up&uptech

R Markdown

```
library(evaluate)
library(plyr)
library(readr)
library(caret)
library(repr)
library(tidyverse)
library(dplyr)
library(boot)
library(aod)
library(ggplot2)
library(pastecs)
library(glmnet)
data <- read.csv("/Users/srinivasansathiamurthy/Desktop/mls-sample.csv")
caseshiller <- read.csv("/Users/srinivasansathiamurthy/Desktop/CSUSHPISA.csv")
#data
```

##Overview of Study

I present this analysis in order of my work flow. Before I begin, let me outline the main goals of this analysis:

1. Find and interpret a well-fitting predictive model of housing prices in Georgia independent of housing price inflation based on relevant (clean) features with sufficient data.

a) The reason I chose the independence of housing price inflation is due to the underlying distribution of housing listing dates in the data (namely, all prices came from only two years) and the exceptionally varied case-schiller index in the last two years. I did not want this to dominate over my other predictor variables, since I desired for this study to be applicable for future years based on fixed factors as opposed to a backward looking analysis of what already occurred.

2. Determine the most relevant features for predicting housing prices (among the ones chosen from a preliminary search of cleanliness)

The order of the overall work flow was:

1. Data cleaning
2. Feature selection/cleaning, model selection and checking model assumptions
3. Checking for multicollinearity
4. Lasso regression Fitting

##Data cleaning process:

First I look at the distribution of each variable. Some glaring observations stood out, and were dealt with accordingly:

1. There are approximately 8000 different houses in this data set. Of those, only 1 is in Virginia and 2 are in Alabama. These are outliers, so I discard them.
2. The school_district feature was very chaotic and hard to deal with, in addition to the fact that half of its data is nonexistent. I discarded this location indicator.
3. lotsize_sqft was null for approximately 900 houses. This is not too large of a sacrifice to exclude given 8000 original samples, and lotsize_sqft seems like a very important predictor of housing prices, so I removed the null variables.
4. sqft was 0 for approximately 246 of the remaining houses. This is also not too large of a sacrifice, and also seems like an important predictor to include.
5. I filtered out id because they were just ordering the elements of the original list, a feature I did not deem relevant.
6. I eliminated the the street variable because it had over 7000 unique elements, which is far too many to deal with in a regression model.

Then I proceeded to choose the features I plan to work with for the remainder of the study:

7. Of the remaining features, I chose to only include latitude, longtitude, baths_full, baths_half, listing_date, list_price, interior square feet, total lot size, property type, year built, and the number of beds the houses had due to cleanliness and encapsulation of the variables I most desire.

a) For the location aspect, I used the longitude and latitude quantitative features since they encapsulate the most information in terms of spatial location.

b) The other features I included all seemed relevant predictor variables.

c) For the features I didn't include, they were either eliminated beforehand, or were text heavy and too qualitative for my liking. One could use nlp/regex techniques.

8. I type casted some columns to make them more useful.
9. I combined the bath_full and bath_half by adding them together (bath_full+1/2*bath_half) into one bath column. I then got rid of those two columns.
10. Property_type needed some data cleaning as well due to erratic string inputs.

```
#table(data$state)
#table(data$street)
data<- data[!(data$state=="VA"| data$state=="AL"| data$lotsize_sqft=="null" | data$sqft==0),]
data <- subset(data, select = -c(state, id, street))

data2 <- subset(data, select = c(lat, lng, baths_full, baths_half, listing_date, list_price, sqft, lots

data2$lotsize_sqft<- as.integer(data2$lotsize_sqft)
data2$baths_full<- as.numeric(data2$baths_full)
data2$baths_half<- as.numeric(data2$baths_half)
data2$baths = data$baths_full+1/2*data$baths_half
data2<- subset(data2, select=-c(baths_full, baths_half))

#table(data2$property_type)
for(i in 1:nrow(data2)){
  if(data2$property_type[i] == "single family home"){
```

```
    data2$property_type[i] <- "Single Family Residence"
  }
  if(data2$property_type[i] == "townhouse"){
     data2$property_type[i] <- "Townhouse"
  }
  if( data2$property_type[i] == "condo"){
     data2$property_type[i] <- "Condominium"
  }
}
#data2
```

##Feature and Model Selection:

Going back to the main goal of the study, I wish to predict and I have training outcomes, so I seek a supervised algorithm. The predictor variable is also continuous, so one can use a regression of some sort. Of the remaining feature variables, I want to find and rank the most significant features in order of importance, and select only the relevant features. To this end, a lasso regression model is useful.

