

412 Project

Data

```
library(readr)
library(dplyr)
```

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

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

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

```
library(ggplot2)
library(DataExplorer)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(caTools)
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##   select
```

```
library(leaps)
library(caret)
```

```
## Loading required package: lattice
```

```
library(pcr)
library(pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:caret':
##
##   R2
```

```

## The following object is masked from 'package:stats':
##
##   loadings
library(Metrics)

##
## Attaching package: 'Metrics'
## The following objects are masked from 'package:caret':
##
##   precision, recall
library(dplyr)
library(randomForest)

## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##   margin
## The following object is masked from 'package:dplyr':
##
##   combine
library(data.table)

##
## Attaching package: 'data.table'
## The following objects are masked from 'package:lubridate':
##
##   hour, isoweek, mday, minute, month, quarter, second, wday, week,
##   yday, year
## The following objects are masked from 'package:dplyr':
##
##   between, first, last
library(leaps)
library(caTools)
library(randomForest)
library(glmnet) #cv.glmnet

## Loading required package: Matrix
## Loaded glmnet 4.1-3

```

Basic EDA

```

set.seed(1)

house <- read.csv('house.csv')
head(house)

```

```
##           id           date    price bedrooms bathrooms sqft_living sqft_lot
## 1 7129300520 20141013T000000 221900         3         1.00        1180    5650
## 2 6414100192 20141209T000000 538000         3         2.25        2570    7242
## 3 5631500400 20150225T000000 180000         2         1.00         770   10000
## 4 2487200875 20141209T000000 604000         4         3.00        1960    5000
## 5 1954400510 20150218T000000 510000         3         2.00        1680    8080
## 6 7237550310 20140512T000000 1225000        4         4.50        5420   101930
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0          3      7        1180          0    1955
## 2      2          0    0          3      7        2170         400    1951
## 3      1          0    0          3      6         770          0    1933
## 4      1          0    0          5      7        1050         910    1965
## 5      1          0    0          3      8        1680          0    1987
## 6      1          0    0          3     11        3890        1530    2001
##   yr_renovated zipcode      lat      long sqft_living15 sqft_lot15
## 1              0   98178 47.5112 -122.257         1340        5650
## 2             1991   98125 47.7210 -122.319         1690        7639
## 3              0   98028 47.7379 -122.233         2720        8062
## 4              0   98136 47.5208 -122.393         1360        5000
## 5              0   98074 47.6168 -122.045         1800        7503
## 6              0   98053 47.6561 -122.005         4760       101930
```

```
dim(house)
```

```
## [1] 21613    21
```

