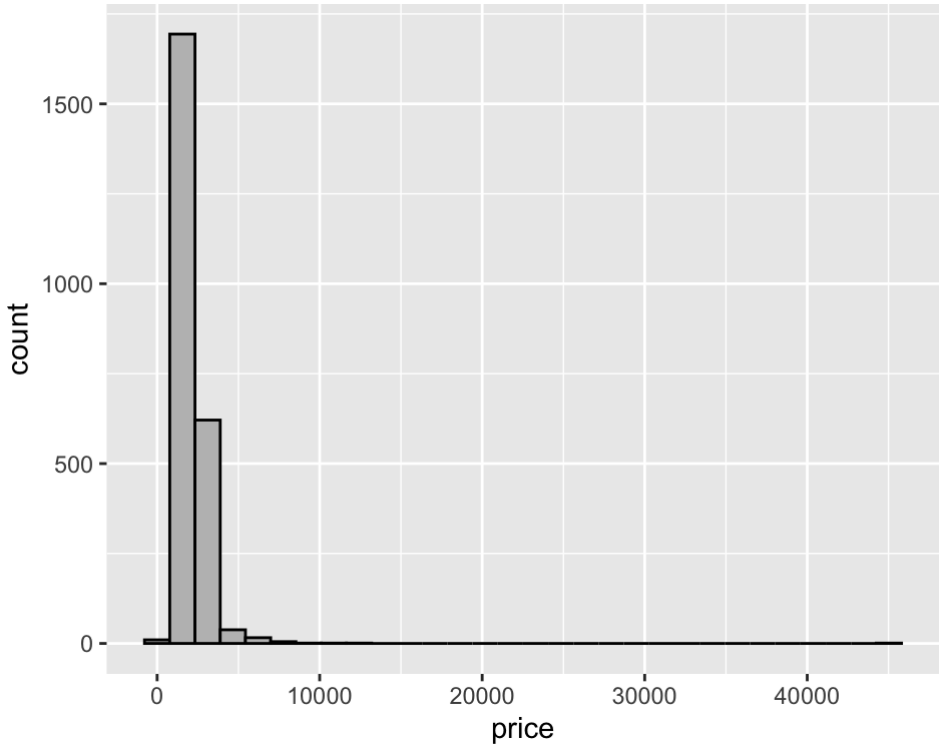
Rent Price Prediction Model

MGSC-310 Final Project

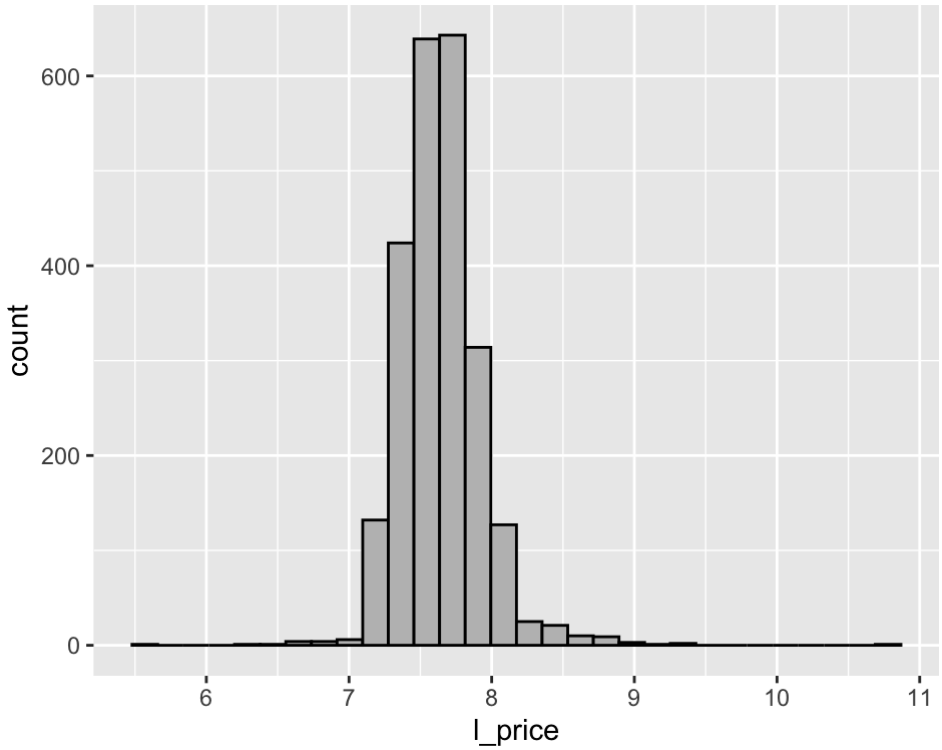
Will Strauss

Cleaning Dataset

```
library("tidyverse")
house <- read.csv("/Users/willstrauss/Documents/MGSC_310/datasets/OC_Rent_Address.csv")
# house <- read.csv(here::here('datasets', 'OC_Rent_Address.csv'))

# distribution of price before cleaning
ggplot(house, aes(price)) + geom_histogram(col = "black", fill = "grey")
```



```
house <- house %>% select(-id, -url, -region, -region_url, -parking_options,
    -image_url, -description, -long, -lat, -state, -AUTO_UNIQUE_ID_2020.12.07_17tonksmig
mailcom_OC_Housing,
    -UpdatedReverseGeocoding, -TimeTaken, -TransactionId, -Source, -ErrorMessage,
    -Version, -ComputedStreetAddress, -ComputedState, -ComputedZipPlus4,
    -ComputedAPN, -laundry_options) %>% mutate(type = as.factor(type),
    comes_furnished = as.factor(comes_furnished), ComputedCity = as.factor(ComputedCit
y),
    ComputedZip = as.factor(ComputedZip), wheelchair_access = as.factor(wheelchair_acces
s),
    smoking_allowed = as.factor(smoking_allowed), dogs_allowed = as.factor(dogs_allowe
d),
    cats_allowed = as.factor(cats_allowed), electric_vehicle_charge = as.factor(electric
_vehicle_charge),
    l_price = log(price + 1)) %>% filter(l_price > 0) %>% drop_na()

