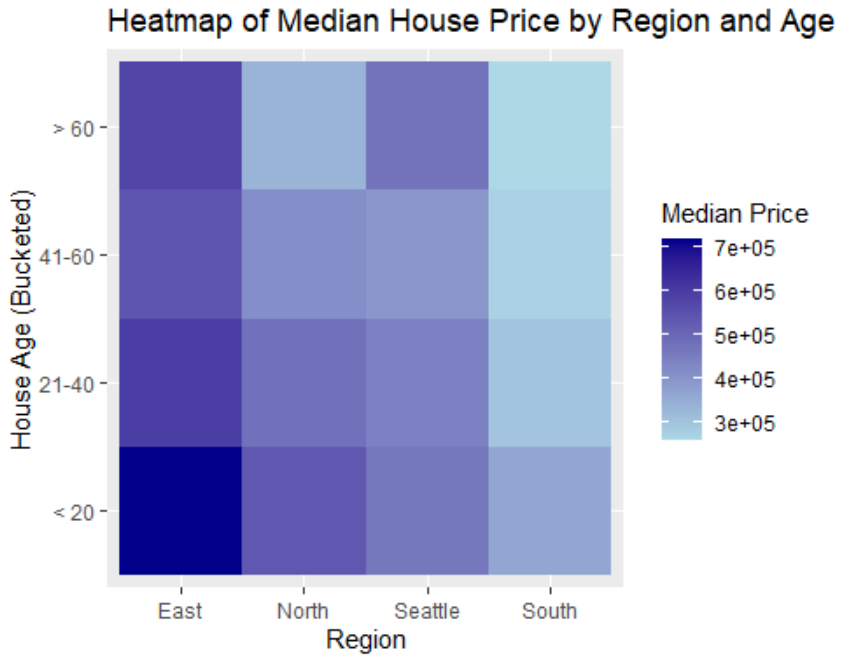
Presenting multivariate visualizations

Multivariable Heatmaps with age buckets

The previous heat map revealed a weak negative correlation between price and house age. To investigate this relationship further, we created another heat map by incorporating the categorical predictor, Region. The negative correlation (where older houses are associated with lower prices) is more pronounced in the South and North regions compared to Seattle and the East area.

We created a new variable, age_bucket, solely for the use in the following visualization. Grouping house by age will make some things easier to see.

```r
# Median Price by region with age buckets
age_bucket <- train %>%
  mutate(age_bucket = cut(age,
                          breaks = c(-Inf, 20, 40, 60, 120),
                          labels = c("< 20", "21-40", "41-60", "> 60"),
                          right = FALSE))
heatmap_data <- age_bucket %>%
  group_by(region, age_bucket) %>%
  summarise(median_price = median(price, na.rm = TRUE)) %>%
  ungroup()

## `summarise()` has grouped output by 'region'. You can override using the
## `.groups` argument.

ggplot(heatmap_data, aes(x = region, y = age_bucket, fill = median_price)) +
  geom_tile() +
  scale_fill_gradient(low = "lightblue", high = "darkblue", name = "Median Price") +
  labs(title = "Heatmap of Median House Price by Region and Age Bucket",x="Region", y = "House Age
(Bucketed)")
```



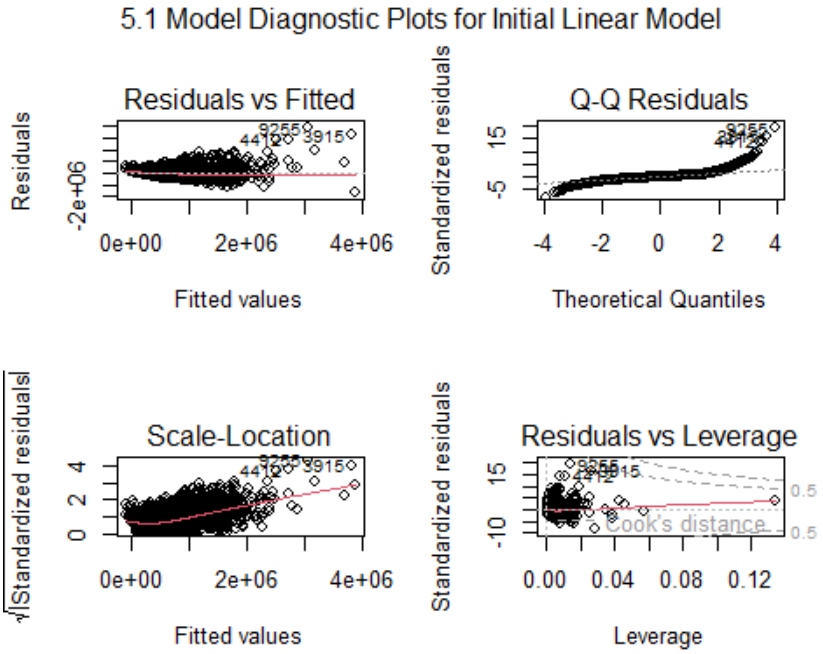
Section 5: Linear Regression

Group four used linear regression to explore the relationship between the price of homes in King county Washington and the other variables. Prior to running the linear regression, group four removed variables to be excluded from the analysis. These are noted at the end of section 3. In order to run the regression analysis we also had to remove $good\_quality$ (this for a later section) and $logprice$ (the regression needs to be run without it first).

Before running and evaluating the model selection criteria, we reviewed the $summary(\ \ )$ results and the model diagnostic plots to assess whether the linear regression assumptions are met. While R^2 is relatively strong for the model as is ($R^2 = 0.6753$), high standard errors of the regression and several coefficient terms indicate underlying issues with the initial full model. The model diagnostic plots indicate that assumptions 1 and 4 are met; however, assumption 2 is not.

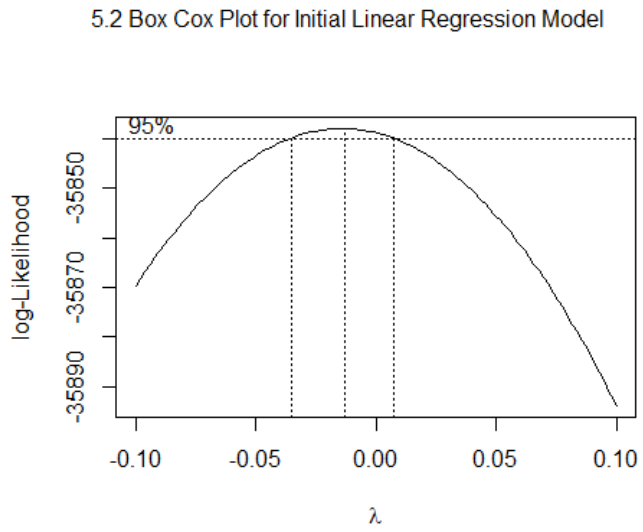
```r
# Create data frame for regression analysis that excludes variables not needed for modeling
activities (i.e. ID,
# variables modified--recoded--by the group for analysis).
trainr <- train[,!names(train) %in% c("good_quality", "logprice")]
```

```r
testr <- test[,!names(test) %in% c("good_quality", "logprice")]

# All predictor model
regfull <- lm(price ~ ., data = trainr)

regeq1 <- regfull

summary(regfull)

par(mfrow = c(2, 2))
plot(regfull)

mtext("5.1 Model Diagnostic Plots for Initial Linear Model",side=3,line=-1,outer=TRUE)
```



In order to handle assumption 2, we must address the response variable, *price*. Running a Box Cox plot indicates that the 95 CI for λ includes 0. Due to the 95 CI including 0 along with a preference for the log transformation, we will apply a log transformation to price to complete the modeling exercise.

```r
MASS::boxcox(regfull, lambda = seq(-0.1, 0.1, 1/10))
mtext("5.2 Box Cox Plot for Initial Linear Regression Model",side=3,line=-1,outer=TRUE)
```

