Week8_AdamsKimberly

2022-07-30

Assignment 6

```
## Set the working directory to the root of your DSC 520 directory
setwd("/Users/kimberlyadams/Documents/GitHub/DSC520-Statistics-and-R")

## Load the `data/r4ds/heights.csv` to
heights_df <- read.csv("data/r4ds/heights.csv")

## Load the ggplot2 library
library(ggplot2)

## Fit a linear model using the `age` variable as the predictor and `earn` as the outcome
age_lm <-  lm(earn ~ age, data = heights_df)

## View the summary of your model using `summary()`
summary(age_lm)
```

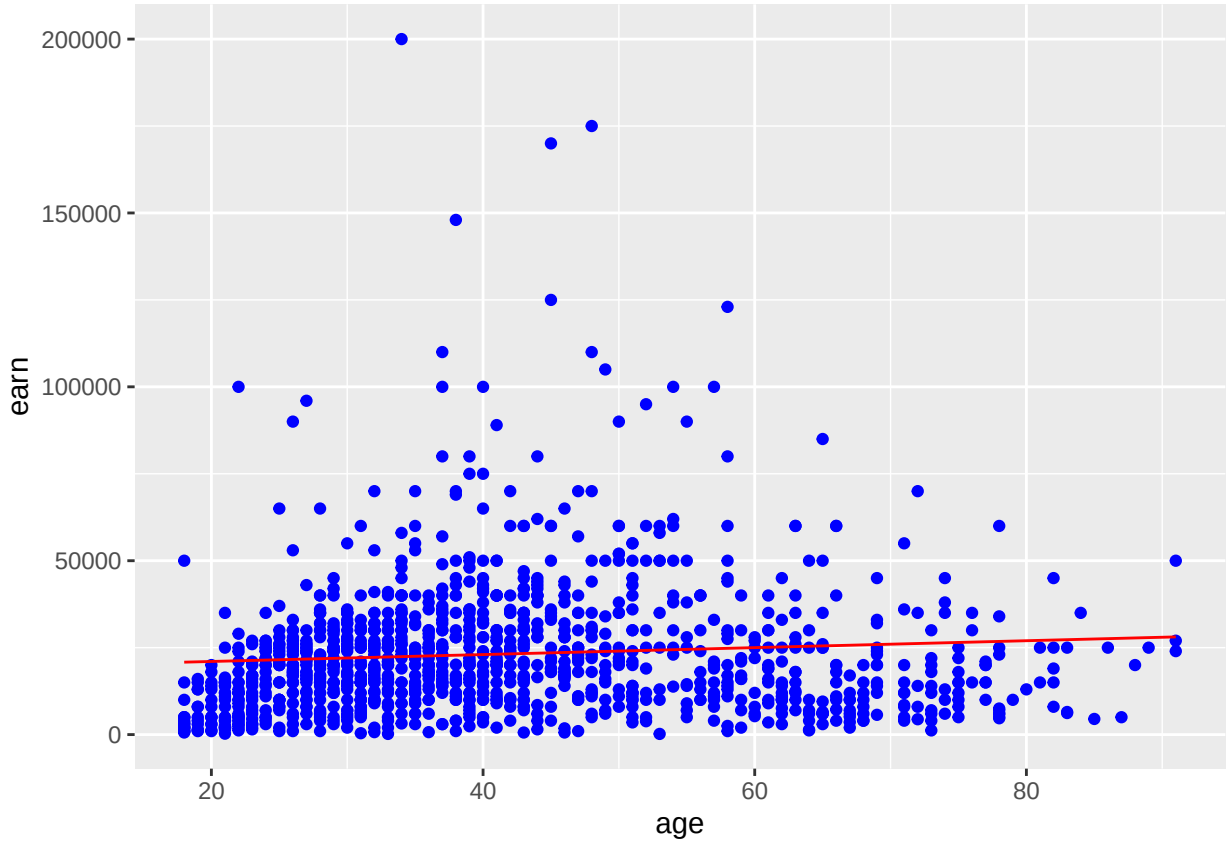
```
##
## Call:
## lm(formula = earn ~ age, data = heights_df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -25098 -12622  -3667   6883 177579
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19041.53    1571.26  12.119  < 2e-16 ***
## age            99.41      35.46   2.804  0.00514 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19420 on 1190 degrees of freedom
## Multiple R-squared:  0.006561,   Adjusted R-squared:  0.005727
## F-statistic:  7.86 on 1 and 1190 DF,  p-value: 0.005137
```

```
## Creating predictions using `predict()`
age_predict_df <- data.frame(earn = predict(age_lm, heights_df), age = c(heights_df$age))
head(age_predict_df)
```

```
##       earn age
```

```
## 1 23514.79  45
## 2 24807.06  58
## 3 21924.29  29
## 4 28087.45  91
## 5 22918.35  39
## 6 21626.08  26
```

```
## Plot the predictions against the original data
ggplot(data = heights_df, aes(y = earn, x = age)) +
  geom_point(color='blue') +
  geom_line(color='red',data = age_predict_df, aes(y = earn, x = age))
```




```
mean_earn <- mean(heights_df$earn)
## Corrected Sum of Squares Total
sst <- sum((mean_earn - heights_df$earn)^2)
## Corrected Sum of Squares for Model
ssm <- sum((mean_earn - age_predict_df$earn)^2)
## Residuals
residuals <- heights_df$earn - age_predict_df$earn
## Sum of Squares for Error
sse <- sum(residuals^2)
## R Squared R^2 = SSM\SST
r_squared <- ssm / sst

## Number of observations
```

```
n <- nrow(heights_df)
## Number of regression parameters
p <- 2
## Corrected Degrees of Freedom for Model (p-1)
dfm <- p - 1
## Degrees of Freedom for Error (n-p)
dfe <- n - p
## Corrected Degrees of Freedom Total:   DFT = n - 1
dft <- n - 1

## Mean of Squares for Model:   MSM = SSM / DFM
msm <- ssm / dfm
## Mean of Squares for Error:   MSE = SSE / DFE
mse <- sse / dfe
## Mean of Squares Total:   MST = SST / DFT
mst <- sst / dft
## F Statistic F = MSM/MSE
f_score <- msm / mse

## Adjusted R Squared R2 = 1 - (1 - R2)(n - 1) / (n - p)
adjusted_r_squared <- 1 - ((1 - r_squared) * (n - 1)) / (n - p)

## Calculate the p-value from the F distribution
p_value <- pf(f_score, dfm, dft, lower.tail=F)
```