# distribution of price after cleaning
ggplot(house, aes(l_price)) + geom_histogram(col = "black", fill = "grey")
```



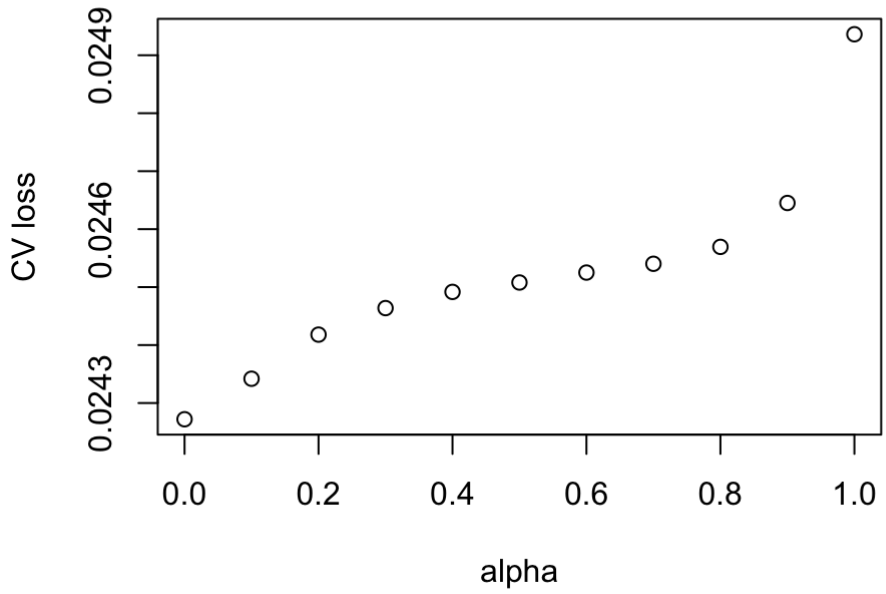
```
glimpse(house)
## Rows: 2,368
## Columns: 14
## $ price                   <int> 2310, 3000, 1805, 2750, 1695, 1620,…
## $ type                    <fct> apartment, apartment, apartment, ap…
## $ sqfeet                  <int> 1005, 1281, 717, 1288, 525, 417, 69…
## $ beds                    <int> 2, 2, 1, 3, 1, 0, 1, 1, 2, 2, 1, 3,…
## $ baths                   <dbl> 2.0, 2.0, 1.0, 2.0, 1.0, 1.0, 1.0, …
## $ cats_allowed            <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1,…
## $ dogs_allowed            <fct> 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1,…
## $ smoking_allowed         <fct> 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1,…
## $ wheelchair_access       <fct> 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,…
## $ electric_vehicle_charge <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
## $ comes_furnished         <fct> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,…
## $ ComputedCity            <fct> Buena Park, irvine, COSTA MESA, hun…
## $ ComputedZip             <fct> 90638, 92614, 92626, 92647, 92630, …
## $ l_price                 <dbl> 7.745436, 8.006701, 7.498870, 7.919…
```

