Stat 4355 Final Report

Data: US Cities Housing Market Data

Team Name: Home Sweet Home

Team Members: Orlando Malanco, Subre Moktar, Sayema Rahman, Elaine Wu

Data Description

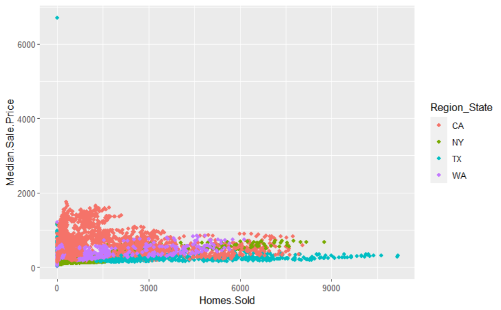
Our objective is to find out which variables contribute the most to the median housing cost and predict future observations of median sale price of houses in metro areas of 4 States Texas, California, New York, and Washington. Originally the data set we chose had around 20 predictors and 21,303 observations. The number of observations for each state was 4463 for California, 3320 for New York, 5581 for Texas, and 2701 for Washington. Based on our objective, we used median sale price as our dependent variable.

Data Cleaning and Final Dataframes

The original dataset needed to be cleaned before we could analyze it. Using the packages tidyr and dplyr, we removed the unnecessary “Metro area” predictors. Furthermore, we used subset selection to remove predictors that were deemed unnecessary by forward selection. Then we removed the $ and % characters so that we could convert the data type from character into numerical so we could apply transformations and parse through the data when required. The “days on market” column had more problems: We found that there were indices in the data set that have “NA” in this column. Also, this column did not convert to a numerical type because it contained commas in values that are over 1000 which introduced commas into the value. To handle this hindrance, we first used gsub() to rid the commas then converted the column into numerical type. After that, we removed the indices with “NA”. Now that we have this cleaned dataset, we chose to concentrate on four of the most populated states in the United States, i.e., California, Texas, New York, and Washington., so we subsetted our tidied dataset into four main data frames by these regions.

Exploratory Data Analysis

We began assessing the data by plotting various scatterplots with one of the predicting variables against median sale price, e.g., median sale price vs. homes sold, as shown below. To grasp the center and spread of the median sale price in different regions, we made boxplots of median sale prices across the chosen states. The results seem to confirm our choice of dividing the data frame by region because the scatterplots seem to fan out rather than have random scatters, and the boxplot shows that the centers of median sale price noticeably vary across these four densely populated states. The boxplot also indicates that there are possible outliers or influential points that might need special attention or transformations. Notice that there is one distinct outlier in the state of Texas with a median sale price of over $6000000!



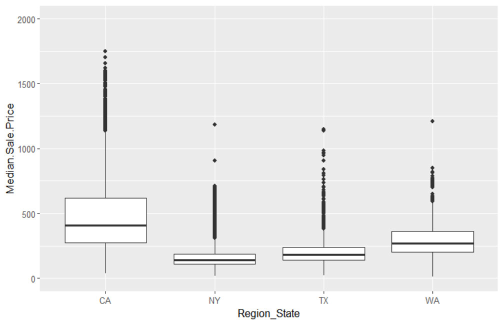
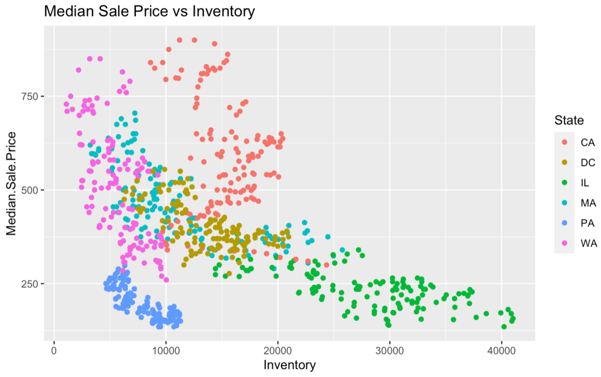


Figure 1: As shown above, there is one significant outlier in the state of Texas when looking at the number of homes sold versus the Median Sale Price

Figure 2: As shown above, you can see that the state of California has the highest Median Sale Price than the other regions

Graphs before Transformation



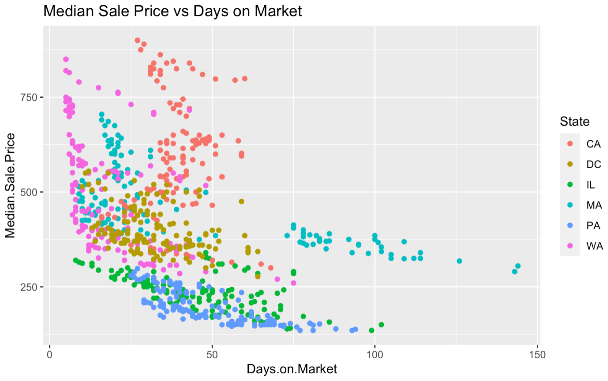
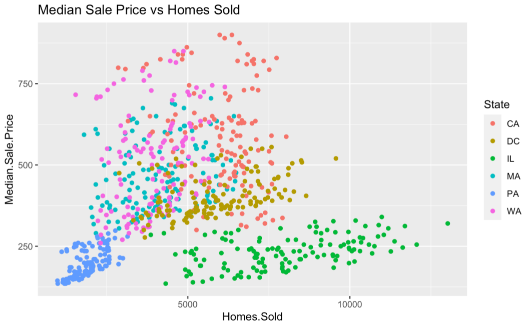


Figure 3: As shown above, there is a slight negative correlation within the graph.

Figure 4: As shown above, there is a slight negative correlation between the number of days the houses stay on the market and the Median Sale Price



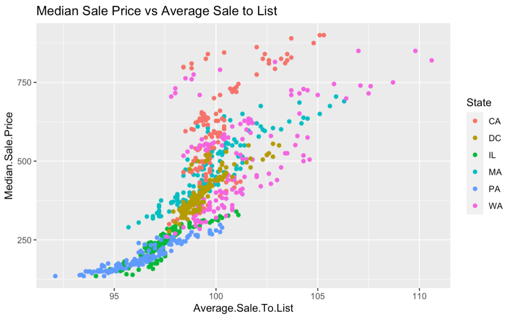
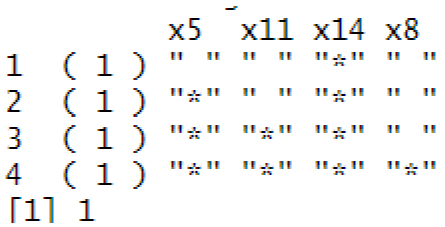


Figure 5: As shown above there is no clear correlation between the Median Sale Price and the total number of Homes Sold in each region

Figure 6: As shown in the figure above there is a clear positive correlation between the Median Sale Price and the Average Sale to List

SUBSET SELECTION:

We did a subset selection on the model to see which predictor would be less important using AIC and forward selection on the entire dataset. We wanted to first model all the metro areas first before separating them, and seeing if we could have a good linear model created from it.



```
          x5  x11 x14 x8
1  ( 1 ) " " " " "*" " "
2  ( 1 ) "*" " " "*" " "
3  ( 1 ) "*" "*" "*" " "
4  ( 1 ) "*" "*" "*" "*"
[1] 1
```

From the image above it shows the forward selection AIC happening. The last line shows the best model that was selected using this technique. This conveys that an appropriate model would just be y equaling to intercept and days on market (x14). This means using the entire metro areas we could have a good model from just using that variable. However, we are concerned with underfitting the model when just using one predictor. Thus, we decided since each state seems to have a different linear shaping, we would create a different model for each state. Since we also separated the data points we also decided not to do more subset selection since we made each region its own model, which reduces the amount of observations for each model.

