Housing Data

Felipe Rodriguez

2023-02-12

Explain any transformations or modifications you made to the dataset

```
library(readxl)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(purrr)
library(lm.beta)
library(ggplot2)
```

```
setwd('/Users/feliperodriguez/OneDrive - Bellevue University/Github/dsc520/')

housing <- read_excel('/Users/feliperodriguez/OneDrive - Bellevue University/Github/dsc520/data/week-7-h
```

Create two variables; one that will contain the variables Sale Price and Square Foot of Lot (same variables used from previous assignment on simple regression) and one that will contain Sale Price and several additional predictors of your choice. Explain the basis for your additional predictor selections.

```
head(housing)
```

```
## # A tibble: 6 x 24
##   `Sale Date`         `Sale Price` sale_~1 sale_~2 sale_~3 sitet~4 addr_~5  zip5
##   <dttm>                     <dbl>   <dbl>   <dbl> <chr>   <chr>   <chr>   <dbl>
## 1 2006-01-03 00:00:00       698000       1       3 <NA>    R1      17021 ~ 98052
## 2 2006-01-03 00:00:00       649990       1       3 <NA>    R1      11927 ~ 98052
## 3 2006-01-03 00:00:00       572500       1       3 <NA>    R1      13315 ~ 98052
```

```
## 4 2006-01-03 00:00:00       420000       1       3 <NA>    R1      3303 1~ 98052
## 5 2006-01-03 00:00:00       369900       1       3 15      R1      16126 ~ 98052
## 6 2006-01-03 00:00:00       184667       1      15 18 51   R1      8101 2~ 98053
## # ... with 16 more variables: ctyname <chr>, postalctyn <chr>, lon <dbl>,
## #   lat <dbl>, building_grade <dbl>, square_feet_total_living <dbl>,
## #   bedrooms <dbl>, bath_full_count <dbl>, bath_half_count <dbl>,
## #   bath_3qtr_count <dbl>, year_built <dbl>, year_renovated <dbl>,
## #   current_zoning <chr>, sq_ft_lot <dbl>, prop_type <chr>, present_use <dbl>,
## #   and abbreviated variable names 1: sale_reason, 2: sale_instrument,
## #   3: sale_warning, 4: sitetype, 5: addr_full
```

```
sale_price <- housing$`Sale Price`
sqft_lot <- housing$sq_ft_lot
sale_price_lm <- lm(`Sale Price` ~ year_built + sq_ft_lot, data=housing)
price_predict_df <- data.frame(`Sale Price` = predict(sale_price_lm, housing), year_built=housing$year_
```

I selected year built and square feet lot. I felt that these predictors would influence the outcome os sale price more than others would.

Execute a summary() function on two variables defined in the previous step to compare the model results. What are the R2 and Adjusted R2 statistics? Explain what these results tell you about the overall model. Did the inclusion of the additional predictors help explain any large variations found in Sale Price?

```
summary_sale_price_lm <-summary(sale_price_lm)
summary_sale_price_lm
```

```
##
## Call:
## lm(formula = `Sale Price` ~ year_built + sq_ft_lot, data = housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2441124  -166193   -48805    74921  3634286
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.170e+07  3.991e+05  -29.32   <2e-16 ***
## year_built   6.191e+03  2.002e+02   30.93   <2e-16 ***
## sq_ft_lot    1.104e+00  6.054e-02   18.23   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 387400 on 12862 degrees of freedom
## Multiple R-squared:  0.08259,    Adjusted R-squared:  0.08245
## F-statistic:   579 on 2 and 12862 DF,  p-value: < 2.2e-16
```

The R Squared value is .08259 and Adjusted R Squared is 0.08245. Our R squared values are low for this model, which means that we are seeing a weak relationship within the variables selected.

Considering the parameters of the multiple regression model you have created. What are the standardized betas for each parameter and what do the values indicate?

```
lm.beta(sale_price_lm)
```

```
##
## Call:
## lm(formula = `Sale Price` ~ year_built + sq_ft_lot, data = housing)
##
## Standardized Coefficients::
## (Intercept)  year_built   sq_ft_lot
##          NA   0.2636342   0.1553802
```

The standardized beta compares the strength of the variable in relation to the dependent variable. For these two fields, year built has more of an effect on sale price than square foot lot.

Calculate the confidence intervals for the parameters in your model and explain what the results indicate.

```
t.test(housing$year_built)
```

```
##
##  One Sample t-test
##
## data:  housing$year_built
## t = 13127, df = 12864, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  1992.705 1993.300
## sample estimates:
## mean of x
##  1993.003
```

```
t.test(housing$sq_ft_lot)
```

```
##
##  One Sample t-test
##
## data:  housing$sq_ft_lot
## t = 44.284, df = 12864, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  21244.67 23212.47
## sample estimates:
## mean of x
##  22228.57
```

The 95 % confidence intervals for the parameter year built is 1992 and 1993 and for square feet lot it is 21244 and 23212. This tells us that 95 percent of values will fall between those parameters.