Splitting the training and testing data

```
library("rsample")

houses = initial_split(house, 0.8)

houses_train = training(houses)
houses_test = testing(houses)
```

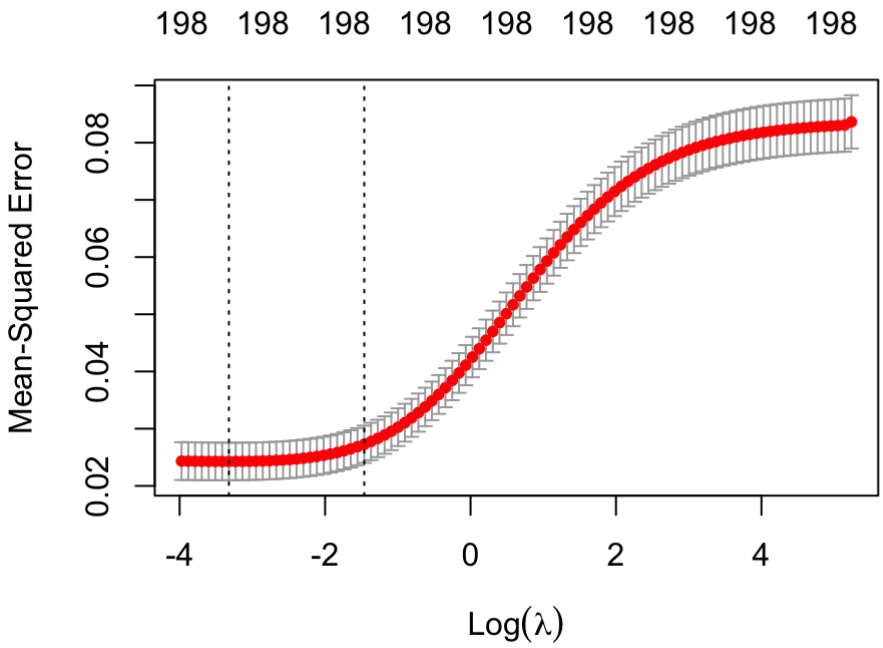
Building the elastic net model and plotting the min-loss

```
library("glmnet")
library("glmnetUtils")
library("forcats")
library("broom")

alpha_list = seq(0, 1, 0.1)
enet <- cva.glmnet(l_price ~ . - price, data = house, alpha = alpha_list)

minlossplot(enet, cv.type = "min")
```



Finding the best alpha value for the elastic net model

```
get_alpha <- function(fit) {
    alpha <- fit$alpha
    error <- sapply(fit$modlist, function(mod) {
        min(mod$cvm)
    })
    alpha[which.min(error)]
}
get_model_params <- function(fit) {
    alpha <- fit$alpha
    lambdaMin <- sapply(fit$modlist, `[[`, "lambda.min")
    lambdaSE <- sapply(fit$modlist, `[[`, "lambda.1se")
    error <- sapply(fit$modlist, function(mod) {
        min(mod$cvm)
    })
    best <- which.min(error)
    data.frame(alpha = alpha[best], lambdaMin = lambdaMin[best], lambdaSE = lambdaSE[bes
t],
        eror = error[best])
}

best_alpha <- get_alpha(enet)
best_mod <- enet$modlist[[which(enet$alpha == best_alpha)]]

plot(best_mod)
```