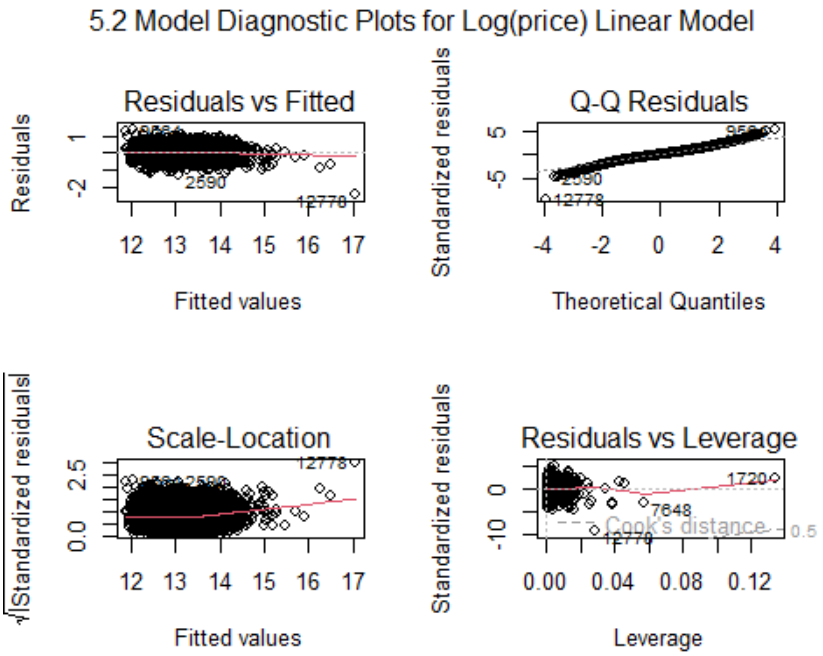


```
# Create data frame for regression analysis regresses the logarithmic transformation of price
against the selected

trainr <- train[,!names(train) %in% c("price", "good_quality")]
testr <- test[,!names(test) %in% c("price", "good_quality")]

# Intercept only model
regnull2 <- lm(logprice ~ 1, data = trainr)

# All predictor model
regfull2 <- lm(logprice ~ ., data = trainr)

summary(regfull2)
```

The log transformation of the price variable improved the model as reflected in the improved diagnostic plots. The residual standard error decreased to 0.2662 from $203{,}400$. In addition, R_p^2 increased to 0.7437.

We now have a good starting point for which applying the model selection criteria.

```
par(mfrow = c(2, 2))
plot(regfull2)
mtext("5.2 Model Diagnostic Plots for Log(price) Linear Model",side=3,line=-1,outer=TRUE)
```



Multicollinearity is unlikely to be a concern in this analysis due to several indicators. Variance Inflation Factors (VIFs) for all predictors are well below the commonly accepted threshold (e.g., 10), indicating minimal linear dependence among the independent variables. Additionally, the standard errors of the coefficient estimates are relatively small, suggesting that the precision of the estimates is not being adversely affected by multicollinearity. Furthermore, the absence of high correlations between the independent variables, supports this conclusion. Together, these factors provide strong evidence that multicollinearity is not significantly influencing the regression results.

```
quantitative_predictors <- train[, c("logprice", "bedrooms", "bathrooms",
"sqft_living","sqft_lot","floors","age")]

##correlation matrix, round to 3 decimal
#round(cor(quantitative_predictors[,-1]),3)
```

```
##VIFs
vif(quantitative_predictors)
```

First, we applied the $regsubsets(\quad)$ function from the $leaps$ package to fit all possible regression models. The adjusted R-squared: $R^2_{Adj,p}$, Mallow's C_p, and BIC all returned the same best model fit:

$$\widehat{Log(Price)} = 12.50 + 0.047bathrooms + 0.00032sqft\_living + 0.61waterfrontI_{yes} + 0.15renovatedI_{yes} - 0.0098lot\_gtI_{yes} - -0.10condition1I_{average} - 0.23grade1I_{below\ average} + 0.17grade1I_{excellent} - 0.48regionI_{South}$$

$R^2_{predicted} = 0.7117$ while $R^2_p = 0.7128$.

```
names(summary(allreg))

which.max(summary(allreg)$adjr2)
coef(allreg, which.max(summary(allreg)$adjr2))

which.min(summary(allreg)$cp)
coef(allreg, which.min(summary(allreg)$cp))

which.min(summary(allreg)$bic)
coef(allreg, which.min(summary(allreg)$bic))

regeq2 <- lm(logprice~bathrooms+sqft_living+waterfront+renovated+lot_gt+condition1+grade1+region,
data = trainr)
```

The $R^2_{predicted}$ value strongly supports the reduced model over the full model. In this case, the reduced model achieves a valid predicted value compared to the full model, indicating that the simpler model better balances complexity and predictive performance. This suggests that the additional predictors in the full model may not contribute meaningful information and could even introduce noise. Consequently, the reduced model is preferred.

Using the results from the adjusted R-squared ($R^2_{Adj,p}$), Mallow's C_p, and BIC selection criteria to reduce our model, we took this reduced set of predictors and applied the automated search procedures to determine if we can further reduce the linear model. We applied forward selection, backward elimination, and stepwise regression.

All methods produced models with the same result.

$$\widehat{Log(Price)} = 12.79 + 0.013bathrooms + 0.00027sqft\_living + 0.60waterfrontI_{yes} + 0.17renovatedI_{yes} - 0.12condition1I_{average} - 0.19condition1I_{below\ average} - 0.21grade1I_{average} - 0.41grade1I_{below\ average} + 0.23grade1I_{excellent} + 0.013grade1I_{not\ built} - 0.14regionI_{North} - 0.0083regionI_{Seattle} - 0.47regionI_{South}$$

```
# Reassign the limited model selected through
regfull2 <- regeq2

# Forward selection, starting with an intercept-only model
step(regnull2, scope=list(lower=regnull2, upper=regfull2), direction="forward")

# Backward elimination, starting with the model with all predictors
step(regfull2, scope=list(lower=regnull2, upper=regfull2), direction="backward")

# Stepwise regression, starting with an intercept-only model
step(regnull2, scope=list(lower=regnull2, upper=regfull2), direction="both")
```

The new model excludes a specific predictor variable without significantly impacting key statistical measures such as F and R^2, as well as p-values for the model and remaining variables. R^2 remains nearly unchanged, indicating

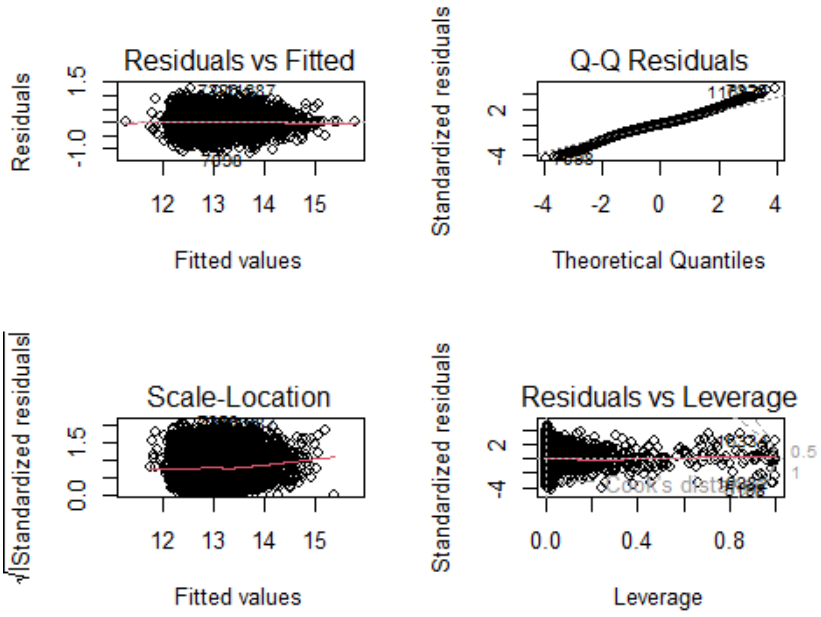
that the removed predictor contributed little to explaining the variance in the dependent variable. Similarly, the overall F-statistic is unaffected, suggesting that the model's goodness-of-fit and overall significance are retained.