Variable Selection

We chose our variables based on the results of our subset selection. The following variables are in our reduced model: homes sold, average sale to list, days on market for all the states combined

VIF (Multicollinearity)

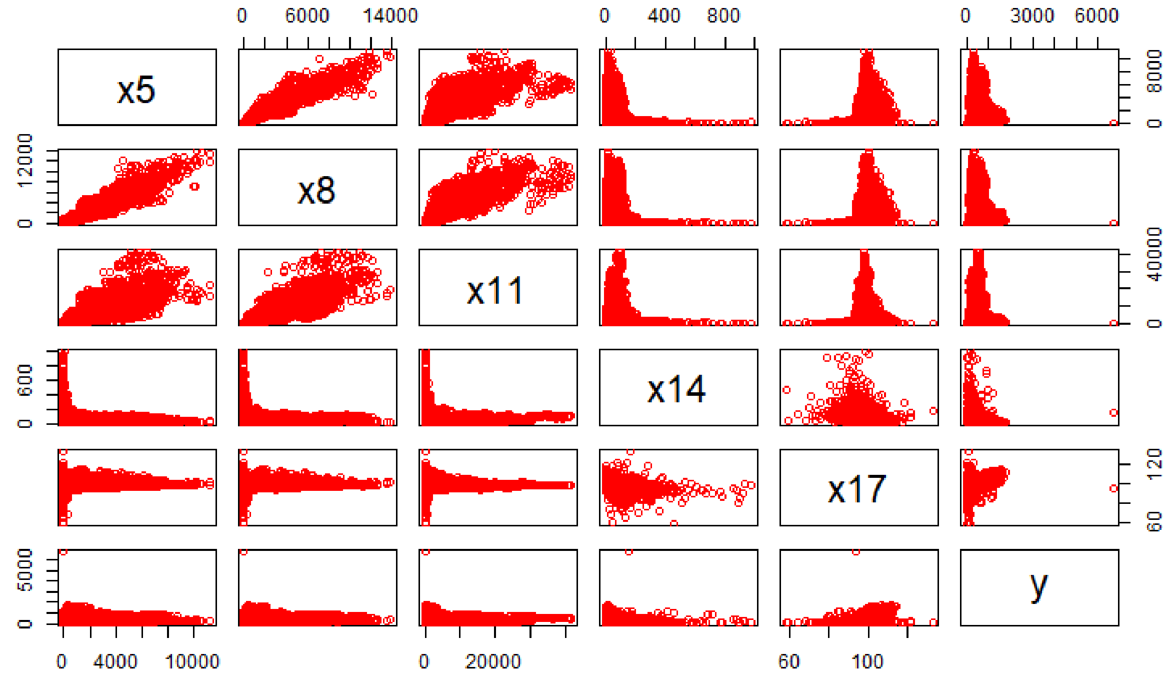
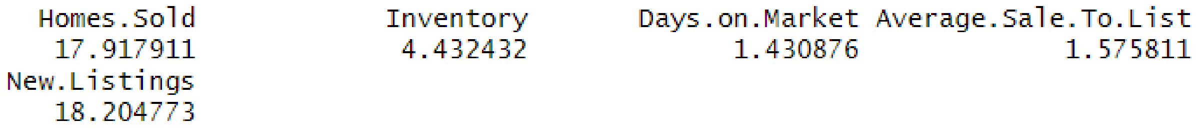


Image 3: The image above shows all the columns plotted against each other, the pairs graph.

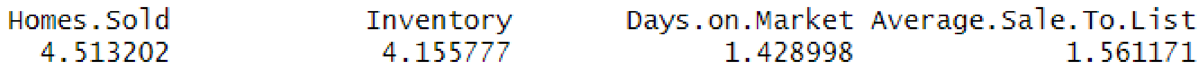
We see a possibility of collinearity between x5(homes sold), x8(new listings) and potentially x11 (Inventory). These predictors may be problematic for our model. Therefore we would need to verify whether or not these predictors have collinearity, and if they do we should deal with them accordingly. Since we created models for each state, we need to check if the VIF is high for all the models. After creating all the full models, it seemed that all of them had very similar VIF violations with values over 10 on the same attributes. We decided to remove x8, which is a new listing, and after running VIF again on models without x8 and all the values were below 10.

Washington Full Model VIF -



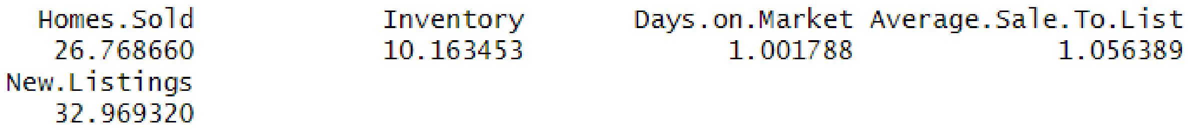
```
         Homes.Sold           Inventory      Days.on.Market Average.Sale.To.List 
          17.917911            4.432432            1.430876             1.575811 
       New.Listings 
          18.204773 
```

Washington Reduced Model VIF -



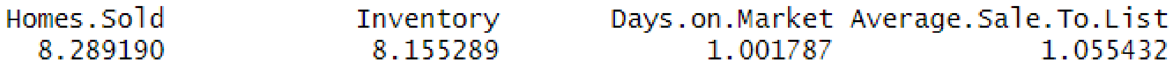
```
 Homes.Sold           Inventory      Days.on.Market Average.Sale.To.List 
   4.513202            4.155777            1.428998             1.561171 
```

Texas Full Model VIF -



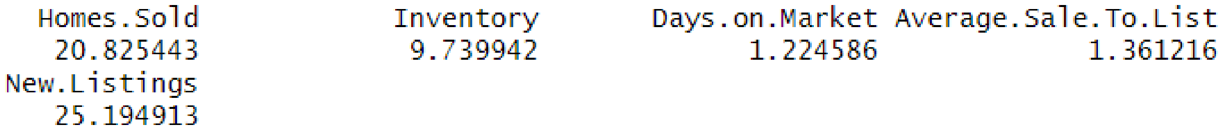
```
     Homes.Sold          Inventory    Days.on.Market Average.Sale.To.List 
      26.768660          10.163453          1.001788             1.056389 
   New.Listings 
      32.969320 
```

Texas Reduced Model VIF -



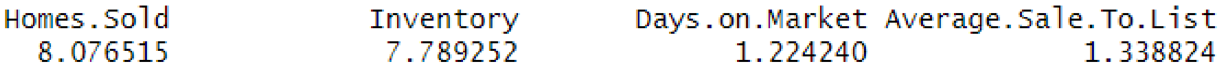
```
    Homes.Sold          Inventory     Days.on.Market Average.Sale.To.List 
      8.289190           8.155289           1.001787             1.055432 
```

California Full Model VIF -



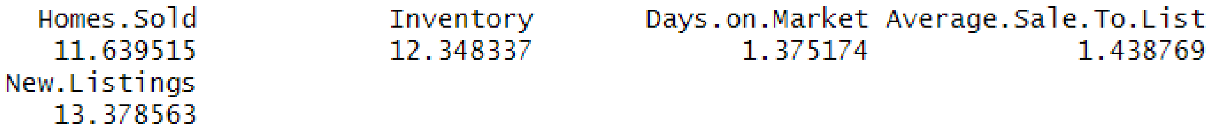
```
     Homes.Sold           Inventory       Days.on.Market Average.Sale.To.List 
      20.825443            9.739942             1.224586             1.361216 
   New.Listings 
      25.194913 
```

California Reduced Model VIF-



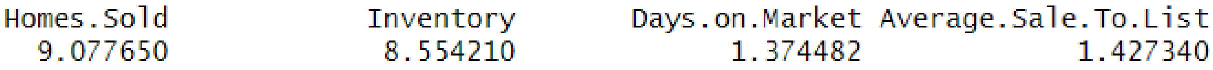
```
  Homes.Sold            Inventory        Days.on.Market Average.Sale.To.List 
    8.076515             7.789252              1.224240             1.338824 
```

New York Full Model VIF -



```
      Homes.Sold           Inventory      Days.on.Market Average.Sale.To.List 
       11.639515           12.348337            1.375174             1.438769 
    New.Listings 
       13.378563 
```

New York Reduced Model VIF -



```
  Homes.Sold            Inventory       Days.on.Market Average.Sale.To.List 
    9.077650             8.554210             1.374482             1.427340 
```