Assignment 7

```
## Set the working directory to the root of your DSC 520 directory
setwd("/Users/kimberlyadams/Documents/GitHub/DSC520-Statistics-and-R")

## Load the `data/r4ds/heights.csv` to
heights_df <- read.csv("data/r4ds/heights.csv")
colnames(heights_df)
```

```
## [1] "earn"   "height" "sex"    "ed"     "age"    "race"
```

```
# Fit a linear model
earn_lm <-  lm(earn ~ height + sex + ed + age + race, data = heights_df)

# View the summary of your model
summary(earn_lm)
```

```
##
## Call:
## lm(formula = earn ~ height + sex + ed + age + race, data = heights_df)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -39423  -9827  -2208   6157 158723
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -41478.4    12409.4  -3.342 0.000856 ***
## height          202.5      185.6   1.091 0.275420
## sexmale       10325.6     1424.5   7.249 7.57e-13 ***
## ed             2768.4      209.9  13.190  < 2e-16 ***
## age             178.3       32.2   5.537 3.78e-08 ***
## racehispanic  -1414.3     2685.2  -0.527 0.598507
## raceother       371.0     3837.0   0.097 0.922983
## racewhite      2432.5     1723.9   1.411 0.158489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 17250 on 1184 degrees of freedom
## Multiple R-squared:  0.2199, Adjusted R-squared:  0.2153
## F-statistic: 47.68 on 7 and 1184 DF,  p-value: < 2.2e-16
```

```
predicted_df <- data.frame(
  earn = predict(earn_lm, heights_df),
  ed = c(heights_df$ed), race = c(heights_df$race), height = c(heights_df$height),
  age = c(heights_df$age), sex = c(heights_df$sex))
head(predicted_df)
```

```
##       earn ed  race   height age    sex
## 1 38666.11 16 white 74.42444  45   male
## 2 28859.09 16 white 65.53754  58 female
## 3 23301.90 16 white 63.62920  29 female
## 4 32189.84 16 other 63.10856  91 female
## 5 27807.39 17 white 63.40248  39 female
## 6 20154.60 15 white 64.39951  26 female
```

```
## Compute deviation (i.e. residuals)
mean_earn <- mean(heights_df$earn)
## Corrected Sum of Squares Total
sst <- sum((mean_earn - heights_df$earn)^2)
## Corrected Sum of Squares for Model
ssm <- sum((mean_earn - predicted_df$earn)^2)
## Residuals
residuals <- heights_df$earn - predicted_df$earn
## Sum of Squares for Error
sse <- sum(residuals^2)
## R Squared
r_squared <- ssm / sst

## Number of observations
n <- nrow(heights_df)
## Number of regression paramaters
p <- 8
## Corrected Degrees of Freedom for Model
dfm <- p - 1
```

```r
## Degrees of Freedom for Error
dfe <- n - p
## Corrected Degrees of Freedom Total:   DFT = n - 1
dft <- n - 1

## Mean of Squares for Model:   MSM = SSM / DFM
msm <- ssm / dfm
## Mean of Squares for Error:   MSE = SSE / DFE
mse <- sse / dfe
## Mean of Squares Total:   MST = SST / DFT
mst <- sst / dft
## F Statistic
f_score <- msm / mse

## Adjusted R Squared R2 = 1 - (1 - R2)(n - 1) / (n - p)
adjusted_r_squared <- 1 - ((1 - r_squared) * (n - 1)) / (n - p)
adjusted_r_squared
```

```
## [1] 0.2152832
```