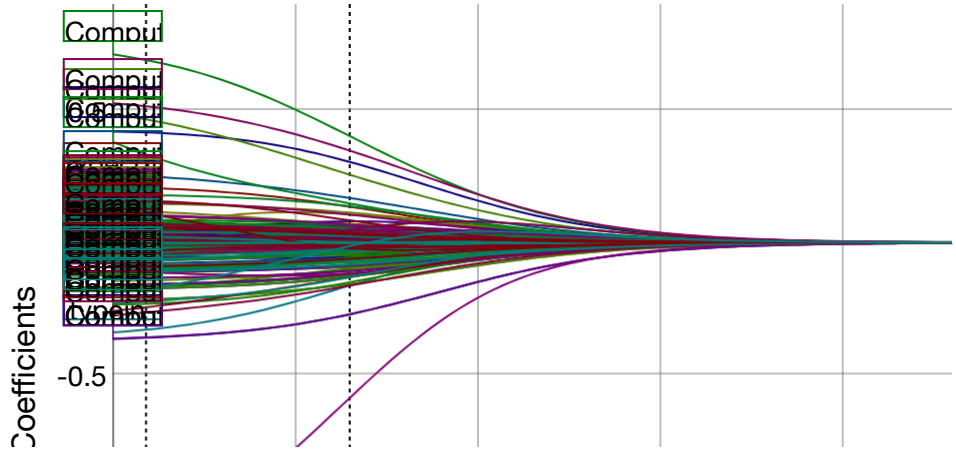

```
best_alpha <- get_alpha(enet)
print(best_alpha)
## [1] 0
get_model_params(enet)
```

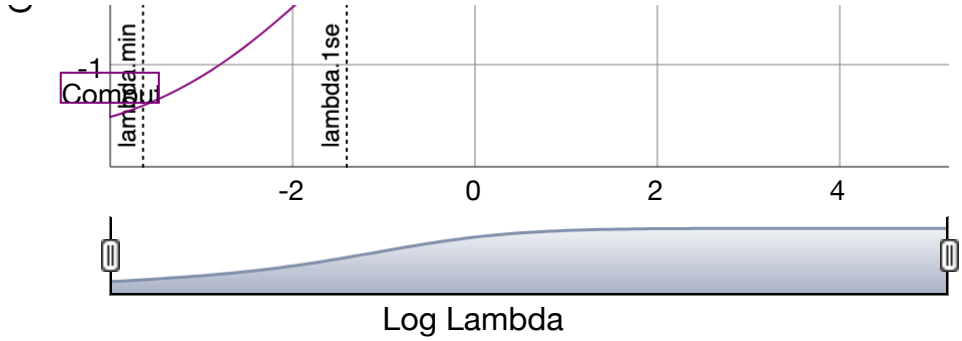
alpha	lambdaMin	lambdaSE	eror
<dbl>	<dbl>	<dbl>	<dbl>
0	0.03607086	0.2318661	0.02427203

1 row

Using toe coefplot library to visualize the coefficients

```
library("coefplot")
ridge = cv.glmnet(l_price ~ . - price, data = houses_train, alpha = best_alpha)
coefpath(ridge)
```