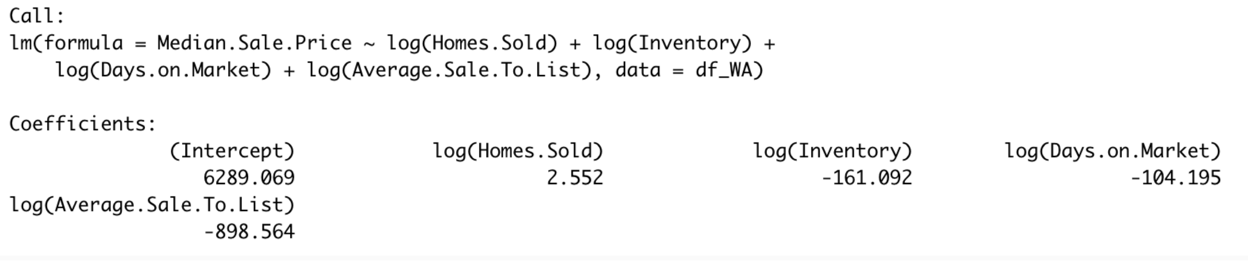
Then we applied a transformation to our model, leading us to check the VIF again of the transformed model. For California, Texas, and Washington the value outputted for homes sold was greater than 10 so, we decided to remove the regressor to bring down the collinearity of the models. Notably, for New York, none of the values for the VIF exceed 10 however are close with values of 8 and 9. We chose to keep them in the model. From this point on we were done looking at VIF's, and our models had no collinearity from this point forward.

From what we have seen the only predictor that we removed is New Listings for each modeling of each metroplex area prior to the transformations and.

Transformations

Although most of the residuals seemed to fall within the confidence bounds for residuals, we noticed that there were unfavorable shapes that were formed with the R-Studentized residuals such as fanning and double bowing. These shapes suggest that the variance is not constant and a transformation may be favorable for data analysis. In an attempt to remove the unfavorable shapes we tried the reciprocal transformation, reciprocal square root transformation, Square root transformation, log(Y) transformation, log(X) transformation, log(Y) and log(X) transformation, and the Box-Cox method for transformation. After transforming the data, we chose at most 3 transformations for each state with the most evenly spread data within the bounds and had minimal shape. Then we checked for normality of the residuals for the selected transformations to determine which transformation would be best.

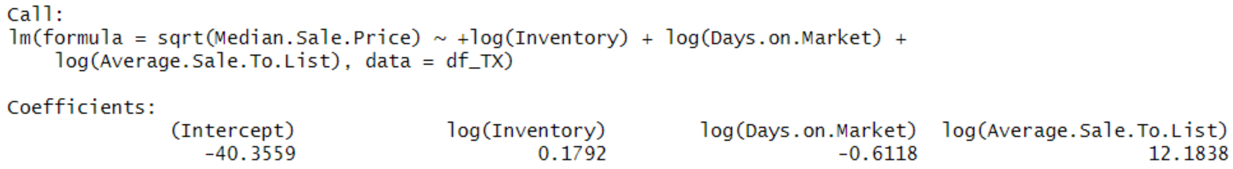
Washington Transformed Model -



```
Call:
lm(formula = Median.Sale.Price ~ log(Homes.Sold) + log(Inventory) +
    log(Days.on.Market) + log(Average.Sale.To.List), data = df_WA)

Coefficients:
              (Intercept)            log(Homes.Sold)             log(Inventory)        log(Days.on.Market)
                 6289.069                      2.552                   -161.092                   -104.195
log(Average.Sale.To.List)
                 -898.564
```

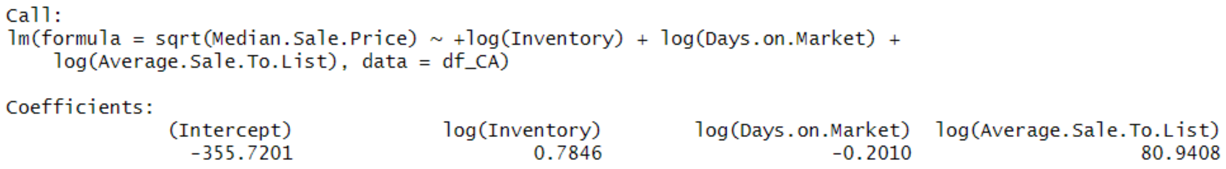
Texas Transformed Model -



```
Call:
lm(formula = sqrt(Median.Sale.Price) ~ +log(Inventory) + log(Days.on.Market) +
    log(Average.Sale.To.List), data = df_TX)

Coefficients:
              (Intercept)             log(Inventory)        log(Days.on.Market)  log(Average.Sale.To.List)
                 -40.3559                     0.1792                    -0.6118                    12.1838
```

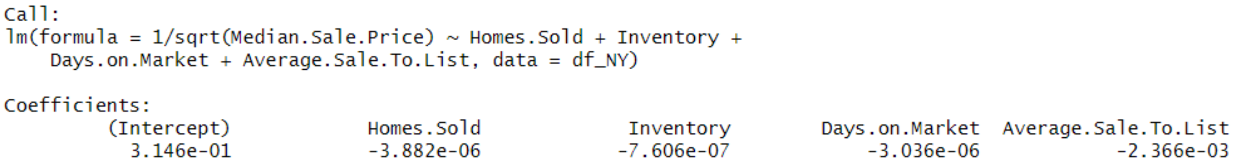
California Transformed Model -



```
Call:
lm(formula = sqrt(Median.Sale.Price) ~ +log(Inventory) + log(Days.on.Market) +
    log(Average.Sale.To.List), data = df_CA)

Coefficients:
              (Intercept)             log(Inventory)        log(Days.on.Market)  log(Average.Sale.To.List)
                -355.7201                     0.7846                    -0.2010                    80.9408
```

New York Transformed Model -



```
Call:
lm(formula = 1/sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
    Days.on.Market + Average.Sale.To.List, data = df_NY)

Coefficients:
         (Intercept)            Homes.Sold             Inventory        Days.on.Market  Average.Sale.To.List
           3.146e-01            -3.882e-06            -7.606e-07            -3.036e-06            -2.366e-03
```

Comparing Models

We noticed that the R^2 for Texas is very low: 0.07622. The Washington model had the highest adjusted R^2 of 0.5275. Although none of our coefficients of determination for these four transformed models do not exceed 0.6, i.e., the model for each state does not thoroughly explain the variance in the median sale price, we believe that these models are still significant because the errors are normalized and the variances are more constant. We preferred models that have no patterns in residuals but with less R^2 than those that have high R^2 but have a clear trend in the residuals plots.

Each of the four transformations we chose have very similar R^2 and adjusted R^2. For example, This seems to confirm that our transformed reduced models are better because we can get a good R^2 with fewer variables. In essence, our transformed reduced models include only the most significant regressors (inventory, days on market, and average sale to list) that do not seem to reduce our model's ability to explain the variance in the median sale price.

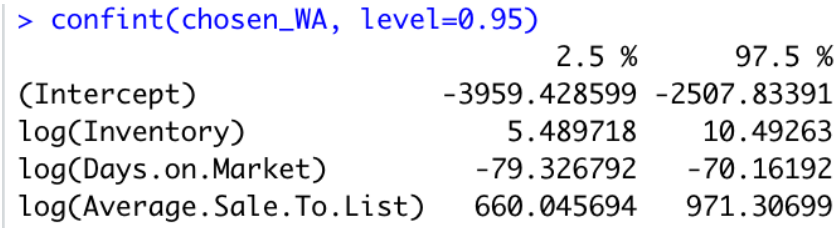
We proceeded with testing significance of regression on the model for each state at a significance level of 0.05 as a way to test for global model adequacy. The test for each model, indicated that there was *at least one variable that was significant* to our model, i.e., the F-statistic we got for each test is far greater than the critical bound of the F-distribution.

Then, we performed a test of significance of individual regressors to see the level of contribution of each particular variable to the model. The null hypothesis is that the coefficient of the individual regressor is zero, e.g., the coefficient of inventory is zero. The variables inventory, days on market, and average sale to list each have miniscule p-values. For example, based on our transformed model for Texas, the p-values for the three variables log(inventory), log(days on market), and log(average sale to list) are all less than 1.0×10^{-13}, which means that there is strong evidence that given the other variables, each variable is significant (unlikely to have a coefficient of zero). This serves as further corroboration that we chose the crucial variables in our model.