Housing Dataset

1) Explain any transformations or modifications you made to the dataset

- When importing the data, I replaced spaces with "." in the column names
- I did not transform the data as often the transformation causes more problems. There is a potential for transforming variables such as sale price to try to normalize the data spread from a positive skew to a more normal distribution, but I did not do this.

2) Create two models; one that will contain the variables Sale Price and Square Foot of Lot and one that will contain Sale Price and several additional predictors of your choice. Explain the basis for your additional predictor selections.

Creating linear models for:

- Price and lot size = how much am I paying for the land space?
- Price and Year Built and lot size = Are older houses cheaper and bigger lots more expensive?

```r
PriceBySqFtLot.lm = lm(Sale.Price ~ sq_ft_lot, data = housing_data)

PriceBuildLotSize.lm = lm(Sale.Price ~ sq_ft_lot + year_built, data = housing_data)
```

3) Execute a summary() function on two variables defined in the previous step to compare the model results.

```
summary(PriceBySqFtLot.lm)
```

```
##
## Call:
## lm(formula = Sale.Price ~ sq_ft_lot, data = housing_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2016064  -194842   -63293    91565  3735109
##
## Coefficients:
##                 Estimate   Std. Error t value            Pr(>|t|)
## (Intercept) 641821.40609   3799.91526  168.90 <0.0000000000000002 ***
## sq_ft_lot        0.85099      0.06217   13.69 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 401500 on 12863 degrees of freedom
## Multiple R-squared:  0.01435,    Adjusted R-squared:  0.01428
## F-statistic: 187.3 on 1 and 12863 DF,  p-value: < 0.00000000000000022
```

```
summary(PriceBuildLotSize.lm)
```

```
##
## Call:
## lm(formula = Sale.Price ~ sq_ft_lot + year_built, data = housing_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2441124  -166193   -48805    74921  3634286
##
## Coefficients:
##                     Estimate      Std. Error t value            Pr(>|t|)
## (Intercept) -11702314.87772    399110.28901  -29.32 <0.0000000000000002 ***
## sq_ft_lot           1.10362         0.06054   18.23 <0.0000000000000002 ***
## year_built       6190.92039       200.15623   30.93 <0.0000000000000002 ***
## ---
## 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: < 0.00000000000000022
```

a) What are the R2 and Adjusted R2 statistics?

