Matthew Bolding
Dr. Nelis Potgieter
MATH 40853–015
April 28th, 2023

Regression and Time Series: Project 2

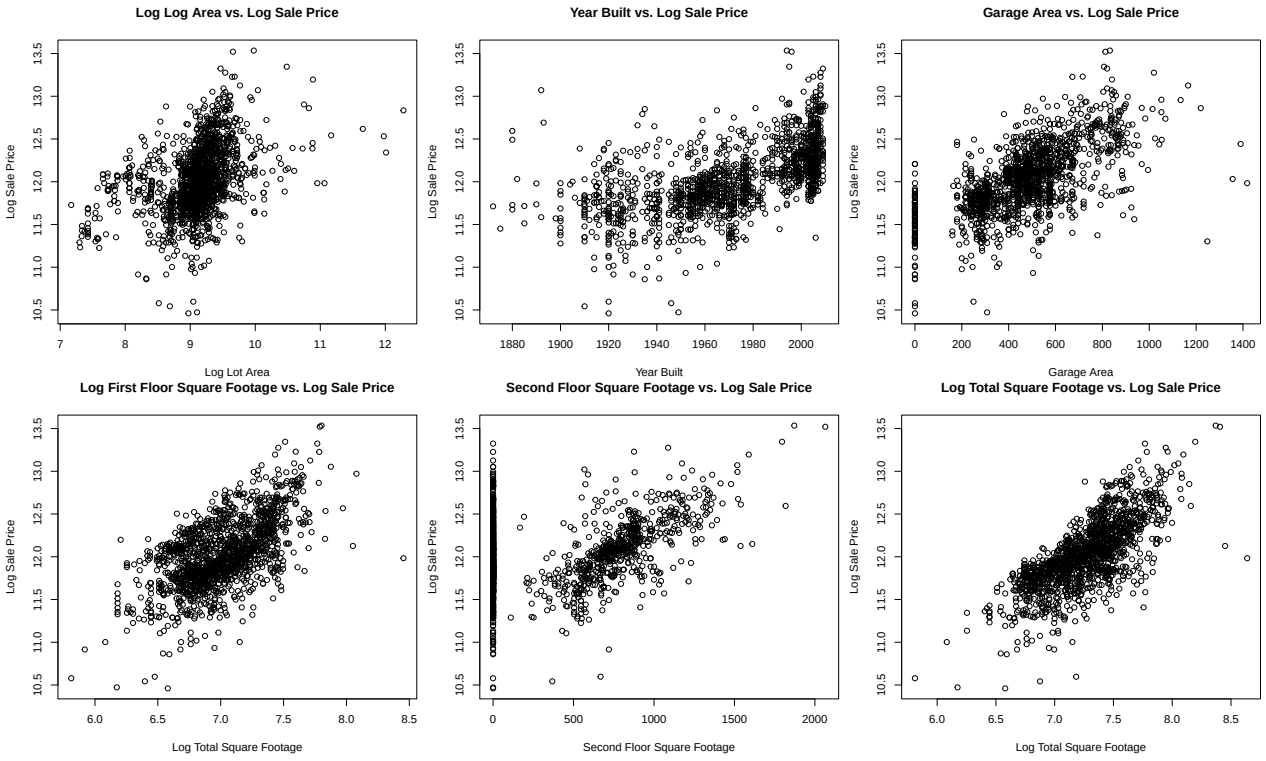
1 Data Preprocessing and Visualization

Before starting with the task *Zillot* has presented us with, we must first consider which, if any, variables should be log-transformed. After importing the data, we may calculate the correlation coefficient between a numeric explanatory variable and the outcome variable, considering all four combinations of log-transformed and non-log-transformed variables. The maximum of these values will indicate which predictors should be log-transformed. The goal is to make the relationship linear, so this method will improve our predictive ability with a *linear* model.