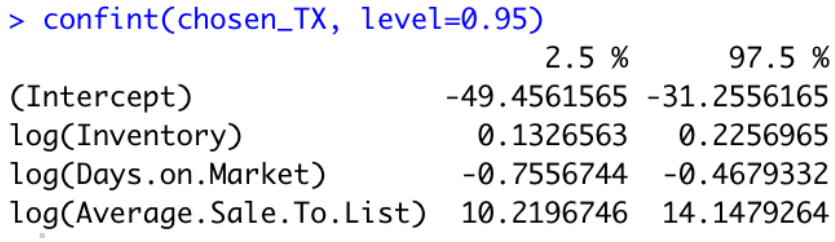
Throughout all this residual and influence analysis, we decided that the best model we have are the transformed models for each state. We fixed many potential issues (e.g., clear patterns in the residuals plot and lack of normality in the residuals) to reach the transformed model. Leaving us with a model that would more accurately predict, and model our data.

After an initial residual and influence analysis, we know that the transformed models are better because there is the most linearization in these models since there is normality of residuals and more scatter in the residuals plots.

Confidence Intervals on the Parameters

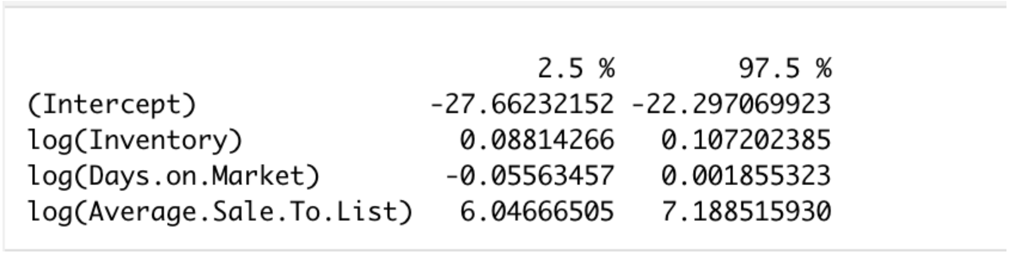


```
> confint(chosen_WA, level=0.95)
                               2.5 %      97.5 %
(Intercept)             -3959.428599 -2507.83391
log(Inventory)              5.489718    10.49263
log(Days.on.Market)       -79.326792   -70.16192
log(Average.Sale.To.List)  660.045694   971.30699
```



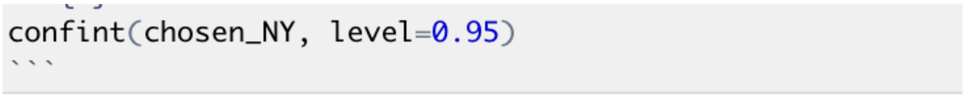
```
> confint(chosen_TX, level=0.95)
                               2.5 %      97.5 %
(Intercept)              -49.4561565 -31.2556165
log(Inventory)             0.1326563   0.2256965
log(Days.on.Market)       -0.7556744  -0.4679332
log(Average.Sale.To.List) 10.2196746  14.1479264
```

For CA:

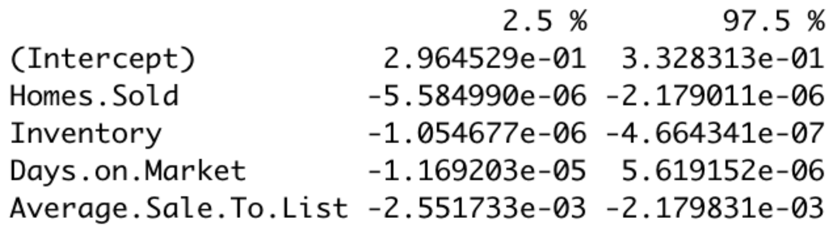


```
                                2.5 %        97.5 %
(Intercept)               -27.66232152 -22.297069923
log(Inventory)              0.08814266   0.107202385
log(Days.on.Market)        -0.05563457   0.001855323
log(Average.Sale.To.List)   6.04666505   7.188515930
```

For NY:



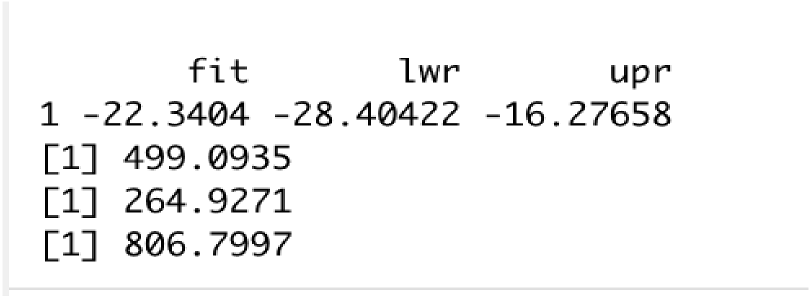
~~~
confint(chosen_NY, level=0.95)
```
~~~



```
                           2.5 %        97.5 %
(Intercept)             2.964529e-01  3.328313e-01
Homes.Sold             -5.584990e-06 -2.179011e-06
Inventory              -1.054677e-06 -4.664341e-07
Days.on.Market         -1.169203e-05  5.619152e-06
Average.Sale.To.List   -2.551733e-03 -2.179831e-03
```

Sample CI for mean response (for TX)

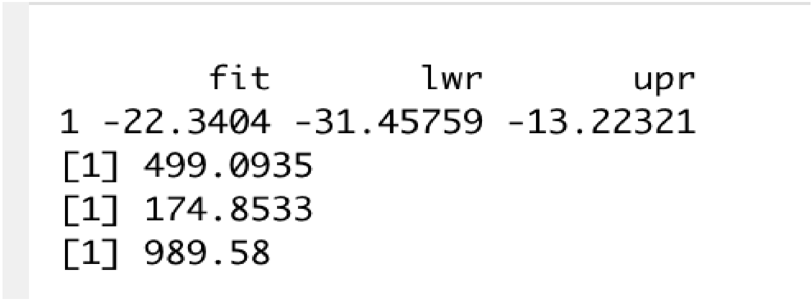
Based on Inventory=8000, Days.on.Market=80, Average.Sale.To.List=97



```
       fit       lwr       upr
1 -22.3404 -28.40422 -16.27658
[1] 499.0935
[1] 264.9271
[1] 806.7997
```

Sample PI for TX

Based on Inventory=8000, Days.on.Market=80, Average.Sale.To.List=97