The p-values for the remaining predictors also show no substantial changes, demonstrating that their relationships with the dependent variable are stable and independent of the excluded variable. This confirms that the dropped predictor was not essential to the model's explanatory or predictive power.

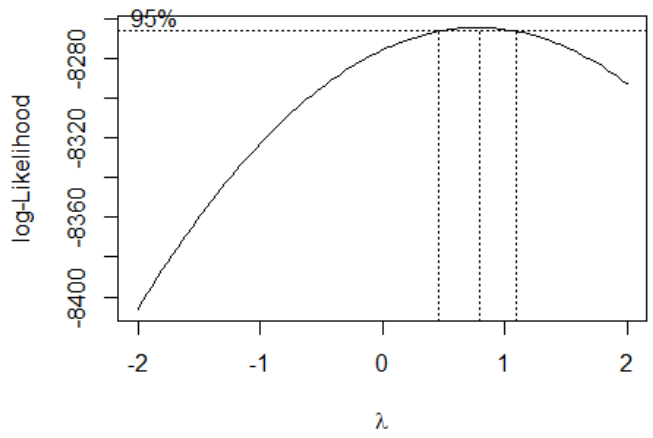
```
regeq3 <- lm(logprice~bathrooms+sqft_living+waterfront+renovated+condition1+grade1+region, data =
trainr)
summary(regeq3)
```

Analyzing the residuals of a linear model is a critical step to evaluate the model's validity and underlying assumptions. By examining residual plots and statistical metrics, we can assess whether the model appropriately captures the relationships in the data or if potential issues need to be addressed.

```
result <- lm(logprice~bathrooms*sqft_living*waterfront*renovated*condition1*grade1*region, data =
trainr)
summary(result)

par(mfrow=c(2,2))
plot(result)
```




```
MASS::boxcox(result)
```



The large sample size and small number of predictors makes comparison to the leverage threshold ($\frac{2p}{n}$) is relatively small, making it easy to identify observations with leverage values that exceed this benchmark. Due to the size of n, group 4 relied more on visual tools, to efficiently examine these points. While high leverage does not always imply undue influence, these observations warrant further scrutiny to ensure they are not unduly affecting the model's performance.

```
hii<-lm.influence(result)$hat ##Leverages
ext.student<-rstudent(result) ##ext studentized res
n<-nrow(trainr)
p<-7

length(ext.student[abs(ext.student)>3])
```

The DFBETAS analysis indicates that there are no influential observations in the dataset. This suggests that no single data point exerts an undue influence on the model's parameter estimates. As a result, the model's coefficients are stable, further supporting the validity and reliability of the regression analysis.

```
DFBETAS<-dfbetas(result)
abs(DFBETAS)>2/sqrt(n)

DFFITS<-dffits(result)
DFFITS[abs(DFFITS)>2*sqrt(p/n)]

COOKS<-cooks.distance(result)
COOKS[COOKS>1]
```

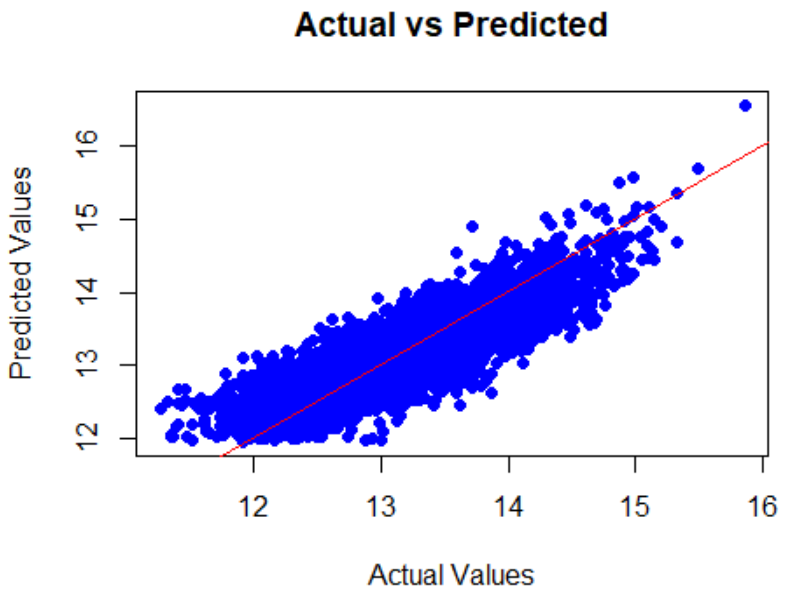
Group 4's linear regression model explores the relationship between housing prices (log-transformed as the response variable) and a set of predictors, including bathrooms, square footage of living space (sqft_living), waterfront property status, renovation status, condition, grade, and region. The model demonstrates strong explanatory power, with an R^2 of 0.7128 and an adjusted R^2 of 0.7124, indicating that approximately 71% of the variability in log-transformed prices is accounted for by the predictors. The overall model is statistically significant ($F = 2058; p < 2.2 \times 10 - 16$).

Key predictors such as sqft_living, waterfront, renovated, condition, and grade show strong statistical significance ($p < 0.001$) with coefficients aligning with expectations. For instance, larger square footage and waterfront properties are associated with higher prices, while lower condition and grade are linked to reductions in price. Regional effects are also significant, with the South region associated with a substantial price decrease compared to the reference region.

The residual analysis suggests a good fit, with a residual standard error of 0.2817 and no extreme deviations. However, some predictors, such as grade1not built and regionSeattle, show no statistical significance, indicating they may not meaningfully contribute to the model. Overall, the model provides a robust framework for understanding the factors influencing housing prices while adhering to statistical assumptions.

```
predictions <- predict(regeq3, newdata = testr)

# You can now view or use the predictions
head(predictions)

# Calculate actual values
actuals <- testr$logprice

# Mean Absolute Error (MAE)
mae <- mean(abs(predictions - actuals))
print(paste("MAE:", mae))

# Mean Squared Error (MSE)
mse <- mean((predictions - actuals)^2)
```

```r
print(paste("MSE:", mse))

# R-squared
rss <- sum((predictions - actuals)^2) # Residual Sum of Squares
tss <- sum((actuals - mean(actuals))^2) # Total Sum of Squares
r_squared <- 1 - rss/tss
print(paste("R-squared:", r_squared))

# Print results
print(paste("MAE:", mae))
print(paste("MSE:", mse))
print(paste("R-squared:", r_squared))

# Plot predictions vs actuals
plot(actuals, predictions, main = "Actual vs Predicted",
     xlab = "Actual Values", ylab = "Predicted Values", pch = 19, col = "blue")
abline(0, 1, col = "red")
```



Section 6: Data Visualizations - Good Quality Homes

The next part of this project dealt with trying to predict whether a home was of good quality or not. A good quality home is defined as a home having a condition of greater than 3 and a grade that is greater than 7. See section 2 and 3 for more detail and how this was coded. To start with this analysis, we will run visualizations about this new variable.