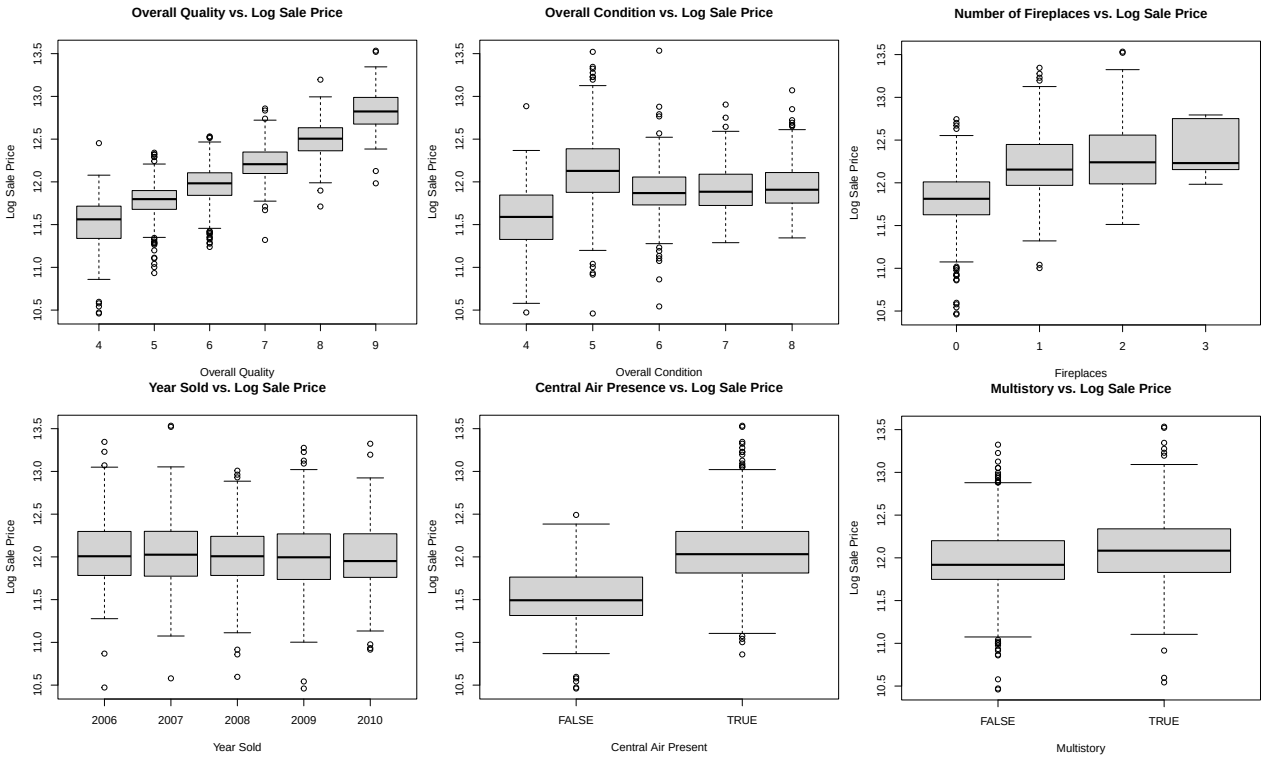
With this criterion in mind, we find that in all situations—except for one special case, `Flr2SF`, due to the high number of observations with no second floor square footage—`SalePrice` should be log-transformed. We also find that `LotArea`, `Flr1SF`, and `TotalSF` should be log-transformed.

We also define two new predictor variables: a numeric predictor `TotalSF` which is the sum of `Flr1SF` and `Flr2SF`, and a categorical predictor `Multistory` which indicates whether a house has more that one story. In addition to creating these two variables, we also group categories with less than 50 observations. Code for these steps can be found here.

The below scatter plots reflect the log-transform decisions. Observe the linear relationships between the variables.



We also have the following box plots.



2 Traning Models

Following the data preprocessing, we have the following variables as numerical predictors: `LotArea`, `YearBuilt`, `Flr1SF`, `Flr2SF`, `TotalSF`, `Fireplaces`, `GarageArea`, and `YearSold`. Categorical predictors include `OverallQual`, `OveralCond`, `CentralAir`, and `Multistory`. These predictor variables work to explain the numerical outcome variable `SalePrice`.