getting the model coefficients

```
lasso_coefs <- data.frame(ridge_min = coef(ridge, s = ridge$lambda.min) %>%
    as.matrix() %>% data.frame() %>% round(3), ridge_1se = coef(ridge,
    s = ridge$lambda.1se) %>% as.matrix() %>% data.frame() %>% round(3)) %>%
    rename(ridge_min = 1, ridge_1se = 2)

print(lasso_coefs)
##                                ridge_min ridge_1se
## (Intercept)                        7.308     7.436
## typeapartment                     -0.002    -0.038
## typecondo                          0.012     0.005
## typecottage/cabin                  0.207     0.136
## typeduplex                        -0.026    -0.009
## typehouse                         -0.060     0.033
## typein-law                        -0.328    -0.161
## typeloft                           0.069     0.044
## typemanufactured                   0.000     0.000
## typetownhouse                      0.050     0.054
## sqfeet                             0.000     0.000
## beds                               0.075     0.054
## baths                              0.064     0.065
## cats_allowed0                      0.014    -0.003
## cats_allowed1                     -0.013     0.003
## dogs_allowed0                     -0.029    -0.014
## dogs_allowed1                      0.028     0.014
## smoking_allowed0                   0.015     0.013
## smoking_allowed1                  -0.015    -0.013
## wheelchair_access0                -0.009    -0.012
## wheelchair_access1                 0.009     0.012
## electric_vehicle_charge0          -0.042    -0.032
## electric_vehicle_charge1           0.042     0.032
## comes_furnished0                  -0.025    -0.022
## comes_furnished1                   0.025     0.022
## ComputedCity                       0.000     0.000
## ComputedCityALISO VIEJO           -0.024    -0.016
## ComputedCityANA                   -0.063    -0.040
## ComputedCityAnaheim               -0.010    -0.017
## ComputedCityANAHEIM               -0.010    -0.011
## ComputedCityBeverly Hills          0.251     0.169
## ComputedCityBRE                   -0.010    -0.009
## ComputedCityBREA                   0.025    -0.002
## ComputedCityBuena Park            -0.088    -0.057
## ComputedCityBUENA PARK            -0.078    -0.053
## ComputedCityCerritos               0.000     0.000
## ComputedCityChino                  0.017     0.006
## ComputedCityChino Hills           -0.056    -0.029
## ComputedCityClaremont             -0.026    -0.017
## ComputedCityCN                     0.000     0.000
## ComputedCityColton                -0.164    -0.118
## ComputedCityCorona                -0.089    -0.061
## ComputedCityCOSTA MESA             0.016     0.011
## ComputedCityCypress               -0.006     0.003
## ComputedCityCYPRESS               -0.012    -0.011
```

```
## ComputedCityDANA POINT             0.038     0.028
## ComputedCityEastvale              -0.109    -0.082
## ComputedCityFontana               -0.132    -0.106
## ComputedCityFOUNTAIN VALLEY        0.002    -0.003
## ComputedCityFULLERTON              0.002     0.001
## ComputedCityGARDEN GROVE          -0.044    -0.053
## ComputedCityHacienda Heights       0.000     0.000
## ComputedCityHawaiian Gardens      -0.033    -0.046
## ComputedCityHighgrove             -0.159    -0.120
## ComputedCityhuntington beach      -0.015    -0.008
## ComputedCityHUNTINGTON BEACH       0.101     0.072
## ComputedCityIndio                 -0.156    -0.117
## ComputedCityirvine                 0.052     0.046
## ComputedCityIRVINE                 0.054     0.052
## ComputedCityLa Habra              -0.109    -0.090
## ComputedCityLA HABRA              -0.088    -0.092
## ComputedCityLAGUNA BEACH           0.043     0.108
## ComputedCityLAGUNA HILLS           0.032     0.046
## ComputedCitylaguna niguel         -0.010    -0.007
## ComputedCityLAGUNA NIGUEL         -0.019    -0.010
## ComputedCityLAGUNA WOODS          -0.111    -0.115
## ComputedCityLake Arrowhead        -0.359    -0.270
## ComputedCitylake forest           -0.035    -0.038
## ComputedCityLAKE FOREST            0.004     0.004
## ComputedCityLake Riverside         0.000     0.000
## ComputedCityLong Beach             0.056     0.046
## ComputedCityLos Angeles           -0.215    -0.156
## ComputedCityMISSION VIEJO          0.038     0.021
## ComputedCitynewport beach          0.147     0.114
## ComputedCityNEWPORT BEACH          0.417     0.306
## ComputedCityNEWPORT COAST          0.000     0.000
## ComputedCityOntario               -0.067    -0.040
## ComputedCityorange                 0.118     0.067
## ComputedCityOrange                -0.016    -0.009
## ComputedCityORANGE                -0.042    -0.012
## ComputedCityPLA                   -0.005    -0.010
## ComputedCityPLACENTIA             -0.005     0.004
## ComputedCityRosemead              -0.015    -0.008
## ComputedCitySan Bernardino        -0.142    -0.103
## ComputedCitySAN CLEMENTE           0.020     0.030
## ComputedCitysan juan               0.021     0.012
## ComputedCitySAN JUAN CAPISTRANO    0.061     0.047
## ComputedCitySANTA ANA             -0.016    -0.020
## ComputedCitySANTA MARGARITA        0.025     0.013
## ComputedCitySD                     0.000     0.000
## ComputedCitySTANTON               -0.172    -0.128
## ComputedCityTRABUCO CANYON        -0.035     0.050
## ComputedCityTUSTIN                 0.049     0.018
## ComputedCityUpland                -0.089    -0.073
## ComputedCityWESTMINSTER           -0.022    -0.018
## ComputedCityYBL                    0.000     0.000
## ComputedCityYORBA LINDA            0.040     0.029
```