Single model with just lot size has a R2 of 0.014 while the model with 2 explanatory variables has an adjusted R2 of 0.083.

b) Explain what these results tell you about the overall model.

The model is not very good at predicting the sale price as it explains less than 10% of the variation within the sale price data.

c) Did the inclusion of the additional predictors help explain any large variations found in Sale Price

The model improved 8x with the addition of the year_built variable, but is still not a great model for this data as it only explains about 8% of the data.

4) Considering the parameters of the multiple regression model you have created.

a) What are the standardized betas for each parameter and what do the values indicate?

```
##  sq_ft_lot year_built
##  0.1553802  0.2636342
```

The standardized beta value is an indication of the how the outcome changes (in units of standard deviations) if the predictor changes by 1 standard deviation. In other words, it gives us an indicator of the importance of the predictor.

The value of year_built is 0.16 and the sq_ft_lot is 0.26 This means that the square footage of the lot is a better predictor than the year the house was built as it has a bigger influence on the Sale Price indicated by the higher beta value.

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

```
cor.test(housing_data$Sale.Price, housing_data$year_built)
```

```
##
##  Pearson's product-moment correlation
##
## data:  housing_data$Sale.Price and housing_data$year_built
## t = 28.371, df = 12863, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.2263401 0.2588660
## sample estimates:
##       cor
## 0.2426713
```

```
cor.test(housing_data$Sale.Price, housing_data$sq_ft_lot)
```

```
##
##  Pearson's product-moment correlation
##
## data:  housing_data$Sale.Price and housing_data$sq_ft_lot
## t = 13.687, df = 12863, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.1027447 0.1368093
## sample estimates:
##       cor
## 0.1198122
```

The confidence interval for Sale.Price and year_built is 0.23-0.25 which gives us confidence that there is a positively linear relationship between the two variables.

The confidence interval for Sale.Price and sq_ft_lot is 0.10-0.13 which also gives us confidence that there is a positively linear relationship between the two variables.

Both variables give very small ranges of confidence interval which is good as it means our estimate is very likely correct as the correlation value is predicted to fall within those ranges 95% of the time.

6) 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.

```
anova(PriceBySqFtLot.lm, PriceBuildLotSize.lm)
```

```
## Analysis of Variance Table
##
## Model 1: Sale.Price ~ sq_ft_lot
## Model 2: Sale.Price ~ sq_ft_lot + year_built
##   Res.Df              RSS Df       Sum of Sq      F                Pr(>F)
## 1  12863 2073376756946868
## 2  12862 1929833267833791  1 143543489113076 956.69 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The F value is 956.69 and the p value is <.001 meaning that there is a significant improvement to the model by adding the year_built variable.

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

```
housing_data$Residuals <- resid(PriceBuildLotSize.lm)
housing_data$StandResiduals <- rstandard(PriceBuildLotSize.lm)
housing_data$StudentResiduals <- rstudent(PriceBuildLotSize.lm)

housing_data$Cooks <- cooks.distance(PriceBuildLotSize.lm)
housing_data$DFBeta <- dfbeta(PriceBuildLotSize.lm)
housing_data$DFFit <- dffits(PriceBuildLotSize.lm)
housing_data$leverage <- hatvalues(PriceBuildLotSize.lm)
housing_data$covarRatio <- covratio(PriceBuildLotSize.lm)
```

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