Assess the improvement of the new model compared to your original model (simple regression model) by testing whether this change is significant by performing an analysis of variance.

```
priceAnova <- aov(`Sale Price` ~ year_built + sq_ft_lot, data=housing)
priceAnova
```

```
## Call:
##    aov(formula = `Sale Price` ~ year_built + sq_ft_lot, data = housing)
##
## Terms:
##                    year_built    sq_ft_lot    Residuals
## Sum of Squares  1.238781e+14 4.986216e+13 1.929833e+15
## Deg. of Freedom             1            1        12862
##
## Residual standard error: 387351.9
## Estimated effects may be unbalanced
```

Perform casewise diagnostics to identify outliers and/or influential cases, storing each function's output in a dataframe assigned to a unique variable name.

```
casewise_diagnostics <- housing %>% mutate(z_score_price = ((`Sale Price` - mean(`Sale Price`))/sd(`Sal
casewise_diagnostics <- arrange(casewise_diagnostics, desc(z_score_price))
casewise_diagnostics <- select(casewise_diagnostics, `Sale Price`, z_score_price)
head(casewise_diagnostics)
```

```
## # A tibble: 6 x 2
##   `Sale Price` z_score_price
##          <dbl>         <dbl>
## 1      4400000          9.25
## 2      4400000          9.25
## 3      4380542          9.20
## 4      4380542          9.20
## 5      4380542          9.20
## 6      4380542          9.20
```

Calculate the standardized residuals using the appropriate command, specifying those that are +-2, storing the results of large residuals in a variable you create.

```
r <- rstandard(sale_price_lm, res = +- 2)
residuals_housing <- housing$`Sale Price` - price_predict_df$Sale.Price
```

Use the appropriate function to show the sum of large residuals.

```
sse <- sum(residuals_housing^2)
```

Which specific variables have large residuals (only cases that evaluate as TRUE)?

```
housing$residuals <- residuals(sale_price_lm)
```

Investigate further by calculating the leverage, cooks distance, and covariance rations. Comment on all cases that are problematics.

```
hats <- as.data.frame(hatvalues(sale_price_lm))
head(hats)
```

```
##   hatvalues(sale_price_lm)
## 1             1.069596e-04
## 2             1.248970e-04
## 3             9.379567e-05
## 4             2.554228e-04
## 5             1.323194e-04
## 6             1.177130e-04
```

```
cookd <- as.data.frame(cooks.distance(sale_price_lm))
head(cookd)
```

```
##   cooks.distance(sale_price_lm)
## 1                  1.308965e-08
## 2                  1.472082e-06
## 3                  2.680544e-07
## 4                  2.944078e-06
## 5                  1.107942e-05
## 6                  7.454685e-05
```

```
covariance <- vcov(sale_price_lm)
head(covariance)
```

```
##               (Intercept)    year_built     sq_ft_lot
## (Intercept)  1.592890e+11 -7.988104e+07 -3.339641e+03
## year_built  -7.988104e+07  4.006252e+04  1.634806e+00
## sq_ft_lot   -3.339641e+03  1.634806e+00  3.665054e-03
```

Perform the necessary calculations to assess the assumption of independence and state if the condition is met or not.

```
chi_squared <- chisq.test(housing$`Sale Price`, housing$year_built)
```

```
## Warning in chisq.test(housing$‘Sale Price‘, housing$year_built): Chi-squared
## approximation may be incorrect
```