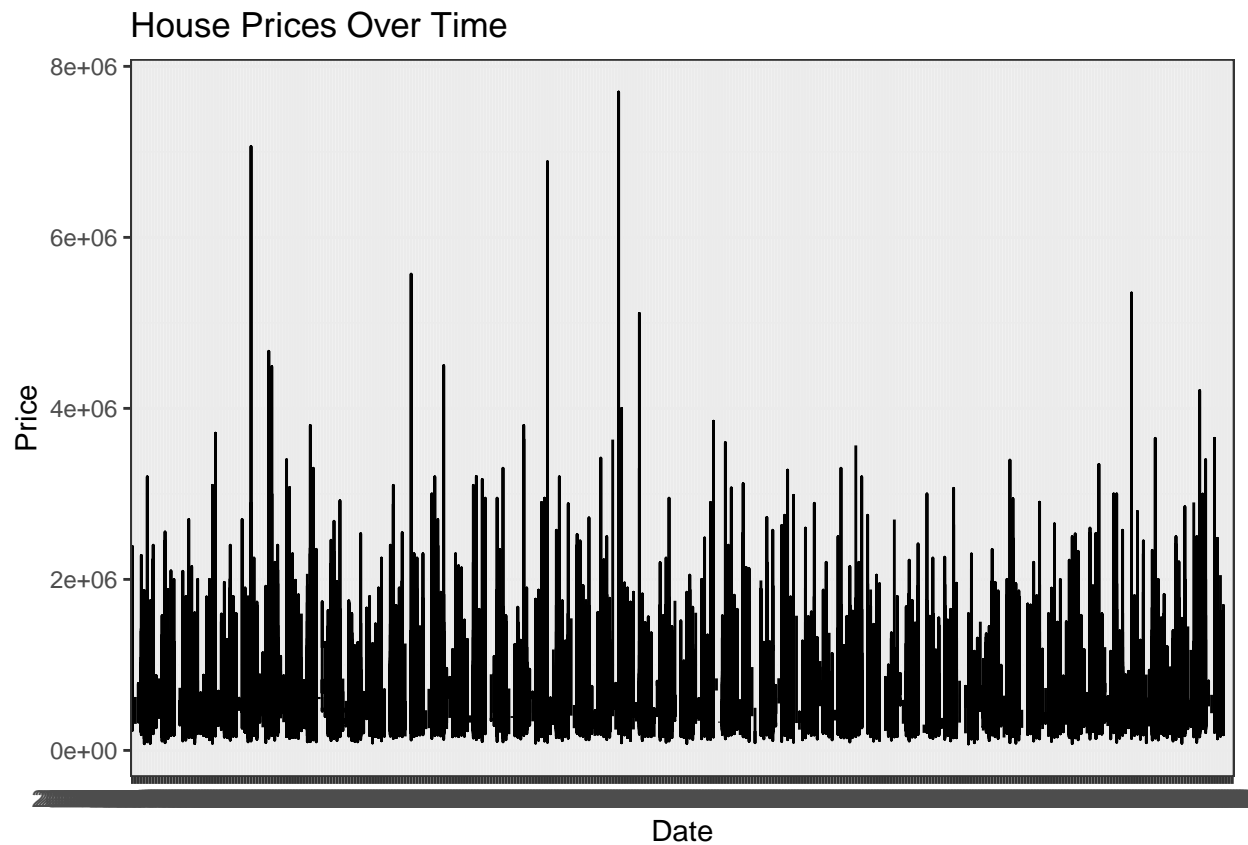
```
num_ob_bf_drop <- dim(house)[1]
```

```
# Add a feature if there is a basement then 1 else 0
```

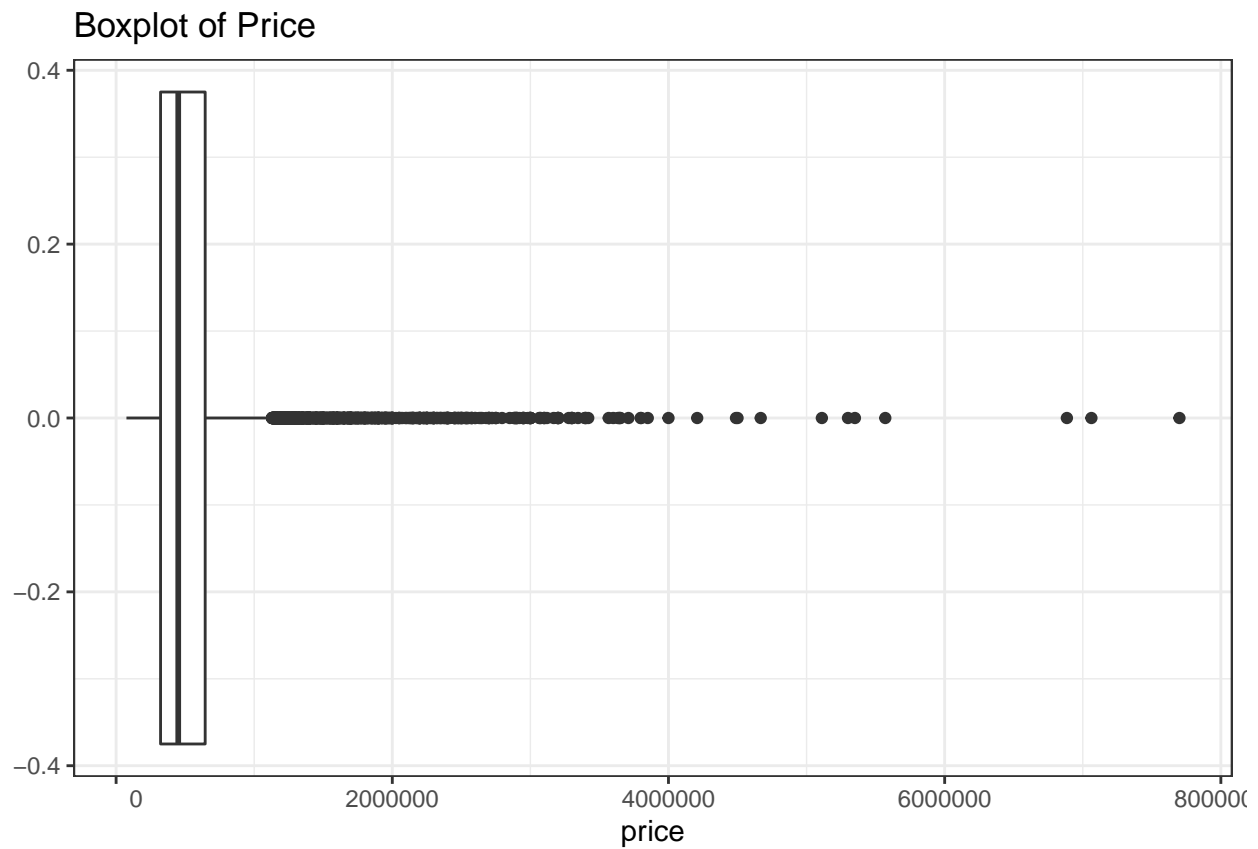
```
for(i in 1: nrow(house)){
  if (house$sqft_basement[i] >0) {
    house$sqft_basement_yesno[i] <- 1
  } else {
    house$sqft_basement_yesno[i] <- 0
  }
}
```

```
# Distribution of Date
```

```
ggplot(house, aes(x=date, y = price))+
  geom_line()+
  xlab('Date')+
  ylab('Price')+
  ggtitle('House Prices Over Time') +
  theme_bw()
```