```
housing_data$LargeResidual <- housing_data$StandResiduals > 2 | housing_data$StandResiduals < -2
```

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

```
sum(housing_data$LargeResidual)
```

```
## [1] 365
```

There are 365 large residuals out of the 12865 rows of data which is equal to roughly 3% of the data.

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

```
## # A tibble: 365 x 3
##    Sale.Price sq_ft_lot StandResiduals
##         <dbl>     <dbl>          <dbl>
##  1     165000    278891          -2.30
##  2    1445000     36446           2.02
##  3    1900000     37017           3.21
##  4    1520000     19173           2.88
##  5    1588359      8752           2.24
##  6    1450000     14043           2.40
##  7    1450000     14043           3.26
##  8     200000    288367          -2.19
##  9     275000    532739          -2.78
## 10      90000    574992          -3.22
## # ... with 355 more rows
```

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

```
## # A tibble: 365 x 3
##       Cooks  leverage covarRatio
##       <dbl>     <dbl>      <dbl>
##  1 0.00331  0.00187        1.00
##  2 0.000113 0.0000831      0.999
##  3 0.000289 0.0000845      0.998
##  4 0.00147  0.000530       0.999
##  5 0.000196 0.000117       0.999
##  6 0.000385 0.000201       0.999
##  7 0.00567  0.00159        0.999
##  8 0.00313  0.00195        1.00
##  9 0.0176   0.00677        1.01
## 10 0.0268   0.00769        1.01
## # ... with 355 more rows
```

Of the properties with large residuals:

- 2 of the properties come close to a Cook's distance of 1 (0.95 and 0.89) as the highest distance is 0.14.
- 19 properties have leverage values 2 times the average leverage value of 0.0002.
- 26 properties have a CVR value greater than 1.0004 = (1 + [3(2 + 1) / 12856]) and 19 have a CVR value less than 0.9996 = (1 - [3(2 + 1) / 12856]).

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

```
car::durbinWatsonTest(PriceBuildLotSize.lm)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.6203378     0.7593244       0
##  Alternative hypothesis: rho != 0
```

The Durbin Watson test returns a value of 0.76 which is not optimal as it is less than one. This indicates that the assumption of independence is not met in this data.

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