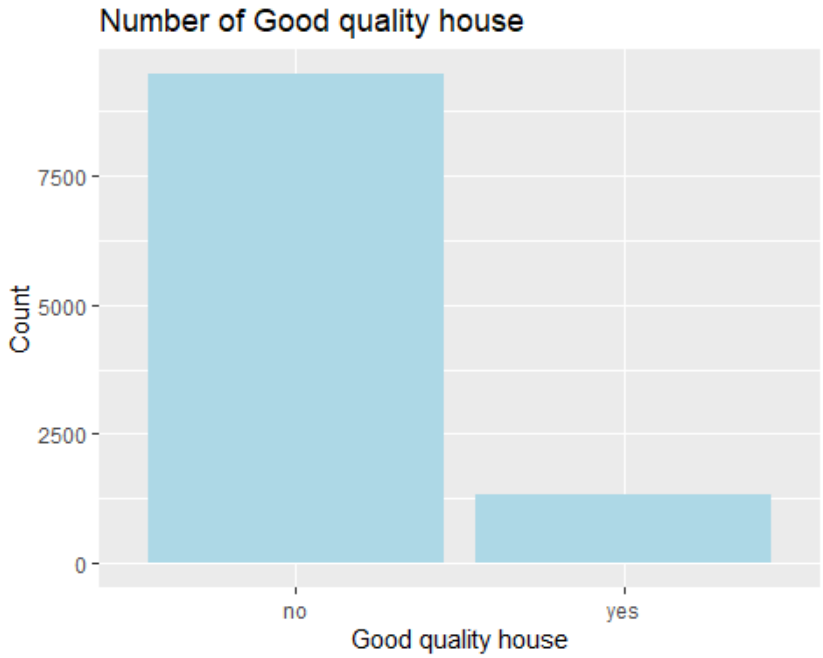
Presenting univariate visualizations

Barchart of Good quality house

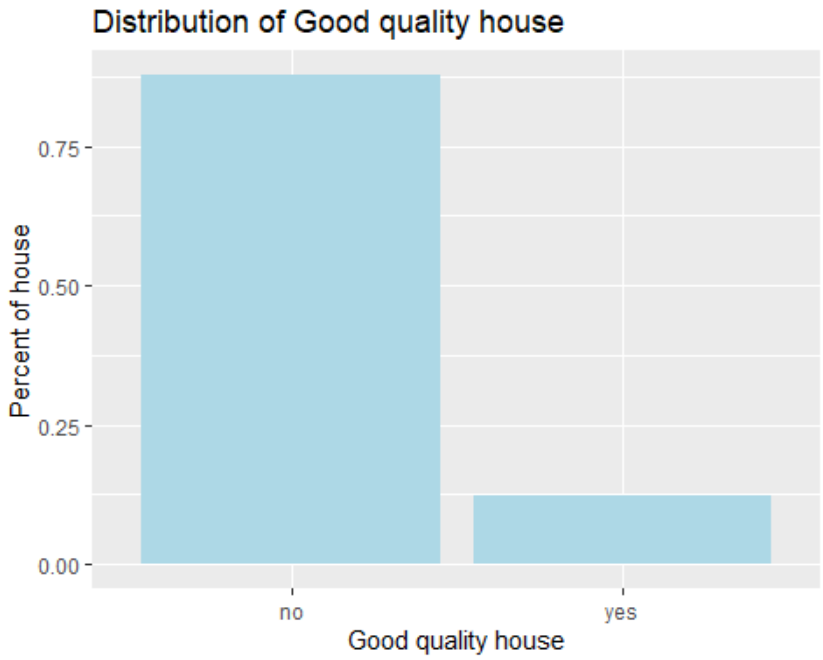
We created bar charts to analyze the distribution of Good Quality Houses, both by count and proportion. The analysis shows that 12.23% of the houses in the dataset are categorized as Good Quality, while the remaining 87.77% are not.

```r
# Barchart of Good quality house number
ggplot(train, aes(good_quality))+
```

```
  geom_bar(fill="lightblue")+
  labs(x="Good quality house", y="Count", title = "Number of Good quality house")
```




```
# Barchart of Good quality house proportion
newtrain<-train%>%
  group_by(good_quality)%>%
  summarise(Counts=n())%>%
  mutate(Percent=Counts/nrow(train))
ggplot(newtrain, aes(x=good_quality,y=Percent))+
  geom_bar(stat="identity", fill="lightblue")+
  labs(x="Good quality house", y="Percent of house", title = "Distribution of Good quality house")
```



Presenting bivariate visualizations

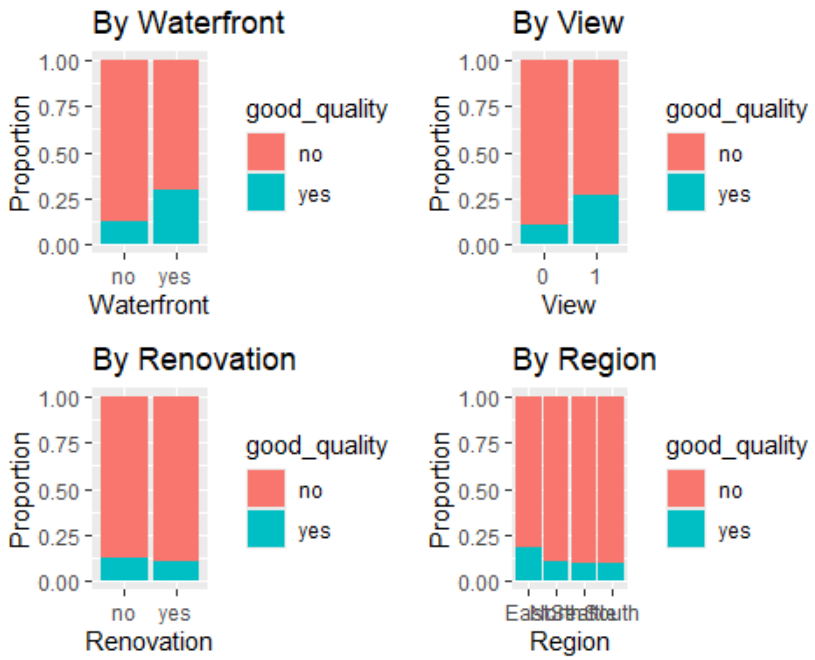
The following bar charts illustrate the distribution of Good Quality Houses across each categorical predictor, presented as proportions.

A higher proportion of Good Quality Houses is found in the East region compared to Seattle, South, and North regions.

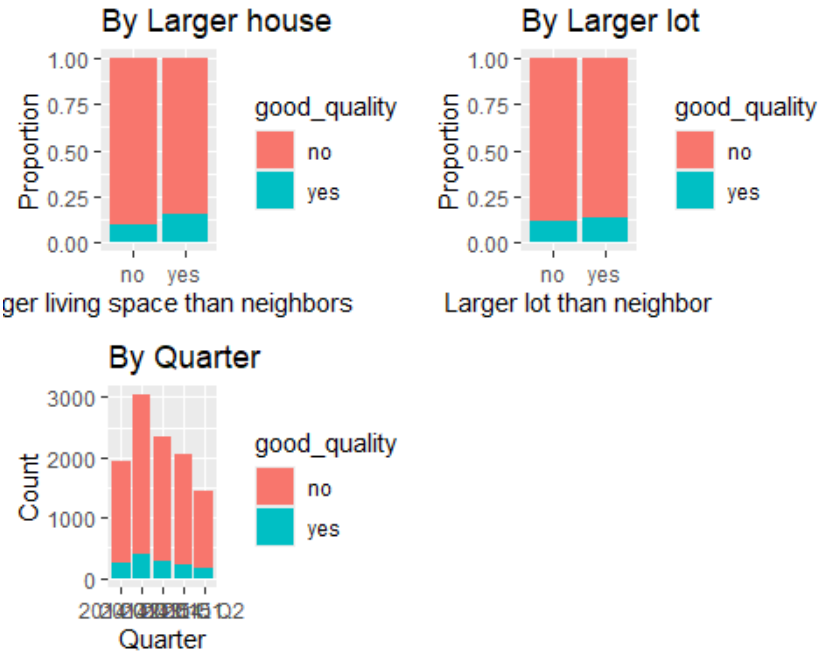
In the Renovation and Large House (living space and lot) categories, the proportions of Good Quality Houses and Non-Good Quality Houses are nearly equal. Additionally, houses with Waterfront or View are more likely to be rated as Good Quality compared to houses without these features.