The lasso regression model is still a generalized linear model, so:

a) The relationship between the feature and outcome variables need to be linear

b) Error terms are conditionally gaussian

c) Variance should be roughly constant (no heteroskedacity)

d) The features are uncorrelated (this is more particular to lasso regression, since two highly correlated variables will create a double penalty on each other and make each others' coefficients smaller relative to the other feature variables.)

We only take care of the first three assumptions for now. For each variable, we first eliminate outliers, and then modify the features so that the above assumptions can be met:

1. For the listing data, I noticed that all the entries were from 2020 and 2021, with most in 2021. I know that the housing prices skyrocketed from 2020 to 2021 due to the supply chain crisis in addition to a lot of household savings. This prompted me to normalize housing prices for general housing price inflation using the case-schiller index to get rid of a potential confounding variable. Plotting housing prices yielded that they were right skewed, and thus lognormally distributed, so I took the log of that column. I then removed the listing date data since I already incorporated that elsewhere through normalizing. When I plotted the histogram of lot_price, both tails seemed continuous, and the outliers weren't egregiously away from the first and third quartiles, so I left the list_price as is at this point.

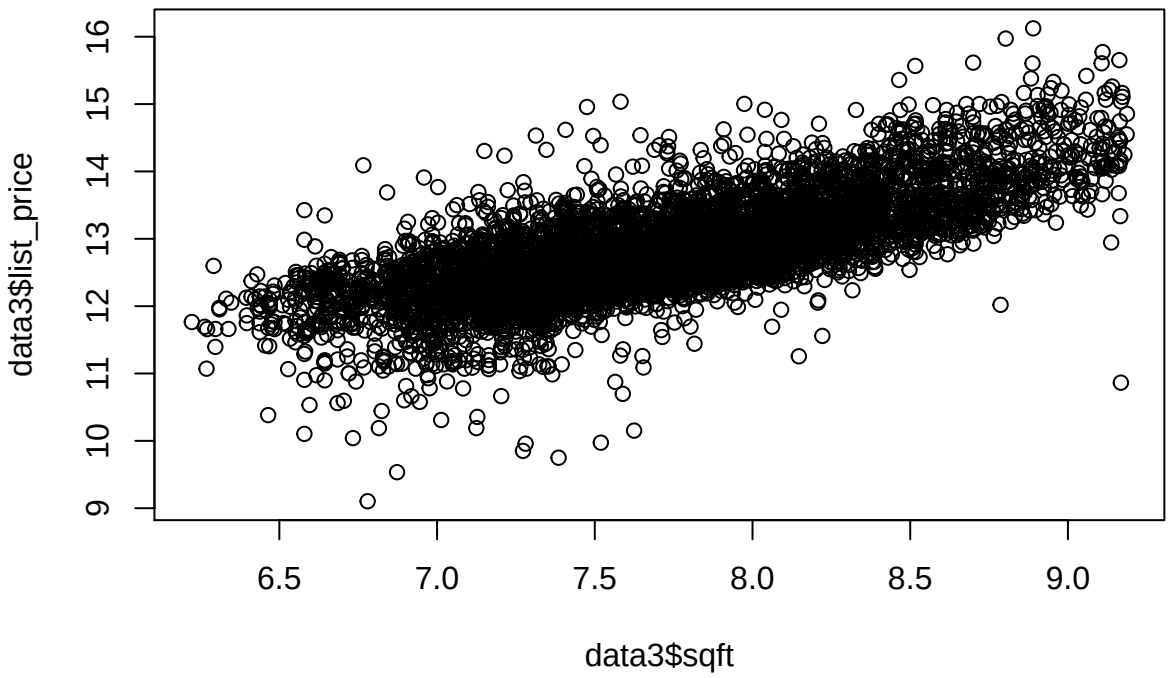
2. For sqft variable, I first plotted it against list_price. The data was heteroskedastic, so I took the log of the sqft variable, and then eliminated some outliers using the interquartile range. When I plotted this against list_price, the data was linear with constant variance and approximately gaussian error. The same was done for the lotsize variable as well.