```
library(car)
```

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

```
vif(PriceBuildLotSize.lm)
```

```
##  sq_ft_lot year_built
##   1.018539   1.018539
```

```
1 / vif(PriceBuildLotSize.lm)
```

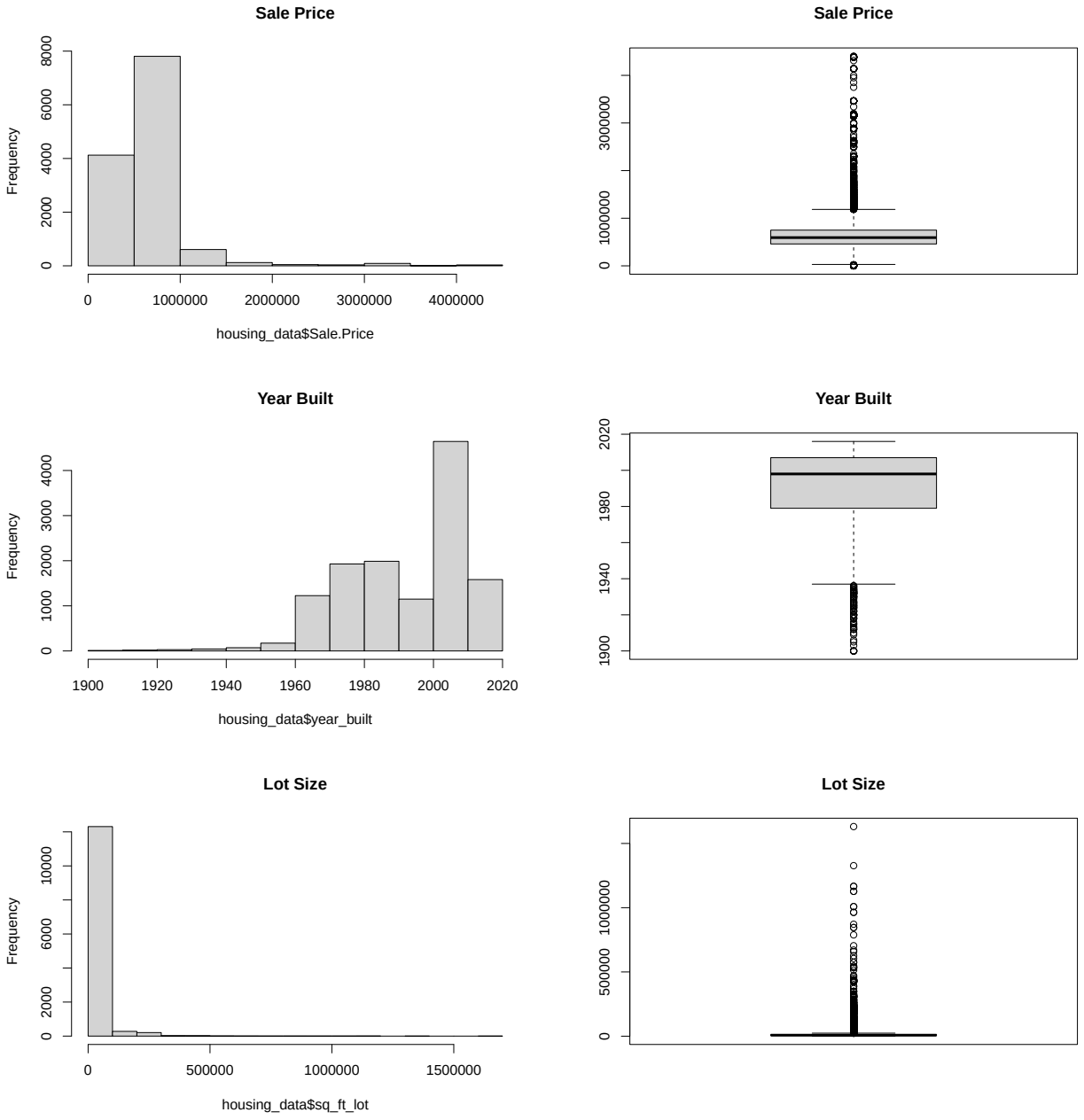
```
##  sq_ft_lot year_built
##  0.9817982  0.9817982
```