The bar chart by quarter shows that the number of house sales peaked in Q3 of 2014 and then declined afterward. The proportion of Good Quality Houses remained consistent throughout the data collection period, with an approximate 15:85 ratio. This suggests that the time of sale does not influence the Good Quality rating of houses.

```
grid.arrange(chart1,chart2, chart3, chart4, ncol=2, nrow=2)
```




```
grid.arrange(chart5,chart6, chart7, ncol=2, nrow=2)
```



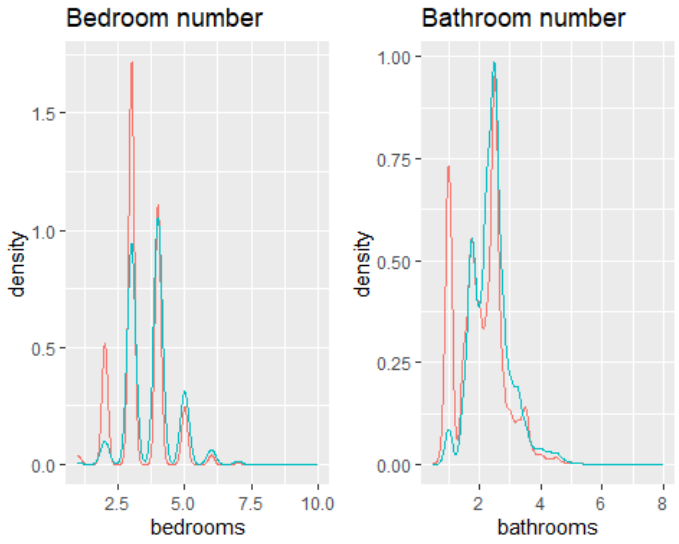
As another bivariate visualization, we created density plots to compare how the quantitative predictors differ between Good Quality Houses and Non-Good Quality Houses.We adjusted the x-axis to display values below 100,000 square feet for the sqft_lot predictor and below 1,500 square feet for the sqft_basement predictor to address extreme skewness, which hindered pattern analysis.

A higher proportion of Non-Good Quality Houses is observed among houses with a living space size of less than 2,000 square feet. A similar pattern is evident for houses with an above-ground space size of less than 1,300 square feet. This suggests that smaller houses are more frequently evaluated with lower grades and conditions compared to larger houses (living space > 2,000 square feet and above-ground space > 1,300 square feet). The Good Quality predictor was derived from the Grades and Conditions predictors.

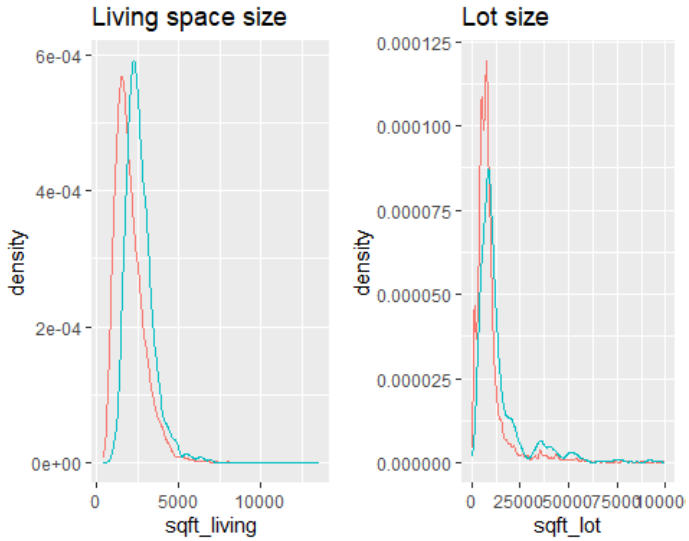
The density plot for house age reveals that a significantly higher proportion of Good Quality Houses is observed among houses aged between 20 and 60 years. In contrast, a larger proportion of houses less than 20 years old failed to receive a Good Quality rating. This suggests that newer houses are not guaranteed to achieve a Good Quality rating.

The number of bathrooms does not appear to significantly affect a house's Good Quality rating, except for houses with fewer than 1.5 bathrooms (e.g., 1 full bathroom plus at least 1 additional bathroom without a tub). This suggests that having at least 1.5 bathrooms may be a minimum requirement to avoid a Non-Good Quality rating.

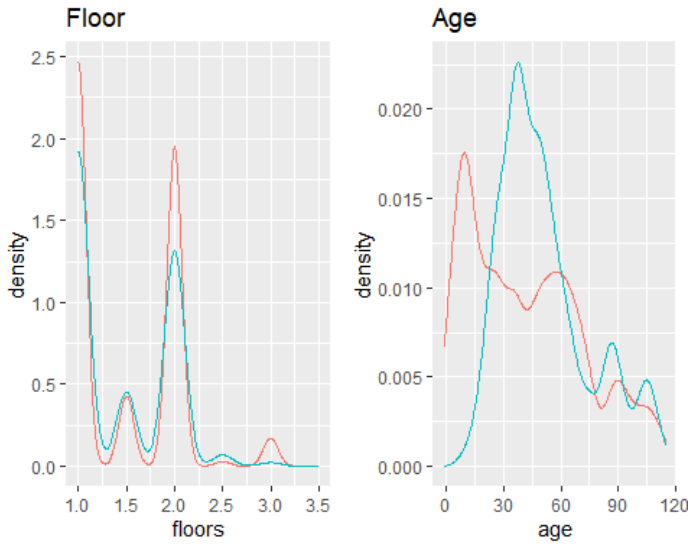
```
grid.arrange(dp1, dp2, ncol=2, nrow=1)
```




```
grid.arrange(dp3, dp4, ncol=2, nrow=1)
```



```
grid.arrange(dp5, dp6, ncol=2, nrow=1)
```



Presenting multivariate visualizations

From the bivariate visualizations, we identified several predictor variables that show noticeable differences in the proportions of Good Quality and Non-Good Quality Houses. These variables include waterfront, view, size of living space, and age. To further investigate these relationships, we performed multivariate visualizations using these four variables. We focused our investigation on houses less than 60 years old, as no significant difference in the proportion of Good Quality and Non-Good Quality Houses was observed for houses older than 60 years.

In both the less-than-20-years and 20-to-60-years age groups, Good Quality Houses with waterfront and view tend to have larger living spaces. In contrast, Non-Good Quality Houses do not exhibit this pattern. This suggests that waterfront and view are impactful features for large houses to achieve a Good Quality rating, regardless of age.

```
matrix_heatmap
```

We created multivariate scatter plots to visualize home price against the size of living space by Good quality homes and one of other categorical predictors.