```
      fit       lwr       upr
1 -22.3404 -31.45759 -13.22321
[1] 499.0935
[1] 174.8533
[1] 989.58
```

Influence Analysis

In general there were many influential points for each state where the only test that was flagged was the covariance ratio test. Due to how strict R is and our data set size we would not remove points simply due to the covariance ratio tests and would take into account a combination of factors such as extreme values within each test, combination of flagged tests, as well as the size of the residual for the point. For Texas we removed 3 points for all the removed values the covariance ratios had extreme values and were flagged for the dffits test signifying that our model was made worse with the values because it negatively impacts our model by decreasing precision and by greatly influencing y hat. This means it is affecting y-hat in a negative manner suggesting that we should get rid of them. The R-studentized residuals for two of the values were over ten standard deviations away as well. Similarly, for New York, three of the four points flagged both covariance and dffits and had extreme values for both the covariance ratio and the dffits suggesting it is significantly affecting our model negatively. For the last point removed in the New York data set the hat test was marked and the covariance ratio test was marked. That alone would not have qualified it for removal since it only affects precision of the model and the variance of the x values however since the R-studentized residual was also large we determined that it heavily impacted the model in a negative way. For California, all of the points removed were outside of the Bonferroni corrected bounds with extreme residuals and were flagged for dffits and hat values which indicate that excessive influence was imposed by the values. For Washington we only removed one point which flagged the dfbetas for the intercept and average sale to list as well as having a high dffit test, an extreme value for covariance ratio which was less than one, cooksd was flagged, and the hat test was flagged. This signifies that this point greatly influenced the intercept and Average sale to list price in a negative way since the covariance ratio test was far below one as well as being flagged for dffits meaning it significantly impacted our model in a negative way.

Conclusion

Originally the data set we chose had around 20 predictors and 21,303 observations. However, we discovered that only five of those predictors would be useful for our model collection. The main reason why we dropped so many predictors is that we sought more interpretability for our model, and some of these predictors were hard to interpret with our model. They generally already had some connection to some of the other predictors, e.g., the month-over-month and year-over-year average sale to list columns. Thus, we removed these predictors from our data. With the new current data set, we have 19,849 observations and five variables which are median sale price, homes sold, inventory, days on market, and average sale to list.

Through the use of transformations we were able to fit a great model for each region state. Each is slightly different from before, although all the models have the same predictors. The best models for each region are referenced in the transformations portion of the report. The transformed Washington model seems to be the best model to use because it has the highest R^2 and reasonable CI's.

The Texas model has the lowest R^2 but reasonable CI's therefore use this model with caution. This is just something to note if we have any problems with the Texas model we created. Our model gave a reasonable confidence interval for the mean response and prediction interval for the values that we tried, which were based on actual x-values in February 2023 for Austin, TX. The actual median sale price for Austin in February was $439,000, which lies in both the confidence interval of mean response and the prediction interval.

We did not remove New York's Homes Sold column, since it didn't reach the full threshold for when we remove it. Although if we do have problems with multicollinearity we already know what predictors would be the problem for the New York model.

Average sale to list seems to have the largest magnitude of regression coefficient for each model; i.e., it affects median sale price the most. Therefore we know what predictors are the most important to our model. With this we can make sure if we ever try to make another model with a new data set, we can expect average sale to list to be an important predictor.

Reflection

This project allowed us to use our skills that we learned in this class and apply that to a real dataset. We were able to identify certain characteristics that make a house more popular in the housing market. Data cleaning required filtered out the rows that did not have the word "National" under the Region column, converting some of the types into integers to help further analyze our data, and getting rid of any incomplete data points. After the data cleaning, we were able to move on to further analyzing our data by splitting our data in the Region column into points with City and State. We split our data set into four different data frames, Washington, Texas, California and New York and created models for each state. We created graphs to evaluate the data using Median Sale Price against the regressors (number of homes sold, inventory, average sale price to list price ratio, and the amount of days on the market). This helps us to see if there is a correlation between any of these pairs.

We applied several transformations on our models to see which transformation is the best for each model. We applied the Log(X), Log(Y), Square Root, Reciprocal Square Root, and Reciprocal transformations. The best for California ended up being the Log(X) and Log(Y) transformations. We also created Influence Index Plots for the New York, Texas, California and Washington region. We then went on to check for multicollinearity after the transformations for each region, creating the residual plots and VIF.

For further analysis, we could collect more data from around the United States and create a linear model for the entire United States. We could also explore the difference in median sale price and trends of inventory, days on market, and average sale to list between pre-pandemic and post-pandemic times. We could collect data that is based off of each state and not just regions to get a more accurate estimation of the housing prices in each state. This could show us which state is more popular in the housing market and which areas are not doing as well.

Appendix

References

"Data Center." *Redfin Real Estate News*, 3 Mar. 2023, www.redfin.com/news/data-center/ .

Team Responsibilities/Roles

Orlando Malanco - Transformations, Influence Analysis, Multicollinearity, Data Cleaning, Residual Analysis
Subre Moktar - Subset Selection, VIF, Full modeling, Data Cleaning
Sayema Rahman - Conclusion, Reflection, Graphs for Exploratory Data Analysis
Elaine Wu - Transformations, Influence Analysis, Residual Analysis, Confidence of regression parameters, prediction Intervals, Confidence of mean and response Intervals

Code

```
library(MASS)
library(dplyr)
library(stringr)
library(pairsD3)
library(ggplot2)
library(readr)
library(car)
library(ggpubr)
getwd()
setwd("C:/Users/subre/Documents/Python Code/Python Data Folder")
df <- read.csv("data.csv", sep = "\t", fileEncoding = "UTF-16LE", header = T)
setwd("C:/Development/R Program/Data Set Saved Here")
df_2 <- read.csv("Metro_Data.csv", sep = "\t", header = T, fileEncoding = "UTF-16LE")


#summary(df)
df2 <- df_2 %>% mutate_at(3:14, parse_number) %>% mutate_at(18:20, parse_number)
df2 <- na.omit(df2)
#df2 <- df2 %>% select(df2 , -c(Median.Sale.Price.MOM,Median.Sale.Price.YoY,
Homes.Sold.MoM,
#                             Homes.Sold.YoY,))
df2$Days.on.Market <- as.numeric(df2$Days.on.Market)
df2$Days.on.Market.MoM <- as.numeric(df2$Days.on.Market.MoM)
df2$Days.on.Market.YoY <- as.numeric(df2$Days.on.Market.YoY)
print(summary(df2))


library(stringr)
a = str_detect(df2$Region, "TX")
b = str_detect(df2$Region, "WA")
c = str_detect(df2$Region, "NY")
d = str_detect(df2$Region, "FL")
e = str_detect(df2$Region, "CA")
df2$Region_State <- ifelse(a == TRUE, "TX",
                    ifelse(b == TRUE, "WA",
                           ifelse(c == TRUE, "NY",
                                  ifelse(d == TRUE, "FL",
                                  ifelse(e == TRUE, "CA", NA)))))
```

```
unique(df2$Region_State)

df2 <- df2[-grep("FL", df2$Region_State),]
ggplot(df2) + geom_point(aes(y = Median.Sale.Price, x = New.Listings, color = Region_State))
ggplot(df2) + geom_point(aes(y = Median.Sale.Price, x = Homes.Sold, color = Region_State))
```

Before Transformation

```
ggplot(df_new) + geom_point(aes(y = Median.Sale.Price, x = Homes.Sold, color = State)) +
ggtitle("Median Sale Price vs Homes Sold")
ggplot(df_new) + geom_point(aes(y = Median.Sale.Price, x = Inventory, color = State)) +
ggtitle("Median Sale Price vs Inventory")
ggplot(df_new) + geom_point(aes(y = Median.Sale.Price, x = Days.on.Market, color = State)) +
ggtitle("Median Sale Price vs Days on Market")
ggplot(df_new) + geom_point(aes(y = Median.Sale.Price, x = Average.Sale.To.List, color =
State)) + ggtitle("Median Sale Price vs Average Sale to List")
```

```
ggplot(df2) + geom_point(aes(y = Median.Sale.Price, x = New.Listings, color = Region_State))
ggplot(df2) + geom_point(aes(y = Median.Sale.Price, x = Homes.Sold, color = Region_State))
library(MASS)
ggplot(df2) + geom_boxplot(aes(y = Median.Sale.Price, x = Region_State))
ggplot(df2) + geom_boxplot(aes(y = Median.Sale.Price, x = Region_State)) + ylim(0,2000)
```

Renaming the Columns

```
names(df2)[1] <- "x1" #Region
names(df2)[2] <- "x2" #Month.of.period.end
names(df2)[3] <- "y" #median.sale.price
names(df2)[4] <- "x3" #median.sale.price.mom
names(df2)[5] <- "x4" #median.sale.price.yoy
names(df2)[6] <- "x5" #homes.sold
names(df2)[7] <- "x6" #homes.sold.mom
names(df2)[8] <- "x7" #homes.sold.yoy
names(df2)[9] <- "x8" #new.listings
names(df2)[10] <- "x9" #new.listings.mom
names(df2)[11] <- "x10" #new.listings.yoy
names(df2)[12] <- "x11" #inventory
names(df2)[13] <- "x12" #inventory.mom
names(df2)[14] <- "x13" #inventory.yoy
names(df2)[15] <- "x14" #days.on.market
names(df2)[16] <- "x15" #days.on.market.mom
names(df2)[17] <- "x16" #days.on.market.yoy
names(df2)[18] <- "x17" #average.sale.to.list
names(df2)[19] <- "x18" #average.sale.to.list.mom
names(df2)[20] <- "x19" #average.sale.to.list.yoy

str(df2)

housing_price <- lm(y ~
x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+x16+x17+x18+x19, data = df2)
summary(housing_price)
library(MASS)
modelSelection <- stepAIC(housing_price, direction = "forward")
modelSelection$anova

library(leaps)
reg <- regsubsets(y ~ x5+x6+x7+x8+x9+x10+x11+x12+x13+x14+x15+x16+x17+x18+x19,
                  data = df2,
                  nvmax = 17)
reg_summary <- summary(reg)
names(reg_summary)
coef(reg, 7)