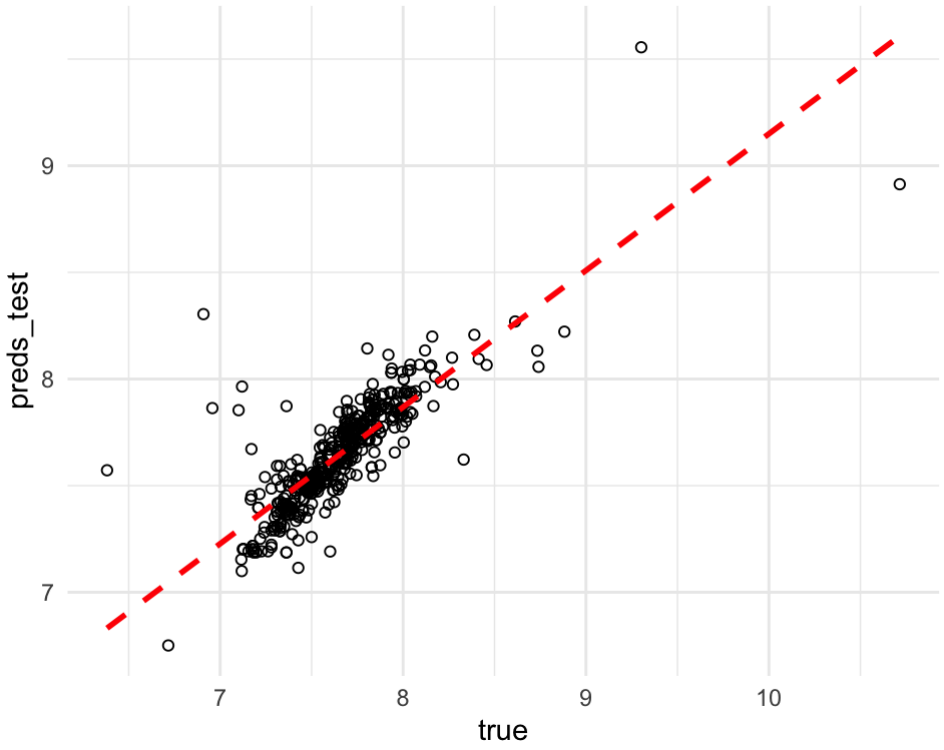
```
## ComputedZip90015          -0.214    -0.156
## ComputedZip90025           0.034     0.081
## ComputedZip90048           0.466     0.257
## ComputedZip90603          -0.109    -0.090
## ComputedZip90605           0.000     0.000
## ComputedZip90620          -0.021    -0.025
## ComputedZip90621          -0.168    -0.096
## ComputedZip90630          -0.012    -0.011
## ComputedZip90631          -0.088    -0.092
## ComputedZip90638          -0.088    -0.057
## ComputedZip90680          -0.172    -0.128
## ComputedZip90701           0.000     0.000
## ComputedZip90716          -0.005     0.003
## ComputedZip90803           0.062     0.042
## ComputedZip90808          -0.033    -0.046
## ComputedZip90814           0.035     0.062
## ComputedZip91709           0.017     0.006
## ComputedZip91710          -0.056    -0.029
## ComputedZip91730           0.000     0.000
## ComputedZip91750          -0.026    -0.017
## ComputedZip91764          -0.066    -0.040
## ComputedZip91770          -0.015    -0.008
## ComputedZip91786          -0.089    -0.073
## ComputedZip92301          -0.358    -0.270
## ComputedZip92324          -0.163    -0.118
## ComputedZip92335          -0.132    -0.106
## ComputedZip92354          -0.231    -0.149
## ComputedZip92410          -0.112    -0.087
## ComputedZip92557          -0.159    -0.120
## ComputedZip92562          -0.156    -0.117
## ComputedZip92592           0.000     0.000
## ComputedZip92602           0.076     0.056
## ComputedZip92603           0.066     0.048
## ComputedZip92604           0.005    -0.004
## ComputedZip92606           0.042     0.033
## ComputedZip92610           0.102     0.062
## ComputedZip92612           0.087     0.066
## ComputedZip92614           0.105     0.068
## ComputedZip92617          -0.089    -0.066
## ComputedZip92618           0.074     0.054
## ComputedZip92620          -0.102    -0.056
## ComputedZip92624          -1.148    -0.586
## ComputedZip92625           0.690     0.403
## ComputedZip92626           0.019     0.013
## ComputedZip92627           0.014     0.009
## ComputedZip92629           0.076     0.048
## ComputedZip92630          -0.002    -0.001
## ComputedZip92637          -0.112    -0.115
## ComputedZip92646           0.082     0.054
## ComputedZip92647          -0.028    -0.021
## ComputedZip92648           0.075     0.053
## ComputedZip92649          -0.009     0.008
```

```
## ComputedZip92651                0.334     0.149
## ComputedZip92653                0.033     0.046
## ComputedZip92656               -0.024    -0.016
## ComputedZip92657                0.154     0.106
## ComputedZip92660                0.180     0.138
## ComputedZip92662                0.516     0.348
## ComputedZip92663               -0.065    -0.017
## ComputedZip92672                0.034     0.032
## ComputedZip92673                0.039     0.040
## ComputedZip92675                0.060     0.046
## ComputedZip92677               -0.013    -0.008
## ComputedZip92678               -0.033     0.050
## ComputedZip92683               -0.022    -0.018