3. I removed bed and bath number outliers using the iqr and quartiles. Plotting these against list_price checked out against the assumptions as well.

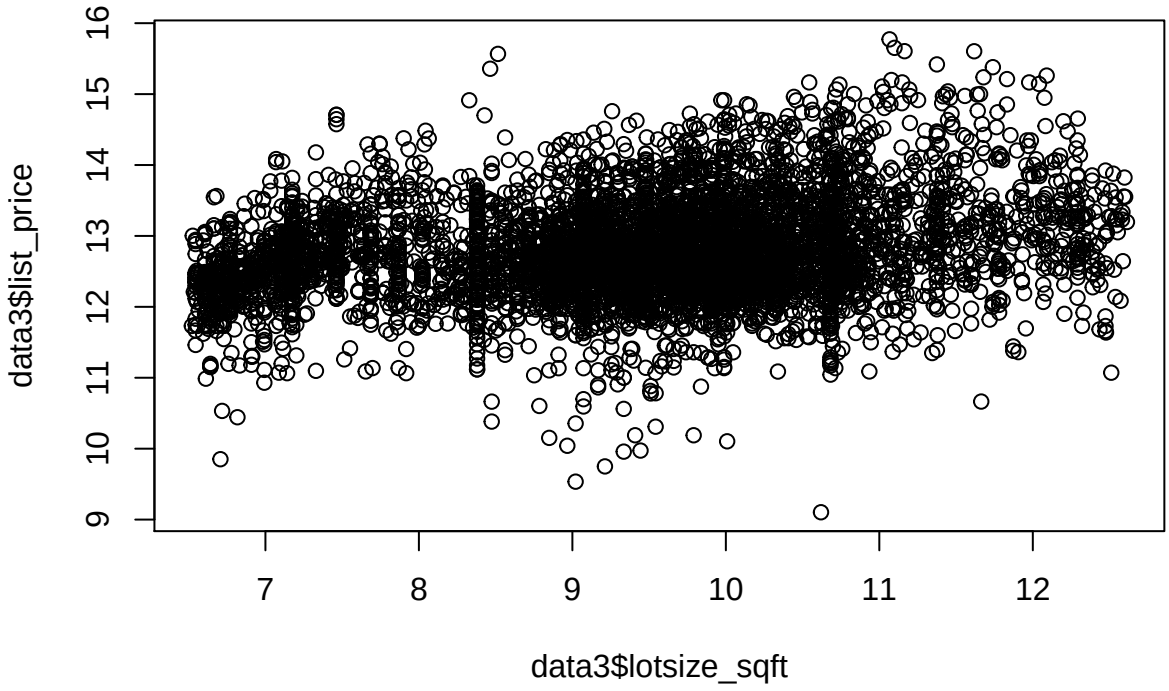
4. I calculated the table of occurrences for year_built, and there were a lot of singular values before 1950, so I eliminated those. Plotting it seems checks out against the assumptions as well, although it is hard to tell if there is indeed a significant relationship since the trend line seems very flat. Lasso regression can help decide the significance of this variable for us, so no more work is needed to be done here.

5. lat/lng analysis: First I realized that the locations of the properties were unimodal, with extreme outliers (by plotting histograms). This prompted me to frame location in terms of distance from the center of population. I then used the haversine formula to find distance between two places of different latitude and longitide coordinates. Then I eliminated for outliers above the third quartile using 3/2 times interquartile range plus the third quartile value. Plotting it seems checks out against the assumptions as well, although it is hard to tell if there is indeed a significant relationship since the trend line seems very flat. Lasso regression can help decide the significance of this variable for us, so no more work is needed to be done here.