#shinypairs(df2[3:20])
```

Pair Plot

```
df_new = df2[,c(-4,-5,-7,-8,-10,-11,-13,-14,-16,-17,-19,-20)]
housing_price_linear <- lm(y ~ x5+x8+x11+x14+x17, data = df_new)
summary(housing_price_linear)
#shinypairs(df_new)
#pairsD3(df_new[,c(4,5,6,7,8,3)], opacity = 0.9, cex = 3, width = 600)
pdf("mygraph.pdf", width = 10, height = 10)
pairs(df_new[,c(4,5,6,7,8,3)], col = 'red', cex = 1 )
```

VIF

```
library(car)
vif(housing_price_linear)
# X5, and x8 are definitly colinear. therefore, we dont need one of them

housing_price_nocolinear <- lm(y ~ x5+x11+x14+x17, data = df_new)
vif(housing_price_nocolinear)
```

Model Selection

```
df2$Region_State<-as.factor(df2$Region_State)
summary(df2)
modelSelection <- stepAIC(housing_price_nocolinear, direction = "forward")
modelSelection$results
reg <- regsubsets(y ~ x5+x11+x14,
                  data = df2)
summary(reg)
reg$nbest
```

RECIPROCAL TRANSFORMATION

```
full_Model_NY <- lm(data=df_NY, 1/Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim = c(-2.5, 2.5))
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)
```

RECIPROCAL SQUARE ROOT TRANSFORMATION (Appears to be Best for NY)

```
full_Model_NY <- lm(data=df_NY, 1/sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "Orange", main = "NY
Residuals After Transformation",
col.main = "Orange", col.lab = "Orange", pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Orange", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Orange", lwd=2)
```

```
par(mfrow=c(1,2))
hist(studres(full_Model_NY), breaks=40, freq=F, col="Orange", col.lab = "Orange",
cex.axis=1.5, cex.lab=1.5, cex.main=1, col.main = "Orange")
qqPlot(full_Model_NY, col.lines = "Orange")
```

SQRT TRANSFORMATION

```
full_Model_NY <- lm(data=df_NY, sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)
```

LOG(Y) TRANSFORMATION

```
full_Model_NY <- lm(data=df_NY, log(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)

par(mfrow=c(1,2))
hist(studres(full_Model_NY), breaks=40, freq=F, col="cornflowerblue",
cex.axis=1.5, cex.lab=1.5, cex.main=2)
qqPlot(full_Model_NY)
```

LOG(X) TRANSFORMATION

```
full_Model_NY <- lm(data=df_NY, Median.Sale.Price ~ log(Homes.Sold) + log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)
```

LOG(X) & LOG(Y) TRANSFORMATION

```
full_Model_NY <- lm(data=df_NY, log(Median.Sale.Price) ~ log(Homes.Sold) + log(Inventory)
+ log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)

#This chunk of code is to verify that after the transformation the errors are normally distributed
after the transformation so we can apply the transormation x* = log(x)

par(mfrow=c(1,2))
hist(studres(full_Model_NY), breaks=40, freq=F, col="cornflowerblue",
cex.axis=1.5, cex.lab=1.5, cex.main=2)
qqPlot(full_Model_NY)
```

BoxCox

```
library(car)
boxcox.df_NY <- boxCox(df_NY$Median.Sale.Price ~ df_NY$Homes.Sold + df_NY$Inventory
+ df_NY$Days.on.Market + df_NY$Average.Sale.To.List, lambda=seq(-2,2,1/10))
boxcox.df_NY
boxcox.df_NY$x[which.max(boxcox.df_NY$y)]

boxcox_Model <- lm(df_NY$Median.Sale.Price^(-.02) ~ df_NY$Homes.Sold +
df_NY$Inventory + df_NY$Days.on.Market + df_NY$Average.Sale.To.List)
boxcox_Model

residualPlot(boxcox_Model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=4.3346, col = "Red", lwd=2)
abline(h=-4.3346, col = "Red", lwd=2)

par(mfrow=c(1,2))
hist(studres(full_Model_NY), breaks=40, freq=F, col="cornflowerblue",
cex.axis=1.5, cex.lab=1.5, cex.main=1)
qqPlot(full_Model_NY)
```

~~~
# This takes away the MOM and YOY Data values
```{r}
head(df_new)
df_new = df_new[,c(-4,-5,-7,-8,-10,-11,-13,-14,-16,-17,-19,-20)]
```

```{r}
head(df_new)
df_new = df_new %>% filter(Region != " National")
```

#this portion of code splits the region into to dataa points one with the City and one with the state
```{r}
library(tidyr)
df_new <- separate(data = df_new, col = "Region", into = c("City", "State"), sep = "\\,")
df_new <- separate(data = df_new, col = "State", into = c("Reject", "State"), sep = " ")
```

#this chunk of code was written to see whether or not we have NA's
```{r}
df_new = df_new[,-2]
summary(df_new)
```

#this chunk of code was used to get rid of incomplete data points
```{r}
df_new = na.omit(df_new)
summary(df_new)
```

```{r}
df_new$Days.on.Market <- as.numeric(gsub(",","",df_new$Days.on.Market))
full_model <- lm(df_new$Median.Sale.Price ~  df_new$Homes.Sold + df_new$New.Listings +
df_new$Days.on.Market + df_new$Average.Sale.To.List + df_new$Inventory)
```

```{r}
df_new = na.omit(df_new)
summary(df_new)
```
~~~

~~~
```{r}
df_WA <- df_new[which(df_new$State == "WA"),]
rownames(df_WA) <- 1:nrow(df_WA)

df_TX <- df_new[which(df_new$State == "TX"),]
rownames(df_TX) <- 1:nrow(df_TX)

df_CA <- df_new[which(df_new$State == "CA"),]
rownames(df_CA) <- 1:nrow(df_CA)

df_NY <- df_new[which(df_new$State == "NY"),]
rownames(df_NY) <- 1:nrow(df_NY)
```

######
###### WA

```{r}
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List + New.Listings)

vif(full_Model_WA)
```
```{r}
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
vif(full_Model_WA)
```


```{r}
Rstudent_Bounds_WA <- 4.2898
print(sd(df_WA$Median.Sale.Price))
print(mean(df_WA$Median.Sale.Price))
```
```{r}
~~~

NO TRANSFORMATION

```
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_WA, type="rstudent", quadratic=F, col = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
```

RECIPROCAL TRANSFORMATION

```
full_Model_WA <- lm(data=df_WA, log(1/Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_WA, type="rstudent", quadratic=F, col = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim = c(-5,5))
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
```

RECIPROCAL SQUARE ROOT TRANSFORMATION

```
full_Model_WA <- lm(data=df_WA, 1/sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_WA, type="rstudent", quadratic=F, col = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim = c(-5,5))
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
```

SQRT TRANSFORMATION

```
full_Model_WA <- lm(data=df_WA, sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_WA, type="rstudent", quadratic=F, col = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim=c(-5,8))
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
```

LOG(Y) TRANSFORMATION

~~~
full_Model_WA <- lm(data=df_WA, log(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_WA, type="rstudent", quadratic=F, col = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim=c(-5,5))
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
~~~

LOG(X) TRANSFORMATION (APPEARS TO BE BEST FOR WASHINGTON)

~~~
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ log(Homes.Sold) + log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_WA, type="rstudent", quadratic=F, col = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim = c(-6,6), main = "WA Residuals After
Transformation", col.main = "Purple", col.lab = "Purple")
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
~~~