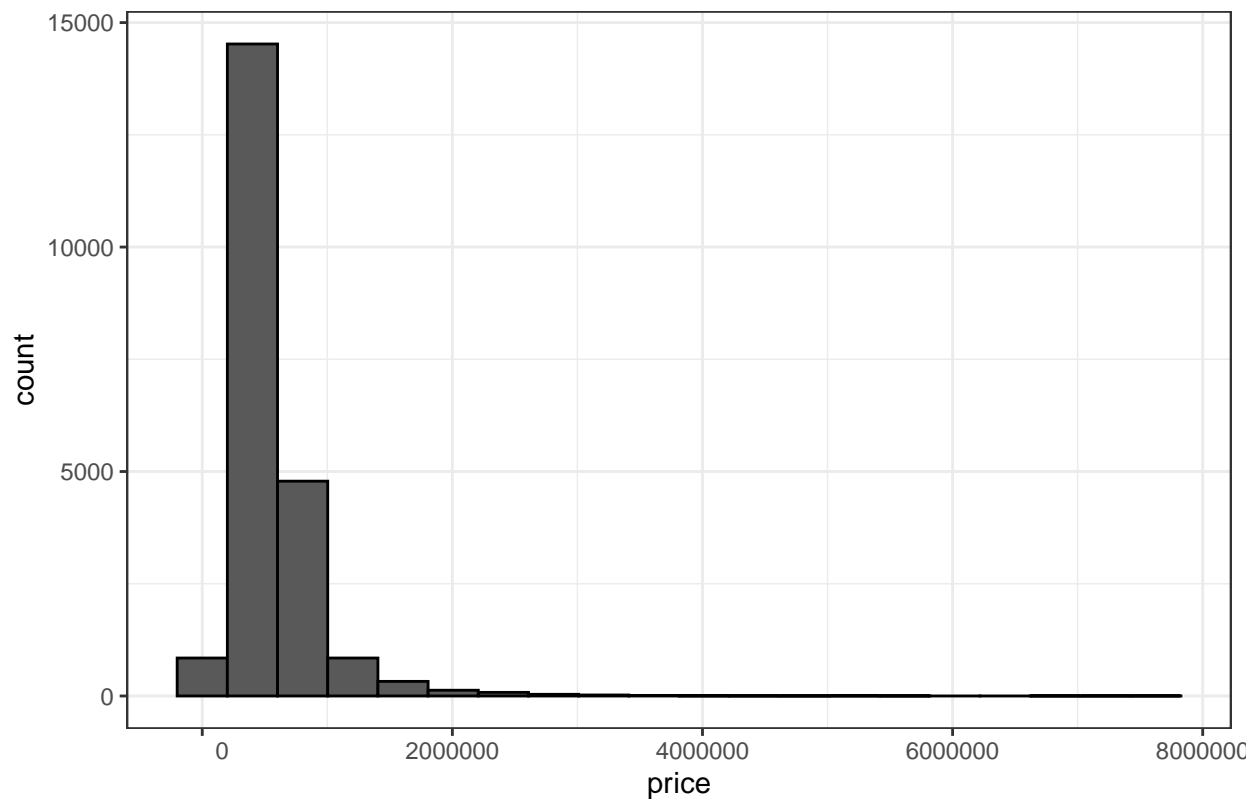


```
# Boxplot of prices
ggplot(house)+
  aes(x=price)+
  geom_boxplot() +
  ggtitle("Boxplot of Price") +
  theme_bw() +
  scale_x_continuous(labels = function(x) format(x, scientific = FALSE))
```



```
# Distribution of price
ggplot(house)+
  aes(x=price)+
  geom_histogram(col = 'black', bins = 20) +
  ggtitle("Distribution of Price (no transformation)") +
  theme_bw() +
  scale_x_continuous(labels = function(x) format(x, scientific = FALSE))
```