2.1 Forward Selection

One method we may train a model is via forward selection. This process adds variables to the model that most improves the model, starting with just the intercept term as the base model. To facilitate this process, we use the `step` function, which goes about performing forward selection using the criterion AIC—a measure of how well the model fits the data. Note that, on its own, AIC does not mean much; AIC is used to compare two model's adherence to the dataset. The smaller the AIC, the better. Since *Zillot* wants the model that best fits the data, using AIC as a metric will allow us to execute on their request.

After the forward selection process competes, we find that this model includes predictors `OverallQual`, `log(TotalSF)`, `YearBuilt`, `OverallCond`, `log(LotArea)`, `Multistory`, and `GarageArea`. This model has an R^2_{adj} value—the proportion of variability in the dataset explained by the model—of 0.8613653 and an AIC of -5549.4832714.

2.2 Best Subset

Another method of training is best subset. This method trains all potential models from the supplied predictors, including an interaction term between `log(TotalSF)` and `Multistory`. Since *Zillot* limits models to a maximum of seven predictor variables, the chosen model has seven variables, despite a model with twelve predictor variables maxing out the models predictive accuracy. The metric by which the best model is chosen is through R^2_{adj}. Since our goal is to train as accurate a model as possible for *Zillot* to predict the sale price of a house, R^2_{adj} is a valid criterion by which to select a model.

After computing all models, we find that the best model includes `log(LotArea)`, `OverallQual`, `OverallCond`, `YearBuilt`, `Fireplaces`, `GarageArea`, and an interaction term between `log(TotalSF)` and `Multistory` as predictors. This model has an R^2_{adj} of 0.8655355 and an AIC of -5593.0861409.

2.3 Backwards Selection

Opposite to the forward selection paradigm, backwards select takes the full model, one with all predictor and removes a predictor which fits some criterion. In this case, we will remove the predictor which decrease the adjusted R^2 value the least. We proceed in this fashion because we want to maximize our adjusted R^2 so that the model explains as much of the variability in the dataset as possible. Unlike the past two methods which used a library function, this process is manually implemented. Like all other model selection procedures, we have a model with seven predictor variables; any combination of the six variables yields a decrease in the adjusted R^2 value.

Additionally, even from the beginning of this model's training, we do not consider square footage—not the first floor, second floor, or the combined square footage—as predictors. *Zillot* imposes a restriction that we provide at least one model without this parameter.

As the junior statistician working on *Zillot's* behalf, I believe that removing these predictor variables in this model, as opposed to the best subset or forward selection model, does not have a large impact on the final model; it was an arbitrary decision to train backwards selection without these predictors. Since the correlation between `log(TotalSF)` and `log(SalePrice)` is so high—0.7371302—you'd expect to see the model not including this key predictor to suffer. (As a matter of fact, the best subset method even confirms that model obtainable with no more than seven predictors excluding square footage has an adjusted R^2 value of 0.8127. By happenstance, that model is the very model trained here.)

Nevertheless, we find that the backward selection model includes predictor variables `log(LotArea)`, `OverallQual`, `OverallCond`, `YearBuilt`, `Fireplaces`, `GarageArea`, and `Multistory`. It has an adjusted R^2 value of 0.8127217 and an AIC of -5110.3836548.