BOXCOX METHOD

~~~
boxcox.df_WA <- boxCox(df_WA$Median.Sale.Price ~ df_WA$Homes.Sold +
df_WA$Inventory + df_WA$Days.on.Market + df_WA$Average.Sale.To.List,
lambda=seq(-2,2,1/10))
boxcox.df_WA
boxcox.df_WA$x[which.max(boxcox.df_WA$y)]
boxcox_Model <- lm(df_WA$Median.Sale.Price^(.465) ~ df_WA$Homes.Sold +
df_WA$Inventory + df_WA$Days.on.Market + df_WA$Average.Sale.To.List)
boxcox_Model

residualPlot(boxcox_Model, type="rstudent", quadratic=F, col = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
```
~~~

~~~
###### TX
```{r}
full_Model_TX <- lm(data=df_TX, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List + New.Listings)
vif(full_Model_TX)
```

```{r}
full_Model_TX <- lm(data=df_TX, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
vif(full_Model_TX)
```
```{r}
Rstudent_Bounds_TX <- 4.445
print(sd(df_TX$Median.Sale.Price))
print(mean(df_TX$Median.Sale.Price))
```
```{r}
~~~

NO TRANSFORMATION

````
full_Model_TX <- lm(data=df_TX, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim = c(-5,6))
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
````

RECIPROCAL TRANSFORMATION

````
full_Model_TX <- lm(data=df_TX, 1/Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
````

RECIPROCAL SQUARE ROOT TRANSFORMATION

```
full_Model_TX <- lm(data=df_TX, 1/sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
```

SQRT TRANSFORMATION

```
full_Model_TX <- lm(data=df_TX, sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
```

LOG(Y) TRANSFORMATION

```
full_Model_TX <- lm(data=df_TX, log(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
```

LOG(X) TRANSFORMATION

```
full_Model_TX <- lm(data=df_TX, Median.Sale.Price ~ log(Homes.Sold) + log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim = c(-5,5))
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
```

SQRT(Y) LOG(X) TRANSFORMATION (Appears to be Best for Texas)

```
full_Model_TX <- lm(data=df_TX, sqrt(Median.Sale.Price) ~ log(Homes.Sold) + log(Inventory)
+ log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1, ylim = c(-5,5), main = "TX Residuals After
Transformation", col.main = "Red", col.lab = "Red")
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)

par(mfrow=c(1,2))
hist(studres(full_Model_TX), breaks=100, freq=F, col="Red",
cex.axis=1.5, cex.lab=1.5, cex.main=1, xlim = c(-5,5), col.main = "Red", col.lab = "Red")
qqPlot(full_Model_TX, col.lines = "Red")
```

LOG(X) & LOG(Y) TRANSFORMATION

```
full_Model_TX <- lm(data=df_TX, log(Median.Sale.Price) ~ log(Homes.Sold) + log(Inventory)
+ log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_TX, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)


#This chunk of code is to verify that after the transformation the errors are normally distributed
after the transformation so we can apply the transormation x* = log(x)

par(mfrow=c(1,2))
hist(studres(full_Model_TX), breaks=50, freq=F, col="Red",
cex.axis=1.5, cex.lab=1.5, cex.main=2, xlim = c(-5,5))
qqPlot(full_Model_TX)

#BoxCox
```

~~~
library(car)
boxcox.df_TX <- boxCox(df_TX$Median.Sale.Price ~ df_TX$Homes.Sold + df_TX$Inventory
+ df_TX$Days.on.Market + df_TX$Average.Sale.To.List, lambda=seq(-2,2,1/10))
boxcox.df_TX
boxcox.df_TX$x[which.max(boxcox.df_TX$y)]

boxcox_Model <- lm(df_TX$Median.Sale.Price^(.06) ~ df_TX$Homes.Sold + df_TX$Inventory
+ df_TX$Days.on.Market + df_TX$Average.Sale.To.List)
boxcox_Model

residualPlot(boxcox_Model, type="rstudent", quadratic=F, col = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
```

######
###### CA

```{r}
full_Model_CA <- lm(data=df_CA, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List + New.Listings)
vif(full_Model_CA)
```

```{r}
full_Model_CA <- lm(data=df_CA, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
vif(full_Model_CA)
```



```{r}
Rstudent_Bounds_CA <- 4.3975
print(sd(df_CA$Median.Sale.Price))
print(mean(df_CA$Median.Sale.Price))
```



```{r}
~~~

```
## NO TRANSFORMATION
full_Model_CA <- lm(data=df_CA, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)
```

RECIPROCAL TRANSFORMATION

```
full_Model_CA <- lm(data=df_CA, 1/Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)
```

RECIPROCAL SQUARE ROOT TRANSFORMATION

```
full_Model_CA <- lm(data=df_CA, 1/sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
```

SQRT TRANSFORMATION

```
full_Model_CA <- lm(data=df_CA, sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)
```

LOG(Y) TRANSFORMATION

```
full_Model_CA <- lm(data=df_CA, log(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
```

```
residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)
```

LOG(X) TRANSFORMATION

```
full_Model_CA <- lm(data=df_CA, Median.Sale.Price ~ log(Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List))
```

```
residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Red", lwd=2)
```

LOG(X) TRANSFORMATION

```
full_Model_CA <- lm(data=df_CA, Median.Sale.Price ~ log(Homes.Sold) + log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
```

```
residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)
```

LOG(X) & LOG(Y) TRANSFORMATION (Appears to be Best for CA)

```
full_Model_CA <- lm(data=df_CA, log(Median.Sale.Price) ~ log(Homes.Sold) + log(Inventory)
+ log(Days.on.Market) + log(Average.Sale.To.List))
```

````
residualPlot(full_Model_CA, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1, main = "CA Residuals After Transformation", col.main =
"Blue", col.lab = "Blue")
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)


#This chunk of code is to verify that after the transformation the errors are normally distributed
after the transformation so we can apply the transormation x* = log(x)

par(mfrow=c(1,2))
hist(studres(full_Model_CA), breaks=40, freq=F, col="Blue", col.lab = "Blue",
cex.axis=1.5, cex.lab=1.5, cex.main=1, col.main = "Blue")
qqPlot(full_Model_CA, col.lines = "Blue")
````

BOXCOX

~~~
library(car)
boxcox.df_CA <- boxCox(df_CA$Median.Sale.Price ~ df_CA$Homes.Sold + df_CA$Inventory
+ df_CA$Days.on.Market + df_CA$Average.Sale.To.List, lambda=seq(-2,2,1/10))
boxcox.df_CA
boxcox.df_CA$x[which.max(boxcox.df_CA$y)]

boxcox_Model <- lm(df_CA$Median.Sale.Price^(.1818) ~ df_CA$Homes.Sold +
df_CA$Inventory + df_CA$Days.on.Market + df_CA$Average.Sale.To.List)
boxcox_Model

residualPlot(boxcox_Model, type="rstudent", quadratic=F, col = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)
```


######
###### NY

```{r}
full_Model_NY <- lm(data=df_NY, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List + New.Listings)
vif(full_Model_NY)
~~~

~~~
full_Model_NY <- lm(data=df_NY, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
vif(full_Model_NY)
```

```{r}
Rstudent_Bounds_NY <- 4.3342
print(sd(df_NY$Median.Sale.Price))
print(mean(df_NY$Median.Sale.Price))
```
~~~

NO TRANSFORMATION

~~~
```{r}
full_Model_NY <- lm(data=df_NY, Median.Sale.Price ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)
~~~

LOG(Y) TRANSFORMATION

~~~
full_Model_NY <- lm(data=df_NY, log(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)

par(mfrow=c(1,2))
hist(studres(full_Model_NY), breaks=40, freq=F, col="cornflowerblue",
cex.axis=1.5, cex.lab=1.5, cex.main=2)
qqPlot(full_Model_NY)
~~~