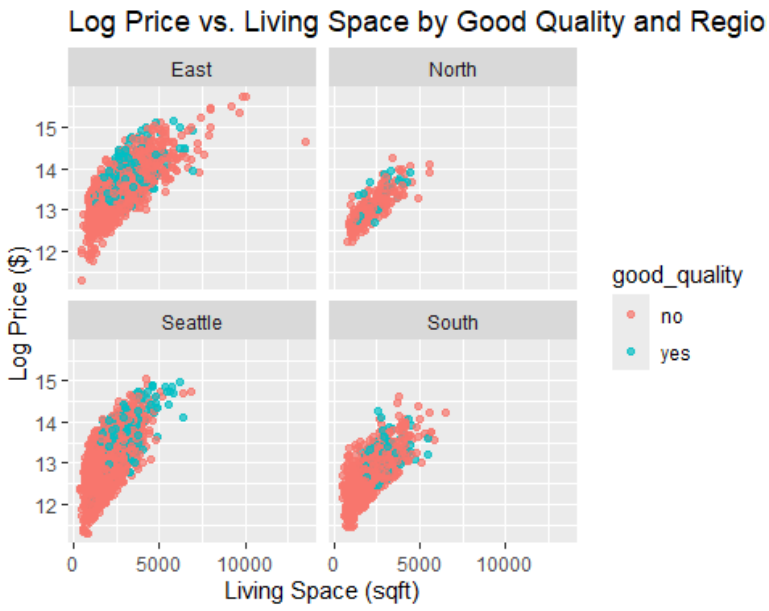
In the scatter plot of Log Price vs. Living Space by Good Quality and Region, 1. Across all regions, there is a positive linear association between log(price) and sqft_living, indicating that larger homes generally have higher prices. 2.Good quality houses tend to cluster in the upper right side of the plots, suggesting that good quality homes are larger and command higher prices compared to lower-quality ones. 3.This trend is most pronounced in Seattle and the South, where the distinction between good quality homes and others is more apparent. In contrast, the trend is less clear in the North.

In the scatter plot of Log Price vs. Living Space by Good Quality and Quarter, 1. Across all five quarters, there is a positive linear association between log(price) and sqft_living, showing that larger homes generally have higher prices. 2.However, there is no noticeable difference among the quarters in terms of the relationship between log(price) and sqft_living. This suggests that the market conditions affecting house prices remained relatively stable over the observed time period, without significant temporal trends impacting the relationship.

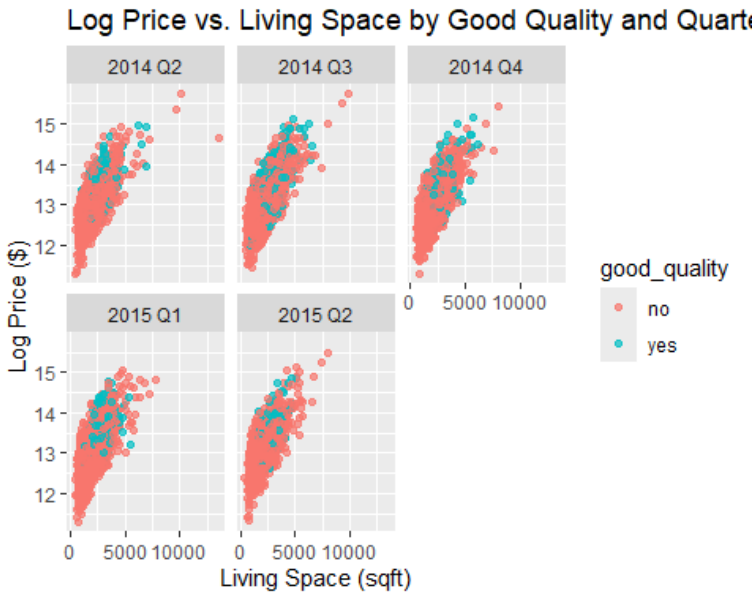
In each scatter plot of Log Price vs. Living Space by Good Quality and Renovation/View/Waterfront, 1.In all scatter plots, there is a positive linear association between log(price) and sqft_living, regardless of whether the house has a renovation, view, or waterfront feature.This indicates that house size consistently correlates with price, regardless of these additional attributes. 2.Separation of Good Quality vs. Non-Good Quality Homes in Houses without Renovation, View, or Waterfront:There is a clear separation between good quality and non-good quality homes, with good quality houses clustering in the upper right (larger size, higher price) and non-good quality houses spread across lower price ranges. This suggests that good quality homes stand out more distinctly in the absence of these additional features. 3.Separation of Good Quality vs. Non-Good Quality Homes in Houses with Renovation, View, or Waterfront:The distinction between good quality and non-good quality homes becomes less pronounced. These features may enhance the value of all homes, regardless of their intrinsic quality, making the separation less noticeable. This implies that renovation, scenic views, or waterfront locations might level the playing field for lower-quality homes by adding significant value.

In conclusion, the findings from the multivariate scatter plots are consistent with those observed in the previous bivariate visualizations.

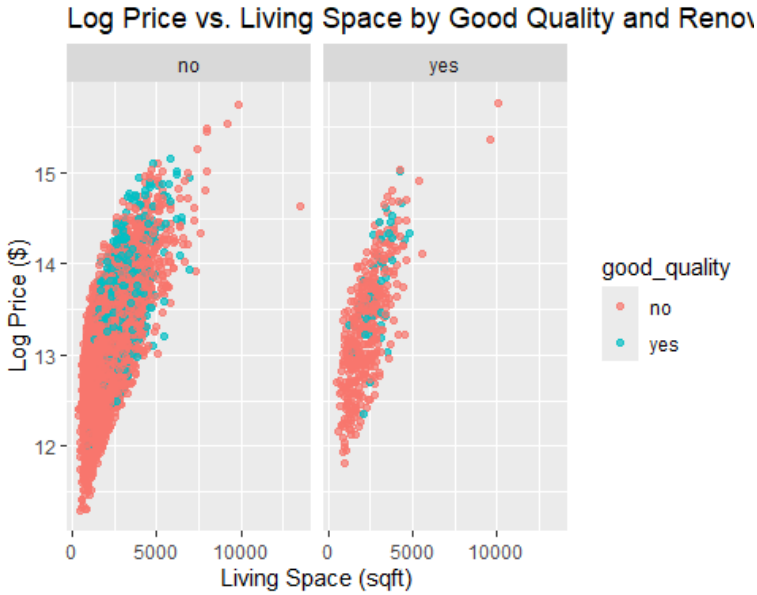
```r
ggplot(train, aes(x = sqft_living, y = log(price), color = good_quality)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~region) +
  labs(
    title = "Log Price vs. Living Space by Good Quality and Region",
    x = "Living Space (sqft)",
    y = "Log Price ($)",
    color = "good_quality"
  )
```



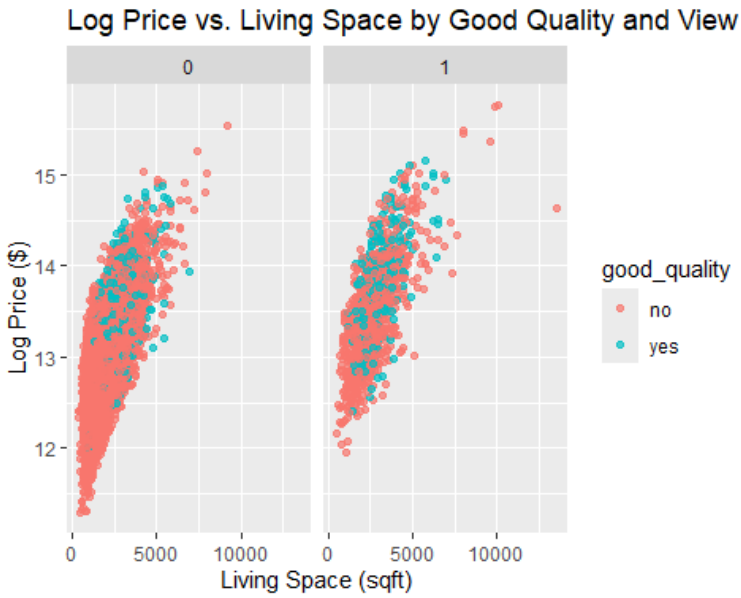
```r
# Log Price vs. Living Space by Good Quality and Quarter
ggplot(train, aes(x = sqft_living, y = log(price), color = good_quality)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~quarter) +
  labs(
    title = "Log Price vs. Living Space by Good Quality and Quarter",
    x = "Living Space (sqft)",
    y = "Log Price ($)",
    color = "good_quality"
  )
```