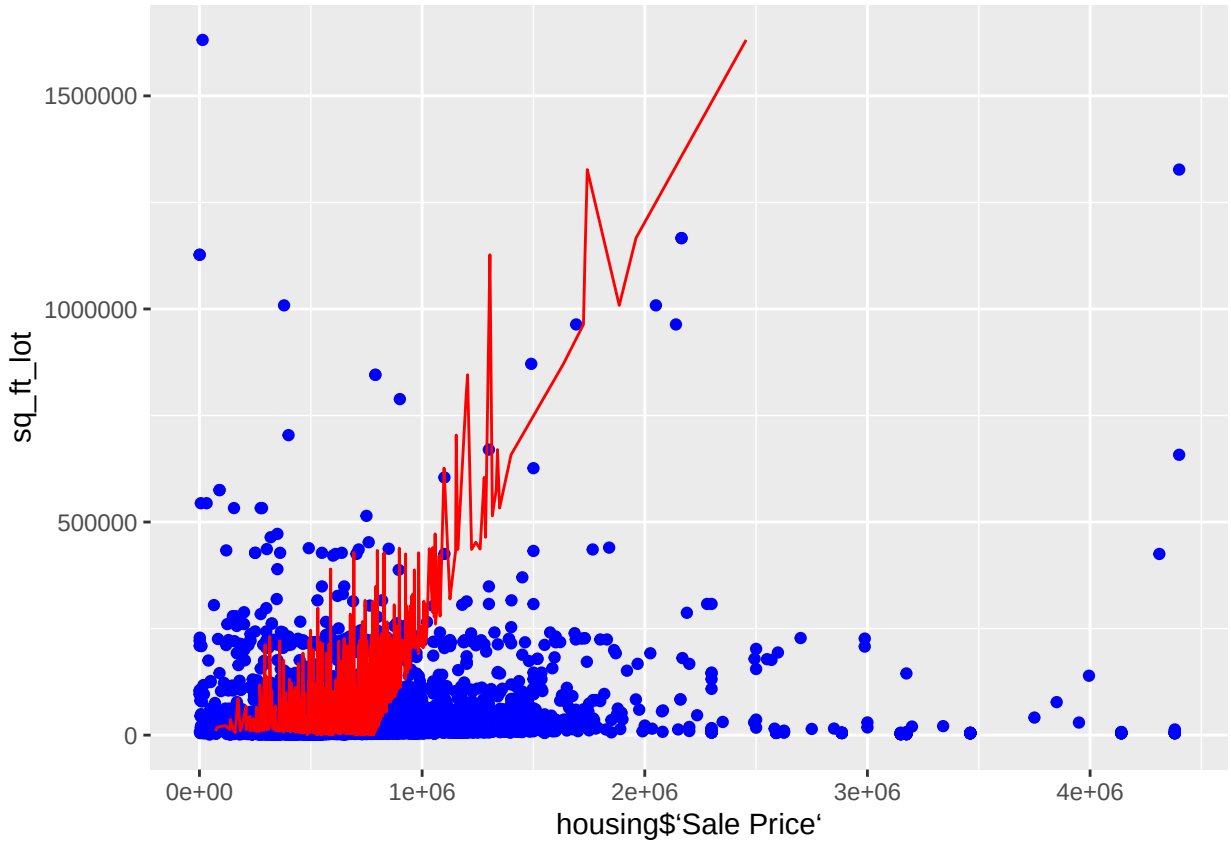
```
chi_squared
```

```
##
##  Pearson’s Chi-squared test
##
## data:  housing$‘Sale Price‘ and housing$year_built
## X-squared = 422223, df = 433944, p-value = 1
```

Perform the necessary calculations to assess the assumption of no multicollinearity and state if the condition is met or not.

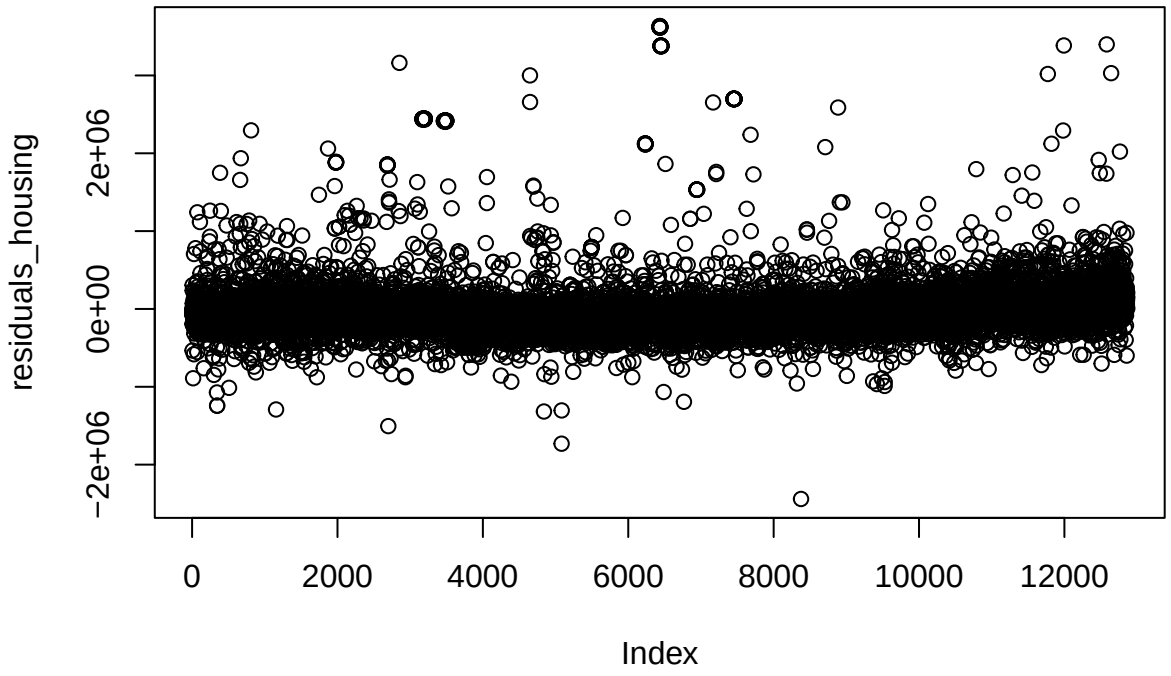
```
ggplot(data = housing, aes(y = sq_ft_lot, x = housing$`Sale Price`)) +
  geom_point(color='blue') +
  geom_line(color='red',data = price_predict_df, aes(y= sq_ft_lot, x= price_predict_df$Sale.Price))
```

```
## Warning: Use of ‘‘ housing$‘Sale Price‘ ‘‘ is discouraged.
## i Use ‘Sale Price‘ instead.
```