LOG(X) TRANSFORMATION

```
full_Model_NY <- lm(data=df_NY, Median.Sale.Price ~ log(Homes.Sold) + log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)


## LOG(X) & LOG(Y) TRANSFORMATION

full_Model_NY <- lm(data=df_NY, log(Median.Sale.Price) ~ log(Homes.Sold) + log(Inventory)
+ log(Days.on.Market) + log(Average.Sale.To.List))

residualPlot(full_Model_NY, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_NY, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_NY, col = "Red", lwd=2)


#This chunk of code is to verify that after the transformation the errors are normally distributed
after the transformation so we can apply the transormation x* = log(x)

par(mfrow=c(1,2))
hist(studres(full_Model_NY), breaks=40, freq=F, col="cornflowerblue",
cex.axis=1.5, cex.lab=1.5, cex.main=2)
qqPlot(full_Model_NY)


#BoxCox
library(car)
boxcox.df_NY <- boxCox(df_NY$Median.Sale.Price ~ df_NY$Homes.Sold + df_NY$Inventory
+ df_NY$Days.on.Market + df_NY$Average.Sale.To.List, lambda=seq(-2,2,1/10))
boxcox.df_NY
boxcox.df_NY$x[which.max(boxcox.df_NY$y)]

boxcox_Model <- lm(df_NY$Median.Sale.Price^(-.02) ~ df_NY$Homes.Sold +
df_NY$Inventory + df_NY$Days.on.Market + df_NY$Average.Sale.To.List)
boxcox_Model
```

~~~~
residualPlot(boxcox_Model, type="rstudent", quadratic=F, col = "dodgerblue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=4.3346, col = "Red", lwd=2)
abline(h=-4.3346, col = "Red", lwd=2)

par(mfrow=c(1,2))
hist(studres(full_Model_NY), breaks=40, freq=F, col="cornflowerblue",
cex.axis=1.5, cex.lab=1.5, cex.main=1)
qqPlot(full_Model_NY)
```
#The next chunks of code are to check for multicolliniarity after the transformations
```{r}
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ log(Homes.Sold) + log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
vif(full_Model_WA)
```


```{r}
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ log(Inventory) + log(Days.on.Market)
+ log(Average.Sale.To.List))
vif(full_Model_WA)
```


```{r}
residualPlot(full_Model_WA, type="rstudent", quadratic=F, col="Purple", col.lab = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
```


```{r}
options(max.print=5000)
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ log(Inventory) + log(Days.on.Market)
+ log(Average.Sale.To.List))
myInf <- influence.measures(full_Model_WA)
summary(myInf)
```


```{r}
df_WA <- df_WA[-676,]
~~~~

~~~
```

```{r}
full_Model_WA <- lm(data=df_WA, Median.Sale.Price ~ log(Inventory) + log(Days.on.Market)
+ log(Average.Sale.To.List))
myInf <- influence.measures(full_Model_WA)
summary(myInf)
```

```{r}
residualPlot(full_Model_WA, type="rstudent", quadratic=F, col="Purple", col.lab = "Purple",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_WA, col = "Purple", lwd=2)
abline(h=-Rstudent_Bounds_WA, col = "Purple", lwd=2)
```
```{r}
full_Model_TX <- lm(data=df_TX, sqrt(Median.Sale.Price) ~ log(Homes.Sold) + log(Inventory)
+ log(Days.on.Market) + log(Average.Sale.To.List))
vif(full_Model_TX)
```
```{r}
full_Model_TX <- lm(data=df_TX, sqrt(Median.Sale.Price) ~ log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
vif(full_Model_TX)
```
```{r}
residualPlot(full_Model_TX, type="rstudent", quadratic=F, col="Red", col.lab = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
```
```{r}
full_Model_TX <- lm(data=df_TX, sqrt(Median.Sale.Price) ~ log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
myInf <- influence.measures(full_Model_TX)
summary(myInf)
```
```{r}
df_TX <- df_TX[-3664,]
df_TX <- df_TX[-2182,]
~~~

~~~~
df_TX <- df_TX[-1299,]
```
```{r}
full_Model_TX <- lm(data=df_TX, sqrt(Median.Sale.Price) ~ log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
myInf <- influence.measures(full_Model_TX)
summary(myInf)
```
```{r}
residualPlot(full_Model_TX, type="rstudent", quadratic=F, col="Red", col.lab = "Red",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_TX, col = "Red", lwd=2)
abline(h=-Rstudent_Bounds_TX, col = "Red", lwd=2)
```
```{r}
full_Model_NY <- lm(data=df_NY, 1/sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
vif(full_Model_NY)
```
```{r}
full_Model_NY <- lm(data=df_NY, 1/sqrt(Median.Sale.Price) ~ Homes.Sold + Inventory +
Days.on.Market + Average.Sale.To.List)
myInf <- influence.measures(full_Model_NY)
summary(myInf)
```
```{r}
df_NY <- df_NY[-2169,]
df_NY <- df_NY[-2142,]
df_NY <- df_NY[-1053,]
df_NY <- df_NY[-1069,]
```
```{r}
full_Model_CA <- lm(data=df_CA, log(Median.Sale.Price) ~ log(Homes.Sold) + log(Inventory)
+ log(Days.on.Market) + log(Average.Sale.To.List))
vif(full_Model_CA)
```
```{r}
full_Model_CA <- lm(data=df_CA, log(Median.Sale.Price) ~ log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
vif(full_Model_CA)
~~~~

~~~~
```
```{r}
residualPlot(full_Model_CA, type="rstudent", quadratic=F, col="Blue", col.lab = "Blue",
pch=16, cex=.3, cex.axis=1, cex.lab=1)
abline(h=Rstudent_Bounds_CA, col = "Blue", lwd=2)
abline(h=-Rstudent_Bounds_CA, col = "Blue", lwd=2)
```
```{r}
full_Model_CA <- lm(data=df_CA, log(Median.Sale.Price) ~ log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
myInf <- influence.measures(full_Model_CA)
summary(myInf)
```
```{r}
df_CA <- df_CA[-3793,]
df_CA <- df_CA[-3790,]
df_CA<- df_CA[-3787,]
df_CA <- df_CA[-852,]
```
```{r}
full_Model_CA <- lm(data=df_CA, log(Median.Sale.Price) ~ log(Inventory) +
log(Days.on.Market) + log(Average.Sale.To.List))
myInf <- influence.measures(full_Model_CA)
summary(myInf)
```
```{r}
Inventory <-  ggplot(df_WA) + geom_point(aes(y = Median.Sale.Price, x = log(Inventory), color
= State))
MarketDays <- ggplot(df_WA) + geom_point(aes(y = Median.Sale.Price, x =
log(Days.on.Market), color = State))
SaleToList <- ggplot(df_WA) + geom_point(aes(y = Median.Sale.Price, x =
log(Average.Sale.To.List), color = State))
final <- ggarrange(MarketDays,Inventory, SaleToList, nrow = 2, ncol = 2)
final
```
```{r}
Inventory <-  ggplot(df_TX) + geom_point(aes(y = sqrt(Median.Sale.Price), x = log(Inventory),
color = State))
MarketDays <- ggplot(df_TX) + geom_point(aes(y = sqrt(Median.Sale.Price), x =
log(Days.on.Market), color = State))
~~~~

~~~
SaleToList <- ggplot(df_TX) + geom_point(aes(y = sqrt(Median.Sale.Price), x = log(Average.Sale.To.List), color = State))
final <- ggarrange(MarketDays,Inventory, SaleToList, nrow = 2, ncol = 2)
final
```
```{r}
Inventory <- ggplot(df_CA) + geom_point(aes(y = log(Median.Sale.Price), x = log(Inventory), color = State))
MarketDays <- ggplot(df_CA) + geom_point(aes(y = log(Median.Sale.Price), x = log(Days.on.Market), color = State))
SaleToList <- ggplot(df_CA) + geom_point(aes(y = log(Median.Sale.Price), x = log(Average.Sale.To.List), color = State))
final <- ggarrange(MarketDays,Inventory, SaleToList, nrow = 2, ncol = 2)
final
```
```{r}
MarketDays <- ggplot(df_NY) + geom_point(aes(y = 1/sqrt(Median.Sale.Price), x = Days.on.Market, color = State))
HomesSold <- ggplot(df_NY) + geom_point(aes(y = 1/sqrt(Median.Sale.Price), x = Homes.Sold, color = State))
Inventory <- ggplot(df_NY) + geom_point(aes(y = 1/sqrt(Median.Sale.Price), x = Inventory, color = State))
SaleToList <- ggplot(df_NY) + geom_point(aes(y = 1/sqrt(Median.Sale.Price), x = Average.Sale.To.List, color = State))
final <- ggarrange(MarketDays,HomesSold,Inventory, SaleToList, nrow = 2, ncol = 2)
final
```


=================================================================
~~~