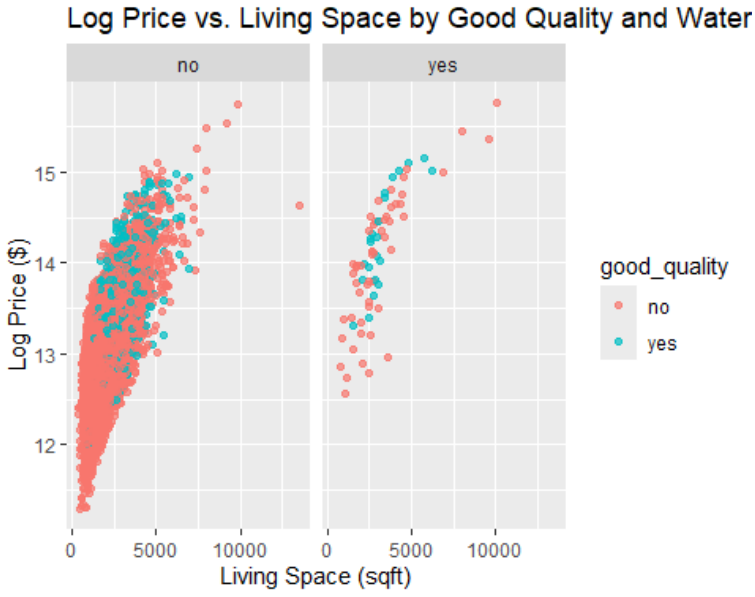

```r
# Log Price vs. Living Space by Good Quality and Renovation
ggplot(train, aes(x = sqft_living, y = log(price), color = good_quality)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~renovated) +
  labs(
    title = "Log Price vs. Living Space by Good Quality and Renovation",
    x = "Living Space (sqft)",
    y = "Log Price ($)",
    color = "good_quality"
  )
```



```r
# Log Price vs. Living Space by Good Quality and View
ggplot(train, aes(x = sqft_living, y = log(price), color = good_quality)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~view1) +
  labs(
    title = "Log Price vs. Living Space by Good Quality and View",
    x = "Living Space (sqft)",
    y = "Log Price ($)",
    color = "good_quality"
  )
```




```r
# Log Price vs. Living Space by Good Quality and Waterfront
ggplot(train, aes(x = sqft_living, y = log(price), color = good_quality)) +
  geom_point(alpha = 0.7) +
  facet_wrap(~waterfront) +
  labs(
    title = "Log Price vs. Living Space by Good Quality and Waterfront",
    x = "Living Space (sqft)",
    y = "Log Price ($)",
    color = "good_quality"
  )
```



Section 7: Logistic Regression

The goal is to now see if we can use the data to create a model that predicts whether a home is of good quality or not. To do this we ran a logistic regression due to the binomial nature of the good_quality variable. We first ran a logistic regression based on the full model from the training data we set up earlier. THe summary and VIFs were also ran.

```
#Create full regression that includes all variables of interest. Print summary of results and
VIFs.
regfull_logit <- glm(good_quality ~ logprice + bedrooms + bathrooms + sqft_living + age + region +
sqft_lot + renovated, family = binomial, data = train)
summary(regfull_logit)
sort(vif(regfull_logit))
```

All variables are statistically significant based on their p-values apart from sqft_living, regionNorth, and sqft_lot. The VIFs are higher than ten for almost every variable suggesting the presence of multicollinearity.

We then ran the automatic search procedures (forward selection, backward elimination, and stepwise) to see if there was a better model. We also tried to optimize the model by optimizing the adjusted R^2, Mallow's C_p, and BIC.

```
#Create intercept-only model and use step function to find preferred model based on AIC. All
directions checked.
regnull_logit <- glm(good_quality ~ 1, family = binomial, data = train)

step(regnull_logit, scope = list(lower = regnull_logit, upper = regfull_logit), direction =
"forward")

step(regfull_logit, scope = list(lower = regnull_logit, upper = regfull_logit),
direction="backward")

step(regnull_logit, scope = list(lower = regnull_logit, upper = regfull_logit), direction =
"both")
```

The model using region1 has a slightly lower AIC than the model using the original region variable. The VIFs between the two models are not very different. Additionally, if we were to evaluate the North region using the Wald test, the Wald test would suggest removing the North region since the p-value is significantly above 0.05. Therefore, based on the lower AIC and the Wald test, we choose to use region1 over the original region variable.

```
#Use regsubsets function to check for model with highest adjusted R-squared, lowest Mallow's Cp,
and lowest BIC.
allreg_logit <- regsubsets(good_quality ~ logprice + bedrooms + bathrooms + sqft_living + sqft_lot
+ age + region + renovated, data = train, method = "exhaustive")
summary(allreg_logit)

which.max(summary(allreg_logit)$adjr2)

which.min(summary(allreg_logit)$cp)

which.min(summary(allreg_logit)$bic)
```

To evaluate which model is preferred based on the different methods, we use the regsubsets and step functions. Using regsubsets, we can evaluate which model is preferred based on the highest adjusted R-squared, the lowest value for Mallow's Cp, or the lowest Bayesian information criterion (BICp). The model with the highest adjusted R-squared includes the following predictors: log(price), bedrooms, bathrooms, age, region, and renovated. The model with the lowest Mallow's Cp includes the same variables as the model chosen by the adjusted R-squared but excludes the North region. The model with the lowest BICp is similar to the model chosen by the lowest Mallow's Cp but also drops bathrooms. Using the step function, the model with the lowest Akaike information criterion (AICp) is the same model identified by the highest adjusted R-squared which only removes sqft_living and sqft_lot. The same model is chosen regardless of the chosen direction the step function works from. Given that

different models were preferred based on the different methods, we must choose which model to use. Since sqft_living and sqft_lot are removed in every method, we remove those variables from the model. Since two methods dropped the North region, we created a new variable, region1, that combines the North region with the East region. We compare two models: one using region and the other using region1. The other predictor variables used in both models include log(price), bedrooms, bathrooms, age, and renovated.

```
#Two models to evaluate: model using original region variable and model using new region variable
that combines North and East regions. Both models exclude sqft_living and sqft_lot based on
previous tests.
reduced1a_logit <- glm(good_quality ~ logprice + age + region + renovated + bedrooms + bathrooms,
family = binomial, data = train)
summary(reduced1a_logit)
sort(vif(reduced1a_logit))

train$region1 <- ifelse(train$region == "Seattle", "Seattle", ifelse(train$region == "South",
"South", "NorthandEast"))

reduced1b_logit <- glm(good_quality ~ logprice + bedrooms + bathrooms + age + region1 + renovated,
family = binomial, data = train)
summary(reduced1b_logit)
sort(vif(reduced1b_logit))
```

The model using region1 has a slightly lower AIC than the model using the original region variable. The VIFs between the two models are not very different. Additionally, if we were to evaluate the North region using the Wald test, the Wald test would suggest removing the North region since the p-value is significantly above 0.05. Therefore, based on the lower AIC and the Wald test, we choose to use region1 over the original region variable. Looking at the model above that uses region1, all the VIFs are high which could be concerning, indicating multicollinearity. To test if any one variable is creating the high VIFs, we remove one variable one at a time and run the model for each version.