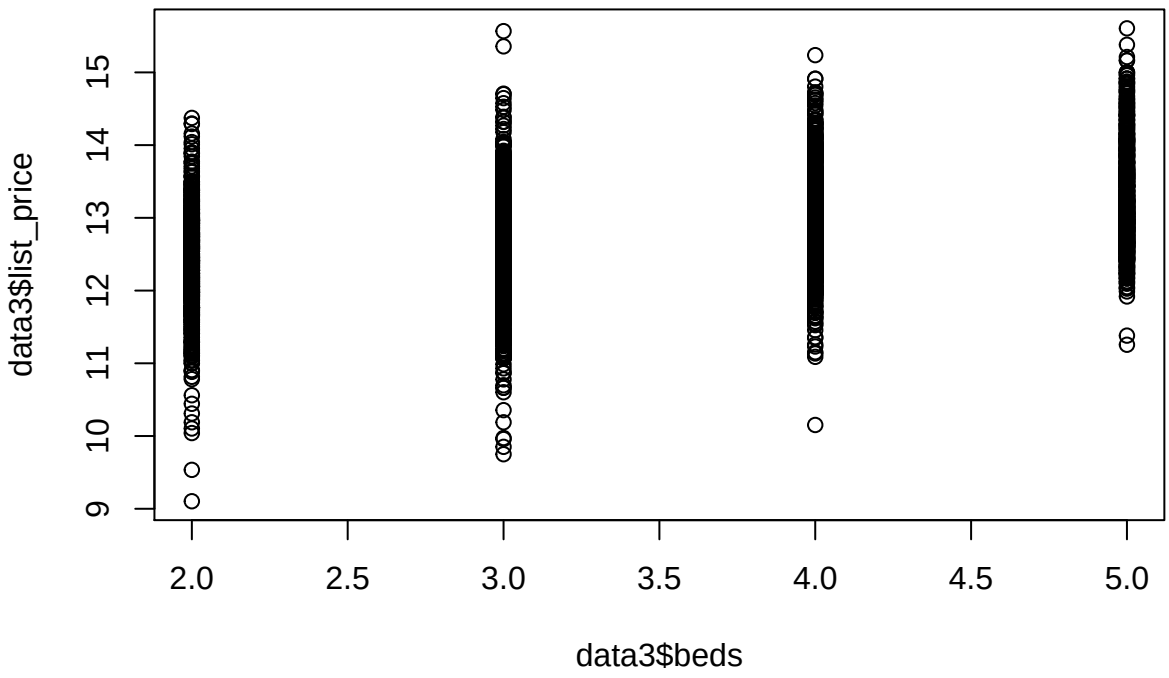
```
data3<- data2

data3$listing_date <- substr(data3$listing_date, 1, 7)
#table(data$listing_date)
caseshiller$DATE <- substr(caseshiller$DATE, 1, 7)
caseshiller$CSUSHPISA <- caseshiller$CSUSHPISA/214.459
for(i in 1:nrow(data3)){
  data3$list_price[i] <- data3$list_price[i]/(caseshiller[which(caseshiller$DATE == data3$listing_date[
}
data3<- subset(data3, select= -c(listing_date))
#hist(data2$list_price, breaks=50) #this is heavily skewed right with larger values being exponentially
vec <-log(data3$list_price)
#hist(vec, breaks=50) #this is better, albeit slightly right skewed
data3$list_price <- log(data3$list_price)
#summary(data2$list_price)
#hist(data3$list_price, breaks=75) #all checks out at this point

#sqft
#plot(data3$sqft, data3$list_price)
data3$sqft<- log(data3$sqft)
#hist(data3$sqft, breaks=50)
#summary(data3$sqft)
data3<- data3[!(data3$sqft<7.325-(8.070-7.325)*3/2 | data3$sqft>8.070+(8.070-7.325)*3/2),]
#hist(data3$sqft, breaks=50)
plot(data3$sqft, data3$list_price)
```