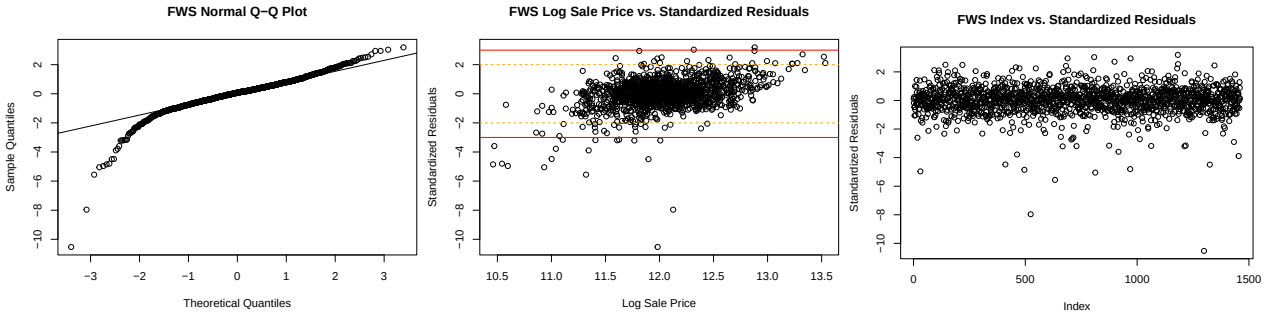
3 Model Evaluation

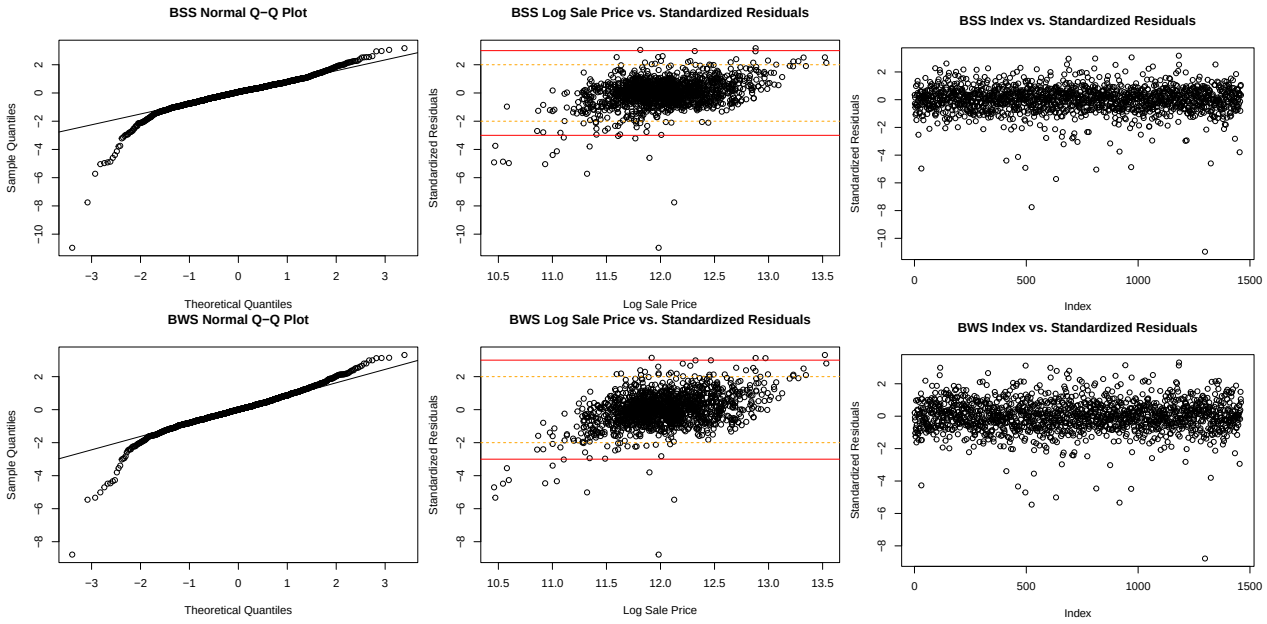
In some sense, it comes without surprise that the model created via the method of best subsets yields the best model, in this case, both in terms of adjusted R^2 and AIC. This model explains 86.6% of the variability in the dataset.

3.1 Residual Analysis

See code for the plots and residual calculations here.

For the discussion on the model's residual analyses, we will use the the following acronyms: FWS corresponds to the model obtained via forward selection, BSS for the best subset mode, and BWS for the backward select model. Although the BSS model is the best in terms of AIC and adjusted R^2, all the models performed rather similarly and subsequently, all the plots for the residual analysis look very similar.





The **Log Sale Price vs. Standardized Residuals** plots show few differences. Recall that only about 5% of the standardized residuals should have an absolute value greater than three for a normal dataset. We find that 1.369863% of the residuals for FWS fall outside this boundary; 1.1643836% of residuals for BSS fall outside the boundary; and 1.1643836% of the residuals for BWS fall outside. There's no large difference—although it is interesting that the BWS model, which does not include square footage as a predictor, shares the lowest proportion of residuals outside the boundary with the BSS model. We can also observe one or two observations that have quite large residuals across all models, and since a linear model's coefficients are weighted averages, these observations, which are potentially outliers, might have an impact on the coefficients and the predictive accuracy. But since this dataset is so large, there's no cause for concern with these few data points. Interesting, we may also view by inspection that a majority larger-valued standardized residuals are negative, indicating that the model tends to overestimate the sale price.