Drop one variable at a time and see how VIFs change. 6 models created.

```
reduced2a_logit <- glm(good_quality ~ logprice + age + region1 + renovated + bedrooms, family =
binomial, data = train)
summary(reduced2a_logit)
sort(vif(reduced2a_logit))

reduced2b_logit <- glm(good_quality ~ logprice + region1 + renovated + bedrooms + bathrooms,
family = binomial, data = train)
summary(reduced2b_logit)
sort(vif(reduced2b_logit))

reduced2c_logit <- glm(good_quality ~ age + region1 + renovated + bedrooms + bathrooms, family =
binomial, data = train)
summary(reduced2c_logit)
sort(vif(reduced2c_logit))

reduced2d_logit <- glm(good_quality ~ logprice + age + region1 + renovated + bathrooms, family =
binomial, data = train)
summary(reduced2d_logit)
sort(vif(reduced2d_logit))

reduced2e_logit <- glm(good_quality ~ logprice + age + renovated + bedrooms + bathrooms, family =
binomial, data = train)
summary(reduced2e_logit)
sort(vif(reduced2e_logit))

reduced2f_logit <- glm(good_quality ~ logprice + age + region1 + bedrooms + bathrooms, family =
binomial, data = train)
summary(reduced2f_logit)
sort(vif(reduced2f_logit))
```

In each case, the VIFs remain relatively constant. Since the VIFs are not very different when one of the variables is removed and the range of the values is not large, we choose to keep all variables in the model. The model we chose is as follows:

good_quality = -28.72 + 1.88*logprice* + *0.13*bedrooms + 0.14*bathrooms* + *0.03*age - 1.25*region1Seattle* + *0.61*region1South – 1.78*renovated

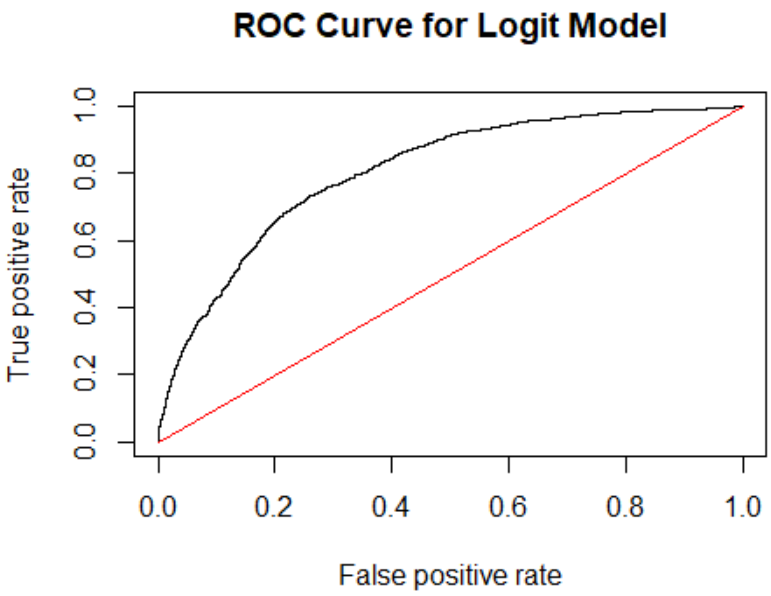
The coefficients are in log odds so we convert them to odds by taking the exponential of each coefficient.

```
#Transform coefficients from log odds to odds.
logit_model <- reduced1b_logit
exp(coef(logit_model)["logprice"])
exp(coef(logit_model)["bedrooms"])
exp(coef(logit_model)["bathrooms"])
exp(coef(logit_model)["age"])
exp(coef(logit_model)["renovatedyes"])
exp(coef(logit_model)["region1Seattle"])
exp(coef(logit_model)["region1South"])
```

Price, bedrooms, bathrooms, age, and the South region are all estimated to increase the probability of a home being considered to have good quality while being renovated at some point and the Seattle region appear to decrease the probability, all else held equal. More specifically, the estimated odds of a home being of good quality are multiplied by 6.58 for every one percent increase in price, holding all other variables constant. For bedrooms, the estimated odds of a home being considered good quality is multiplied by 1.14 for every additional bedroom, controlling for the other variables. The estimate for bathrooms is very close to the estimate for bedrooms at 1.16. Age has a smaller effect on a per-unit basis with the odds of a home being good quality multiplied by 1.03 for every additional year of age of the house, all else equal. If a house has been renovated, the odds of being good quality are multiplied by 0.17 compared to a home that has not been renovated. Lastly, we have the various regions showing strong effects on the odds of being good quality where a house being in Seattle, compared to the North or East, multiplies the odds by 0.29 while a house in the South, compared to the North or East, multiplies the odds by 1.85.

To test how well the model predicts when a house is of good quality, we first create the receiver operating characteristic (ROC) curve using the test data and is shown below.

```
#Create ROC curve.
test$logprice <- log(test$price)
test$region1 <- ifelse(test$region == "Seattle", "Seattle", ifelse(test$region == "South",
"South", "NorthandEast"))

preds <- predict(logit_model, newdata = test, type="response")

rates <- prediction(preds, test$good_quality)
roc_result <- performance(rates, measure = "tpr", x.measure = "fpr")
plot(roc_result, main = "ROC Curve for Logit Model")
lines(x = c(0, 1), y = c(0, 1), col = "red")
```



Calculate AUC.

```
auc <- performance(rates, measure = "auc")
auc@y.values
```

Since the ROC curve is above the red line, pictured above, at all times, the true positive rate is greater than the false positive rate at every threshold value, indicating the model is better than random guessing. Next, we calculate the area under the curve (AUC) which measures the area under the ROC curve. The AUC is 0.806 which suggests that, since it is greater than 0.50, that the model performs better than random guessing.

Percentage of observations where good_quality == yes: 13.94% (unbalanced)

Create confusion matrix with 0.5 threshold.

```
confusion <- table(test$good_quality, preds > 0.5)
confusion
```

Lastly, we create a confusion matrix using a threshold of 0.5, shown below.

```
#Calculate error rate, accuracy rate, false positive rate, false negative rate, and true positive
rate.
TN1 <- confusion[1]
FN1 <- confusion[2]
FP1 <- confusion[3]
TP1 <- confusion[4]

n <- (TN1 + FN1 + FP1 + TP1)

error_rate1 <- (FP1 + FN1) / n
error_rate1

accuracy1 <- (TN1 + TP1) / n
accuracy1

fpr1 <- FP1 / (TN1 + TP1)
fpr1

fnr1 <- FN1 / (FN1 + TP1)
```

```
fnr1

tpr1 <- TP1 / (FN1 + TP1)
tpr1
```

The error rate is 12 percent and the accuracy rate is 88 percent. We also calculate the false positive rate, false negative rate and true positive rate at 1.3, 87.8 and 12.2 percent, respectively. However, the response variable is very unbalanced with only about 14 percent of the homes in the data being considered good quality so the threshold should be lowered to compensate for this unbalance. We lowered the threshold to 0.2 and present the results below.

```
confusion2 <- table(test$good_quality, preds > 0.2)
confusion2

#Calculate error rate, accuracy rate, false positive rate, false negative rate, and true positive
rate.
TN2 <- confusion2[1]
FN2 <- confusion2[2]
FP2 <- confusion2[3]
TP2 <- confusion2[4]
n <- (TN2 + FN2 + FP2 + TP2)

error_rate2 <- (FP2 + FN2) / n
error_rate2

accuracy2 <- (TN2 + TP2) / n
accuracy2

fpr2 <- FP2 / (TN2 + TP2)
fpr2

fnr2 <- FN2 / (FN2 + TP2)
fnr2

tpr2 <- TP2 / (FN2 + TP2)
tpr2
```

Now, the error rate is 17.7 percent and the accuracy rate is 82.3 percent. The false positive rate, false negative rate and true positive rate is 14, 49.7 and 50.3 percent, respectively. Based on these results, we conclude that the model is better than random guessing.