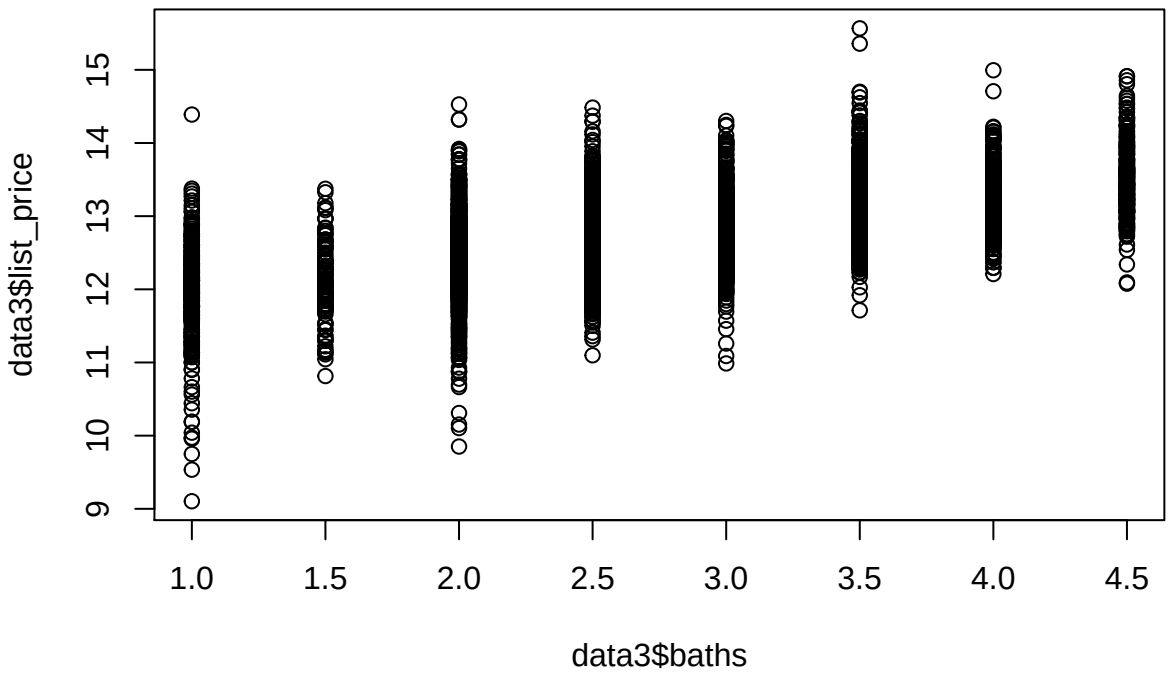

```
#lotsize
#plot(data3$lotsize_sqft, data3$list_price)
data3$lotsize_sqft<- log(data3$lotsize_sqft)
#plot(data3$lotsize_sqft, data3$list_price)
#summary(data3$lotsize_sqft)
data3<- data3[!(data3$lotsize_sqft<8.798-(10.325-8.798)*3/2 | data3$lotsize_sqft>10.325+(10.325-8.798)*3
plot(data3$lotsize_sqft, data3$list_price)
```