```
mean(vif(PriceBuildLotSize.lm))
```

```
## [1] 1.018539
```

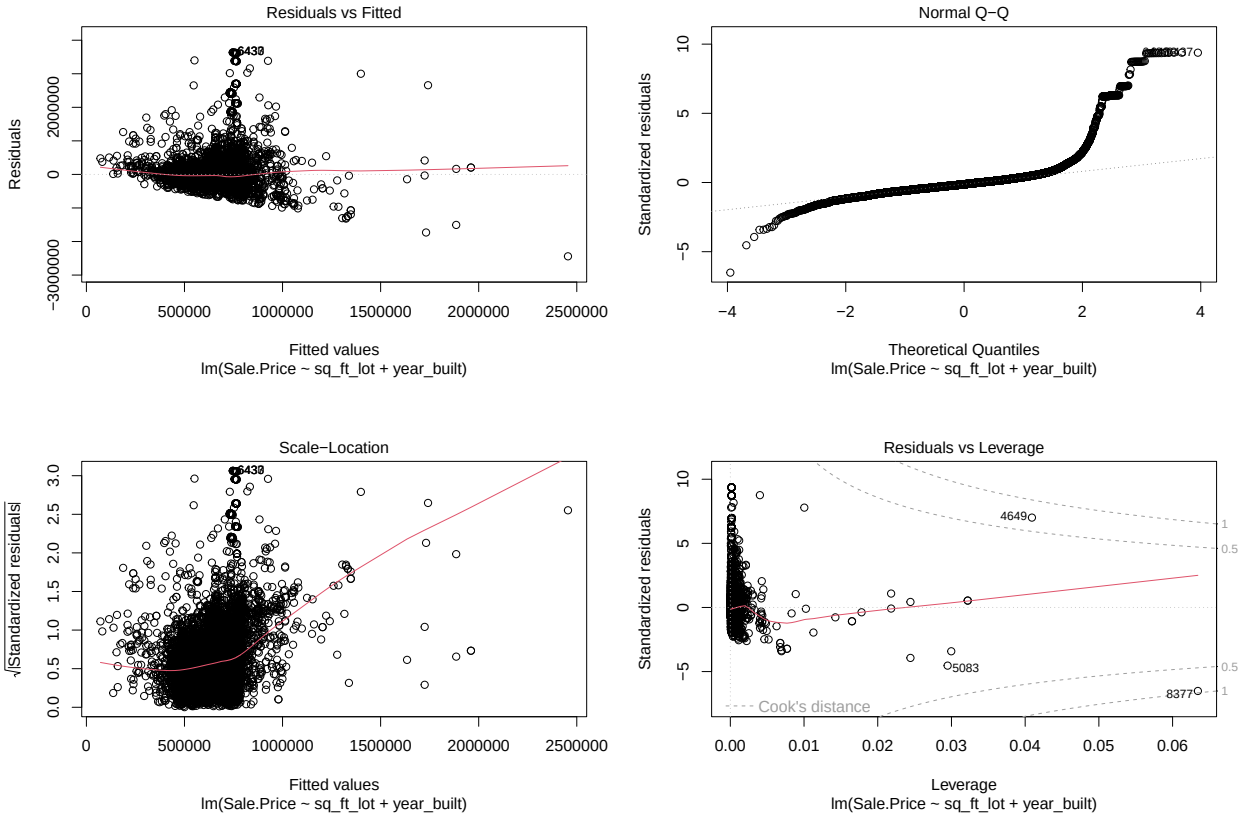
- VIF is not greater than 10 so no worries there.
- The average VIF is very slightly greater than 1 thus the regression may be every so slightly biased.
- Tolerance levels are around 0.98 so no worries there either.
- Based on these observations, we can safely conclude that there is no collinearity within the data.

14) 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.



Sale.Price and sq_ft_lot both have positively skewed distributions while year_built has a negatively skewed distribution.The sq_ft_lot has the strongest skew.

The skewing shown in the histograms is echoed in boxplots of each of the variables showing that there are many potential outliers at the tail ends of the data.



The first plot the Residuals vs Fitted, shows a not so random pattern indicating that the assumptions of heteroscedasticity, linearity and randomness have NOT been met. This reinforces what we found earlier that the assumption of independence was NOT met in this data.

The second plot (The QQ Plot) shows that the values in the lower range do have some properties of normal distribution (based on the closeness of the data to line), but as the data value increases, the normal distribution disappears and the data points very farther from the line indicating a positive skew.

In the third plot, Scale-Location, the red line is not horizontal indicating the assumption of homoscedasticity is NOT met. Also the points are bunched up in the lower x values indicating greater variance at that end of the regression.

The last Residuals vs Leverage plot show there there is one data point (#8377) that has significant leverage (compared to the other data points in this set) but very little residual meaning it may have swayed the model to fit itself. Likewise towards the top of the graph point #4649 also has a high cook's distance, but has less leverage which resulted in a higher residual value as it didn't have as much sway as point 8377. Both points are influential points. Most of the data points have very high residuals meaning that they don't fit the linear model very well, but have little leverage to pull the line towards them.

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

- The VIF value was around 1 which indicates that there is very little multicollinearity bias present.
- If we consider points 8377 and 4649 outtliers, then they are applying some bias to the model, by pulling the linear regression towards themselves.

- Because the data appears to be unbiased, that means that we can assume that it is representative of the population.