The **Normal Q-Q** plots tell a similar story—the models perform similarly and the data appears to be reasonably normal. However, on closer, qualitative inspection, we can see that on the upper end of the theoretical quantiles for the BWS mode, the sample quantiles start to deviate from the normal line faster than for the other two models. On the lower end of the theoretical quantiles, all the models drop off the normal line by a fair margin, yet the BWS model has smallest minimum sample quantile.

Finally, the **Index vs. Standardized Residuals** plots do not show anything groundbreaking. The index and the standardized residuals do not share a relationship, so the dataset does not suffer from heteroscedasticity.

4 Final Recommendations

I would recommend for *Zillot* to use the best subsets model; it has the best characteristics of all models trained in this report, and does a rather good job at explaining the variability in the dataset. I would recommend to *Zillot*, however, that other variables should be considered for the sale price of a home, namely those that don't have anything to do with the house itself. How close is it to the nearest hospital? What the surrounding area's crime rate? How good is the house's school district? How close is the nearest park or public recreation area? What's the trend in the house's property value over the last couple of years? Is it in a gated community? Is there an HOA? These factors, along with increasing the number of permitted predictor variables to include these non-intrinsic factors might lend themselves to obtaining a better, more predictive, and complete model, one that captures aspects not only of the house itself but of the surrounding area too.

5 Appendix

Here is all the code I used to compile this report.

Importing the data and performing preprocessing.

```
housing_data <- read.csv("housing_data.csv")

# Create new predictor variables.
housing_data$TotalSF <- housing_data$Flr1SF + housing_data$Flr2SF
housing_data$Multistory <- as.factor(ifelse(housing_data$Flr2SF == 0, FALSE, TRUE))

housing_data$CentralAir <- as.factor(ifelse(housing_data$CentralAir == "Y", TRUE, FALSE))

# Create new groupings.
# Merge groups 1, 2, and 3 into 4.
housing_data$OverallQual <- ifelse(housing_data$OverallQual %in% c(1, 2, 3), 4,
                                   housing_data$OverallQual)
# Merge group 10 into 9.
housing_data$OverallQual <- ifelse(housing_data$OverallQual %in% c(10), 9,
                                   housing_data$OverallQual)

# Merge groups 1, 2, and 3 into 4.
housing_data$OverallCond <- ifelse(housing_data$OverallCond %in% c(1, 2, 3), 4,
                                   housing_data$OverallCond)
# Merge group 9 into 8.
housing_data$OverallCond <- ifelse(housing_data$OverallCond %in% c(9), 8,
                                   housing_data$OverallCond)

# Set OverallQual and OverallCond as categorical predictors.
housing_data$OverallQual <- as.factor(housing_data$OverallQual)
housing_data$OverallCond <- as.factor(housing_data$OverallCond)
```

Determining which numeric predictors should be log-transformed.

```
# LotArea
cor((housing_data$LotArea), (housing_data$SalePrice))
cor(log(housing_data$LotArea), (housing_data$SalePrice))
cor((housing_data$LotArea), log(housing_data$SalePrice))
cor(log(housing_data$LotArea), log(housing_data$SalePrice)) # Maximum

# YearBuilt
cor((housing_data$YearBuilt), (housing_data$SalePrice))
cor(log(housing_data$YearBuilt), (housing_data$SalePrice))
cor((housing_data$YearBuilt), log(housing_data$SalePrice)) # Maximum
cor(log(housing_data$YearBuilt), log(housing_data$SalePrice))

# GarageArea
cor((housing_data$GarageArea), (housing_data$SalePrice))
cor(log(housing_data$GarageArea), (housing_data$SalePrice))
cor((housing_data$GarageArea), log(housing_data$SalePrice)) # Maximum
cor(log(housing_data$GarageArea), log(housing_data$SalePrice))

# Flr1SF
cor((housing_data$Flr1SF), (housing_data$SalePrice))
cor(log(housing_data$Flr1SF), (housing_data$SalePrice))
```

```
cor((housing_data$Flr1SF), log(housing_data$SalePrice))
cor(log(housing_data$Flr1SF), log(housing_data$SalePrice)) # Maximum

#Flr2SF
cor((housing_data$Flr2SF), (housing_data$SalePrice)) # Maximum
cor(log(housing_data$Flr2SF), (housing_data$SalePrice))
cor((housing_data$Flr2SF), log(housing_data$SalePrice))
cor(log(housing_data$Flr2SF), log(housing_data$SalePrice))

#TotalSF
cor((housing_data$TotalSF), (housing_data$SalePrice))
cor(log(housing_data$TotalSF), (housing_data$SalePrice))
cor((housing_data$TotalSF), log(housing_data$SalePrice))
cor(log(housing_data$TotalSF), log(housing_data$SalePrice)) # Maximum
```