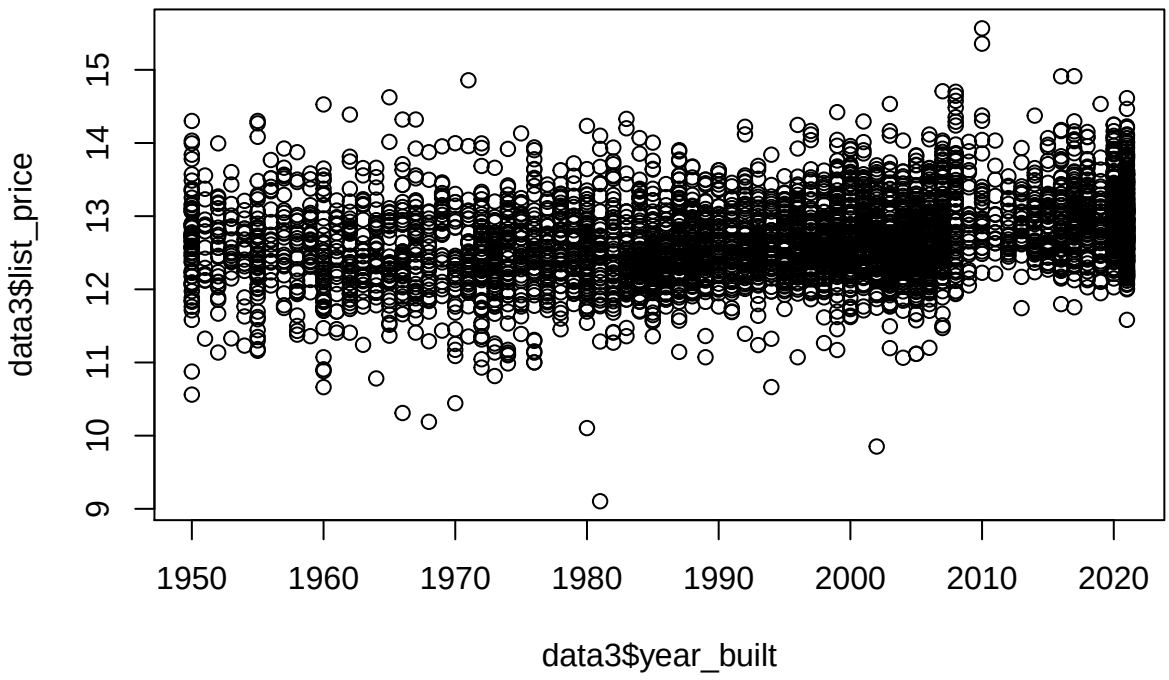

```
#bed and bath data cleaning
#table(data3$beds)
#summary(data3$beds)
data3<- data3[!(data3$beds<2 | data3$beds>5),]
#table(data3$baths)
plot(data3$beds, data3$list_price)
```



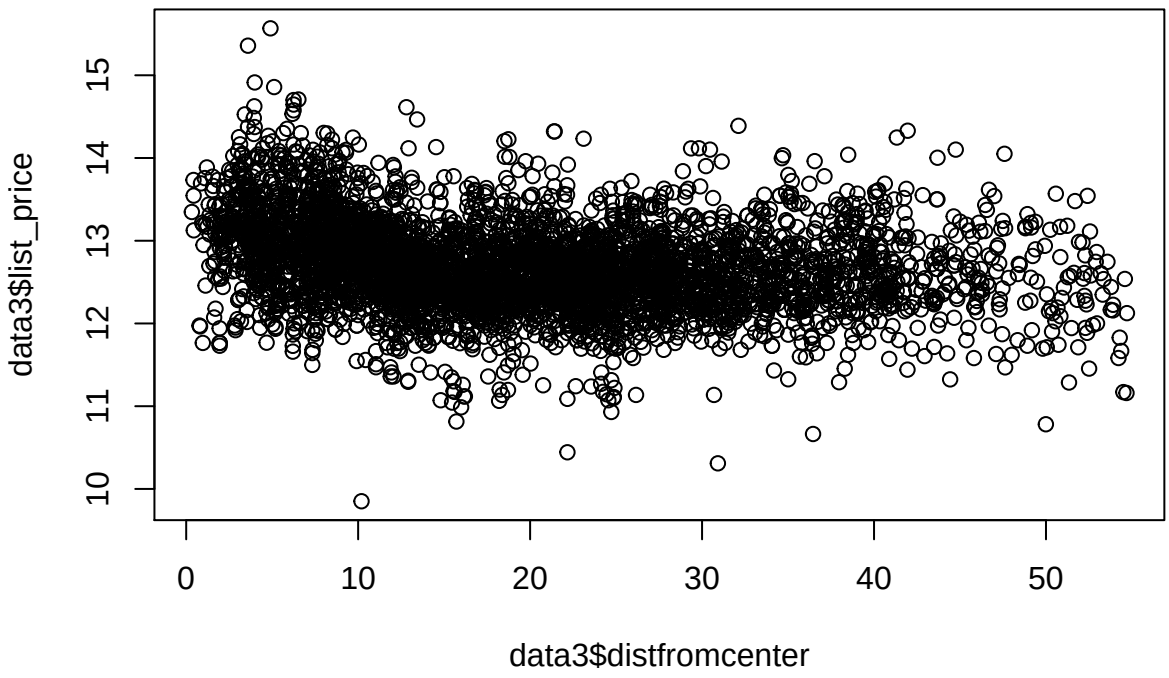

```
data3<- data3[!(data3$baths<1 | data3$baths>4.5),]
#table(data3$baths)
plot(data3$baths, data3$list_price)
```




```
#year_built data cleaning
#table(data3$year_built)
data3<- data3[!(data3$year_built<1950 | data3$year_built>2021),]
plot(data3$year_built, data3$list_price)
```




```
#summary(data3$lat) #there are some significant outliers
#summary(data3$lng) #there are some significant outliers
#hist(data3$lat, breaks=500) #unimodal
#hist(data3$lng, breaks=500) #unimodal
latmean = mean(data3$lat)
lngmean = mean(data3$lng)
latmean <- latmean/(180/pi)
lngmean <- lngmean/(180/pi)
distfromcenter<- vector()
for(i in 1:nrow(data3)){
  val = 3963*acos(sin(latmean)*sin(data3$lat[i]/(180/pi))+cos(latmean)*cos(data3$lat[i]/(180/pi))*cos(da
  distfromcenter<- c(distfromcenter, val)
}
data3$distfromcenter <- distfromcenter

#summary(data3$distfromcenter)
data3<- data3[!(data3$distfromcenter>28.310+1.5*(28.310-10.707)),]
data3<- subset(data3, select= -c(lat, lng))
plot(data3$distfromcenter , data3$list_price) #can't tell if it's linear, so I'll use lasso regression
```




```
#data3
```