## ComputedZip92688                0.017     0.012
## ComputedZip92691                0.008     0.024
## ComputedZip92692               -0.028     0.000
## ComputedZip92694                0.004     0.027
## ComputedZip92701               -0.046    -0.029
## ComputedZip92703               -0.019    -0.013
## ComputedZip92704               -0.033    -0.019
## ComputedZip92705               -0.023    -0.005
## ComputedZip92706               -0.263    -0.162
## ComputedZip92707               -0.064    -0.034
## ComputedZip92708                0.002    -0.003
## ComputedZip92780               -0.040    -0.004
## ComputedZip92782                0.006     0.026
## ComputedZip92801               -0.099    -0.061
## ComputedZip92802                0.071     0.037
## ComputedZip92804               -0.085    -0.060
## ComputedZip92805                0.061     0.039
## ComputedZip92806                0.024     0.008
## ComputedZip92807                0.027     0.009
## ComputedZip92808                0.042     0.025
## ComputedZip92821               -0.007    -0.008
## ComputedZip92831               -0.105    -0.072
## ComputedZip92832                0.010     0.010
## ComputedZip92833               -0.049    -0.023
## ComputedZip92835                0.159     0.085
## ComputedZip92840               -0.067    -0.031
## ComputedZip92841               -0.244    -0.130
## ComputedZip92843                0.000     0.000
## ComputedZip92844               -0.033    -0.015
## ComputedZip92845               -0.222    -0.130
## ComputedZip92865                0.014    -0.013
## ComputedZip92866                0.000     0.000
## ComputedZip92867               -0.006     0.007
## ComputedZip92868                0.025     0.003
## ComputedZip92869                0.112    -0.021
## ComputedZip92870               -0.004     0.002
## ComputedZip92879               -0.089    -0.061
## ComputedZip92882               -0.109    -0.082
```

```
## ComputedZip92886                0.055      0.032
## ComputedZip92887               -0.173     -0.013
```

Predicting the testing data then plotting predicted values vs true values

```
preds_test = as.double(predict(ridge, newdata = na.omit(houses_test), s = ridge$lambda.m
in))
preds_train = as.double(predict(ridge, newdata = na.omit(houses_train),
    s = ridge$lambda.min))

results = data.frame(true = houses_test$l_price, pred = preds_test)

ggplot(results, aes(x = true, y = preds_test)) + geom_point(color = "black",
    shape = 1) + theme_minimal() + geom_smooth(method = "lm", se = FALSE,
    linetype = "dashed", col = "red")  #+ xlim(0, 10000) + ylim(0, 10000)
```



Calculating R2

```
rss <- sum((preds_train - houses_train$l_price)^2)  ## residual sum of squares
tss <- sum((houses_train$l_price - mean(houses_train$l_price))^2)  ## total sum of squar
es
ltrain_rsq <- 1 - rss/tss

rss <- sum((preds_test - houses_test$l_price)^2)  ## residual sum of squares
tss <- sum((houses_test$l_price - mean(houses_test$l_price))^2)  ## total sum of squares
ltest_rsq <- 1 - rss/tss

# Training R2
ltrain_rsq
## [1] 0.7718124
# Testing R2:
ltest_rsq
## [1] 0.6730513
```