Plotting variables with scatter plots.

```
plot(log(housing_data$LotArea), log(housing_data$SalePrice),
     main = "Log Log Area vs. Log Sale Price",
     ylab = "Log Sale Price",
     xlab = "Log Lot Area")
plot(housing_data$YearBuilt, log(housing_data$SalePrice),
     main = "Year Built vs. Log Sale Price",
     ylab = "Log Sale Price",
     xlab = "Year Built")
plot(housing_data$GarageArea, log(housing_data$SalePrice),
     main = "Garage Area vs. Log Sale Price",
     ylab = "Log Sale Price",
     xlab = "Garage Area")
plot(log(housing_data$Flr1SF), log(housing_data$SalePrice),
     main = "Log First Floor Square Footage vs. Log Sale Price",
     ylab = "Log Sale Price",
     xlab = "Log Total Square Footage")
plot((housing_data$Flr2SF), log(housing_data$SalePrice),
     main = "Second Floor Square Footage vs. Log Sale Price",
     ylab = "Log Sale Price",
     xlab = "Second Floor Square Footage")
plot(log(housing_data$TotalSF), log(housing_data$SalePrice),
     main = "Log Total Square Footage vs. Log Sale Price",
     ylab = "Log Sale Price",
     xlab = "Log Total Square Footage")
```

Plotting variables with box plots.

```
boxplot(log(housing_data$SalePrice) ~ housing_data$OverallQual,
        main = "Overall Quality vs. Log Sale Price",
        ylab = "Log Sale Price",
        xlab = "Overall Quality")
boxplot(log(housing_data$SalePrice) ~ housing_data$OverallCond,
        main = "Overall Condition vs. Log Sale Price",
        ylab = "Log Sale Price",
        xlab = "Overall Condition")
boxplot(log(housing_data$SalePrice) ~ housing_data$Fireplaces,
        main = "Number of Fireplaces vs. Log Sale Price",
        ylab = "Log Sale Price",
```

```
        xlab = "Fireplaces")
boxplot(log(housing_data$SalePrice) ~ housing_data$YrSold,
        main = "Year Sold vs. Log Sale Price",
        ylab = "Log Sale Price",
        xlab = "Year Sold")
boxplot(log(housing_data$SalePrice) ~ housing_data$CentralAir,
        main = "Central Air Presence vs. Log Sale Price",
        ylab = "Log Sale Price",
        xlab = "Central Air Present")
boxplot(log(housing_data$SalePrice) ~ housing_data$Multistory,
        main = "Multistory vs. Log Sale Price",
        ylab = "Log Sale Price",
        xlab = "Multistory")
```

Training the forward selection model.

```
fws_base <- lm(log(SalePrice) ~ 1, data = housing_data)
fws_full <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                 YearBuilt + YrSold + CentralAir + Fireplaces + GarageArea +
                 log(Flr1SF) + Flr2SF + log(TotalSF) + Multistory + log(TotalSF)*Multistory,
               data = housing_data)

fws_model <- step(fws_base, direction = 'forward', scope = formula(fws_full), trace = 0, steps = 7)
```