##Checking for Multicollinearity

We first turn the property_type into a numerical binary variable since that is necessary for lasso regression. Then we plot the correlation matrix.

It is concerning that nearly all pairs of variables have a absolute value of correlation above 0.3, which is relatively nonsignificant. This is an issue. However, lasso should still be able to provide a rough estimate on which features are more relevant than others, so we proceed.

The good news is that list_price seems to has absolute value correlations for each indicator except with the last three predictor variables 0.3 and above (excluding lotsize_sqft), so our predictor variables for the most part are correlated with our outcome variable.

```
data4<- data3

data4$condo = as.numeric((data4$property_type=="Condominium"))
data4$townhouse = as.numeric((data4$property_type=="Townhouse"))
data4$singlefamilyhome = as.numeric((data4$property_type=="Single Family Residence"))

data4<- subset(data4, select= -c(property_type))
#data4

cor(data4[1:10])
```

```
##                   list_price        sqft lotsize_sqft year_built       beds
```

```
## list_price        1.000000000  0.66935550   0.06936058  0.2674058  0.3481464
## sqft              0.669355500  1.00000000   0.26409786  0.3510881  0.6779733
## lotsize_sqft      0.069360581  0.26409786   1.00000000 -0.1913633  0.3634095
## year_built        0.267405758  0.35108813  -0.19136331  1.0000000  0.1927608
## beds              0.348146418  0.67797328   0.36340954  0.1927608  1.0000000
## baths             0.599684416  0.75663992   0.08681612  0.4139824  0.6302619
## distfromcenter   -0.268651039  0.05734365   0.43099042  0.2440135  0.1465578
## condo            -0.003633554 -0.25886603  -0.45968454 -0.0118908 -0.4279966
## townhouse        -0.024320211 -0.06092168  -0.45656201  0.1913946 -0.2011251
## singlefamilyhome  0.021575116  0.23160802   0.68333456 -0.1410089  0.4613332
##                        baths distfromcenter        condo   townhouse
## list_price        0.59968442    -0.26865104 -0.003633554 -0.02432021
## sqft              0.75663992     0.05734365 -0.258866029 -0.06092168
## lotsize_sqft      0.08681612     0.43099042 -0.459684542 -0.45656201
## year_built        0.41398244     0.24401349 -0.011890805  0.19139458
## beds              0.63026190     0.14655783 -0.427996554 -0.20112511
## baths             1.00000000    -0.01950276 -0.198596140  0.09621970
## distfromcenter   -0.01950276     1.00000000 -0.311048851 -0.19416792
## condo            -0.19859614    -0.31104885  1.000000000 -0.10378265
## townhouse         0.09621970    -0.19416792 -0.103782648  1.00000000
## singlefamilyhome  0.06604725     0.37275951 -0.629876882 -0.70713055
##                  singlefamilyhome
## list_price             0.02157512
## sqft                   0.23160802
## lotsize_sqft           0.68333456
## year_built            -0.14100887
## beds                   0.46133316
## baths                  0.06604725
## distfromcenter         0.37275951
## condo                 -0.62987688
## townhouse             -0.70713055
## singlefamilyhome       1.00000000
```