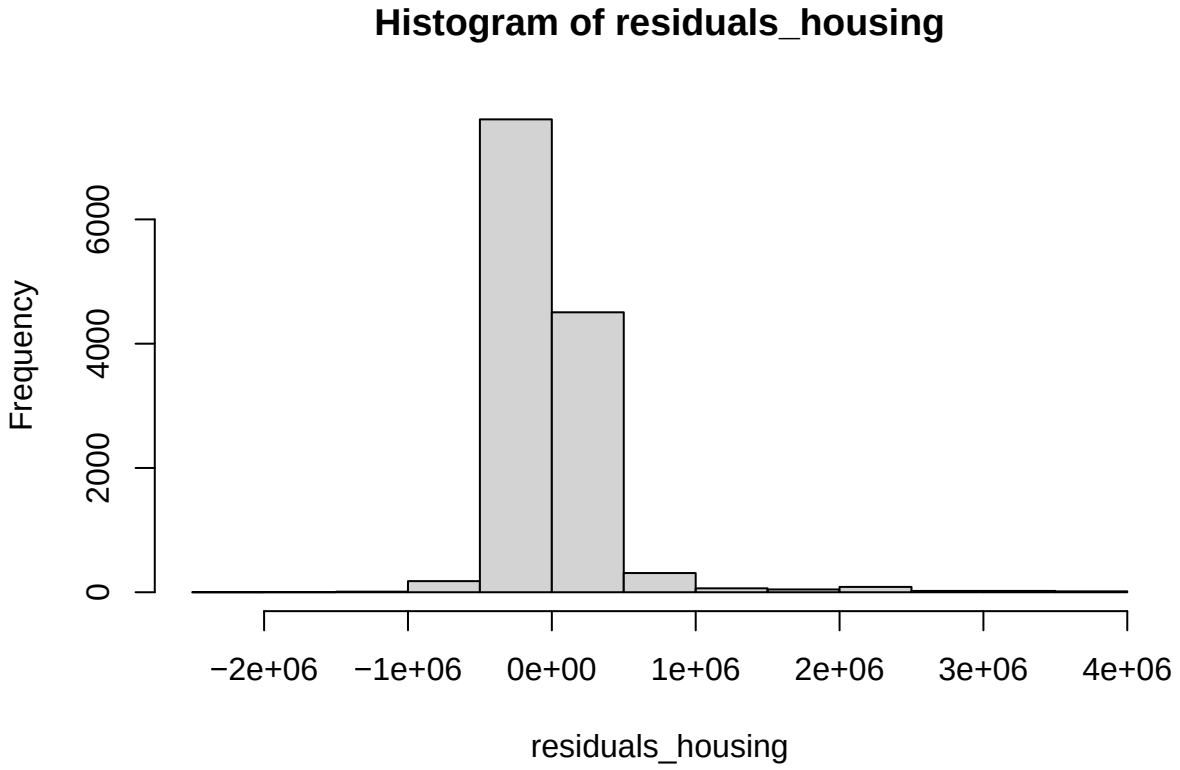


Visually check the assumptions related to the residuals using the plot() and hist() functions. Summarize what each graph is informing you of and if any anomalies are present.

```
plot(residuals_housing)
```




```
hist(residuals_housing)
```



Overall, is this regression model unbiased? If an unbiased regression model, what does this tell us about the sample vs. the entire population model?

I believe this model is unbiased. The plots show that the residuals lie in a central locations, and if we have random samples, those random samples will most likley fall into the model.