Training the best subset model.

```
library(olsrr)
full_model <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                 YearBuilt + YrSold + CentralAir + Fireplaces + GarageArea +
                 log(Flr1SF) + Flr2SF + log(TotalSF) + Multistory + log(TotalSF)*Multistory,
                 data = housing_data)

# From the output of this function, we can determine which combination of n variables
# makes the best model.
ols_step_best_subset(full_model, metric = "adjr")

# In the case of seven predictor variables, we have these as the best explanatory variables.
# Note that we must exclude Multistory and log(TotalSF), as these are included automatically
# in the presence of their interaction term.
best_subset_model <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +
                          Fireplaces + GarageArea + log(TotalSF)*Multistory
                        - Multistory - log(TotalSF), data = housing_data)
```

Training the backward selection model.

```
# Start with the full model.
bws_model_1_0 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + YrSold + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_1_0_adjr2 <- summary(bws_model_0)$adj.r.squared

# Train all models leaving out one predictor variable.
bws_model_1_1 <- lm(log(SalePrice) ~ OverallQual + OverallCond +
                      YearBuilt + YrSold + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
# Calclate the difference in adjusted R^2 values.
```

```r
# We will end up removing the predictor whose difference with the main model's adj.
# R^2 is smallest because we want the model to remain predicatively accurate.
bws_model_1_1_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_1)$adj.r.squared

bws_model_1_2 <- lm(log(SalePrice) ~ log(LotArea) + OverallCond +
                      YearBuilt + YrSold + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_1_2_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_2)$adj.r.squared

bws_model_1_3 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual +
                      YearBuilt + YrSold + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_1_3_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_3)$adj.r.squared

bws_model_1_4 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YrSold + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_1_4_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_4)$adj.r.squared

bws_model_1_5 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_1_5_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_5)$adj.r.squared

bws_model_1_6 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + YrSold + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_1_6_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_6)$adj.r.squared

bws_model_1_7 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + YrSold + CentralAir + GarageArea + Multistory,
                    data = housing_data)
bws_model_1_7_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_7)$adj.r.squared

bws_model_1_8 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + YrSold + CentralAir + Fireplaces + Multistory,
                    data = housing_data)
bws_model_1_8_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_8)$adj.r.squared

bws_model_1_9 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + YrSold + CentralAir + Fireplaces + GarageArea,
                    data = housing_data)
bws_model_1_9_adjr2_diff <- bws_model_1_0_adjr2 - summary(bws_model_1_9)$adj.r.squared

bws_model_1_1_adjr2_diff
bws_model_1_2_adjr2_diff
bws_model_1_3_adjr2_diff
bws_model_1_4_adjr2_diff
bws_model_1_5_adjr2_diff # Minimum value; remove YrSold from the model.
bws_model_1_6_adjr2_diff
bws_model_1_7_adjr2_diff
bws_model_1_8_adjr2_diff
bws_model_1_9_adjr2_diff
```

```r
# The process repeats itself.
bws_model_2_0_adjr2 <- summary(bws_model_1_5)$adj.r.squared

bws_model_2_1 <- lm(log(SalePrice) ~ OverallQual + OverallCond +
                      YearBuilt + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_2_1_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_1)$adj.r.squared

bws_model_2_2 <- lm(log(SalePrice) ~ log(LotArea) + OverallCond +
                      YearBuilt + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_2_2_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_2)$adj.r.squared

bws_model_2_3 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual +
                      YearBuilt + CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_2_3_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_3)$adj.r.squared

bws_model_2_4 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      CentralAir + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_2_4_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_4)$adj.r.squared

bws_model_2_5 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_2_5_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_5)$adj.r.squared

bws_model_2_6 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + CentralAir + GarageArea + Multistory,
                    data = housing_data)
bws_model_2_6_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_6)$adj.r.squared

bws_model_2_7 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + CentralAir + Fireplaces + Multistory,
                    data = housing_data)
bws_model_2_7_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_7)$adj.r.squared

bws_model_2_8 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + CentralAir + Fireplaces + GarageArea,
                    data = housing_data)
bws_model_2_8_adjr2_diff <- bws_model_2_0_adjr2 - summary(bws_model_2_8)$adj.r.squared

bws_model_2_1_adjr2_diff
bws_model_2_2_adjr2_diff
bws_model_2_3_adjr2_diff
bws_model_2_4_adjr2_diff
bws_model_2_5_adjr2_diff # Minimum value; remove CentralAir from the model.
bws_model_2_6_adjr2_diff
bws_model_2_7_adjr2_diff
bws_model_2_8_adjr2_diff

bws_model_3_0_adjr2 <- summary(bws_model_2_5)$adj.r.squared
```

```
bws_model_3_1 <- lm(log(SalePrice) ~ OverallQual + OverallCond +
                      YearBuilt + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_3_1_adjr2_diff <- bws_model_3_0_adjr2 - summary(bws_model_3_1)$adj.r.squared

bws_model_3_2 <- lm(log(SalePrice) ~ log(LotArea) + OverallCond +
                      YearBuilt + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_3_2_adjr2_diff <- bws_model_3_1_adjr2 - summary(bws_model_3_2)$adj.r.squared

bws_model_3_3 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual +
                      YearBuilt + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_3_3_adjr2_diff <- bws_model_3_2_adjr2 - summary(bws_model_3_3)$adj.r.squared

bws_model_3_4 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      Fireplaces + GarageArea + Multistory,
                    data = housing_data)
bws_model_3_4_adjr2_diff <- bws_model_3_0_adjr2 - summary(bws_model_3_4)$adj.r.squared

bws_model_3_5 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + GarageArea + Multistory,
                    data = housing_data)
bws_model_3_5_adjr2_diff <- bws_model_3_0_adjr2 - summary(bws_model_3_5)$adj.r.squared

bws_model_3_6 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + Fireplaces + Multistory,
                    data = housing_data)
bws_model_3_6_adjr2_diff <- bws_model_3_0_adjr2 - summary(bws_model_3_6)$adj.r.squared

bws_model_3_7 <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + Fireplaces + GarageArea,
                    data = housing_data)
bws_model_3_7_adjr2_diff <- bws_model_3_0_adjr2 - summary(bws_model_3_7)$adj.r.squared

# Here, all models show a decrease in adjusted R^2, so stop the backward selection.
bws_model_3_1_adjr2_diff
bws_model_3_2_adjr2_diff
bws_model_3_3_adjr2_diff
bws_model_3_4_adjr2_diff
bws_model_3_5_adjr2_diff
bws_model_3_6_adjr2_diff
bws_model_3_7_adjr2_diff

# Training the backward selection model.
bws_model <- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond +
                      YearBuilt + Fireplaces + GarageArea + Multistory,
                    data = housing_data)
```