##Lasso Regression

I first split the data set into a 20/80 split for testing vs training data. I then found the a good lambda, and fit a model on the training data using that lambda.

The model had a r^2 value of 0.59, which is well above the accepted industry standards of 0.30 for model adequacy. The model on the testing data had a r^2 value of 0.60, which is also satisfactory, and suggests that overfitting was not a significant issue.

In terms of the significance of our coefficients, square feet is by far the most important predictor variable for listing price, followed by the number of baths it has, whether or not it is a townhouse, the number of beds it has, whether or not it is a condo, the lotsize, distance from the center, and then the year it was built.

Intuitively, this makes sense, since the size of a building should be most important, followed by the rooms inside of it and the type of living space it is. Interestingly, the year a building was built does not impact listing price as much according to our model. Neither does the distance a house is away from the center of all of the listed houses in this data set. There are too many confounding variables for the distance a house is away from the center of our data to gain much information from that, but for the year a building was built, perhaps that might be due rennovations that we did not account for.

```
data5<- data4
#data4
set.seed(33)
```

```
rows<- sample(nrow(data5))
data5$rows<- rows
data5<- data5[order(rows),]
data5<- subset(data5, select= -c(rows))

test<- data5[1:983, ]
train<- data5[984:4916, ]

testfeatures<- as.matrix(test[2:10])
testoutcome<- as.matrix(test[1])

trainfeatures<- as.matrix(train[2:10])
trainoutcome<- as.matrix(train[1])

lambdas <- 10^seq(2, -2, by =-0.01)

lasso_reg <- cv.glmnet(trainfeatures, trainoutcome, alpha = 1, lambda = lambdas, standardize = TRUE, nf
lambda_best <- lasso_reg$lambda.min
#lambda_best

lasso_model <- glmnet(trainfeatures, trainoutcome, alpha = 1, lambda = lambda_best, standardize = TRUE)
predictions_train <- predict(lasso_model, s = lambda_best, newx = trainfeatures)
R_square_model <- 1 - sum((predictions_train - trainoutcome)^2)/sum((trainoutcome - mean(trainoutcome))
R_square_model
```

```
## [1] 0.5905459
```

```
predictions_test <- predict(lasso_model, s = lambda_best, newx = testfeatures)
R_square_model <- 1 - sum((predictions_test - testoutcome)^2)/sum((testoutcome - mean(testoutcome))^2)
R_square_model
```

```
## [1] 0.6069025
```

```
coef(lasso_model)
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                              s0
## (Intercept)       2.181144965
## sqft              0.717163047
## lotsize_sqft      0.034372029
## year_built        0.002460154
## beds             -0.081368118
## baths             0.132096102
## distfromcenter   -0.014470803
## condo             0.058040389
## townhouse        -0.105542441
## singlefamilyhome  .
```

##Citations

S&P Dow Jones Indices LLC, S&P/Case-Shiller U.S. National Home Price Index [CSUSHPISA], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CSUSHPISA, December 1, 2021.

Wikipedia contributors. "Haversine formula." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 9 Nov. 2021. Web. 3 Dec. 2021.