Residual analysis plots and figure calculations.

```
create_plots <- function(model, method, data) {
  # Calculate the Standardized Residuals
  residuals <- model$residuals
```

```
  standardizedResiduals <- residuals/summary(model)$sigma

  # QQ Plot
  qqnorm(standardizedResiduals, main = paste(method, "Normal Q-Q Plot"))
  qqline(standardizedResiduals)

  # Plot Log Sale Price vs. Standardized Residuals
  plot(log(data$SalePrice), standardizedResiduals,
       xlab = "Log Sale Price",
       ylab = "Standardized Residuals",
       main =  paste(method, "Log Sale Price vs. Standardized Residuals"))

  abline(a = -2, b = 0, lty = 2, col = "orange")
  abline(a = 2, b = 0, lty = 2, col = "orange")
  abline(a = -3, b = 0, col = "red")
  abline(a = 3, b = 0, col = "red")

  # Plot Index vs. Standardized Residuals
  plot(standardizedResiduals,
       ylab = "Standardized Residuals",
       main = paste(method, "Index vs. Standardized Residuals"))
}

# Calculate the proportion of std. residuals whose absolute value is greater than 3.
ninty_fifth_percentile <- function(model, data) {
  residuals <- model$residuals
  standardizedResiduals <- residuals/summary(model)$sigma
  num_obs_above_3 <- sum(abs(standardizedResiduals) > 3)
  return((num_obs_above_3/nrow(data)) * 100)
}

create_plots(fws_model, "FWS", housing_data)
create_plots(best_subset_model, "BSS", housing_data)
create_plots(bws_model, "BWS", housing_data)

ninty_fifth_percentile(fws_model, housing_data)
ninty_fifth_percentile(best_subset_model, housing_data)
ninty_fifth_percentile(bws_model, housing_data)
```