Distribution of Price (no transformation)



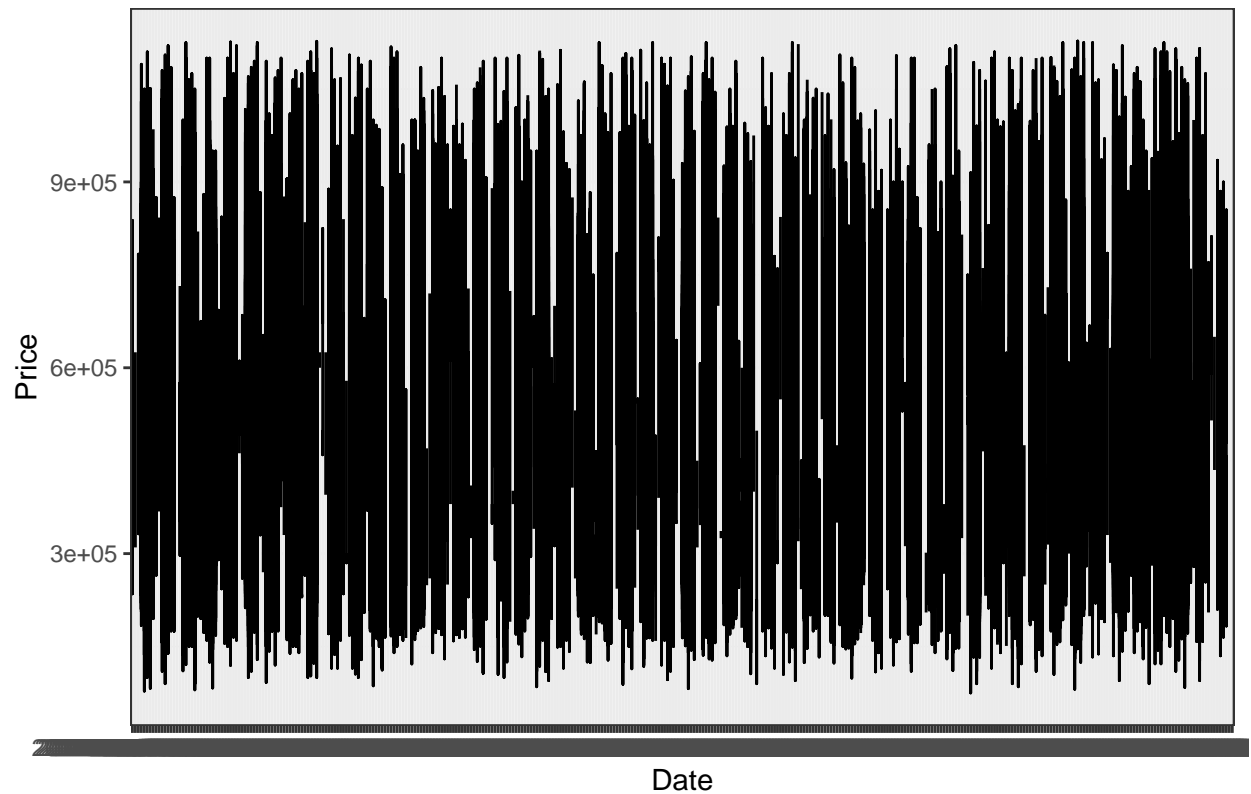
```
# Get rid of outliers (price-wise)
summary(house$price)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  75000  321950  450000  540088  645000 7700000
```

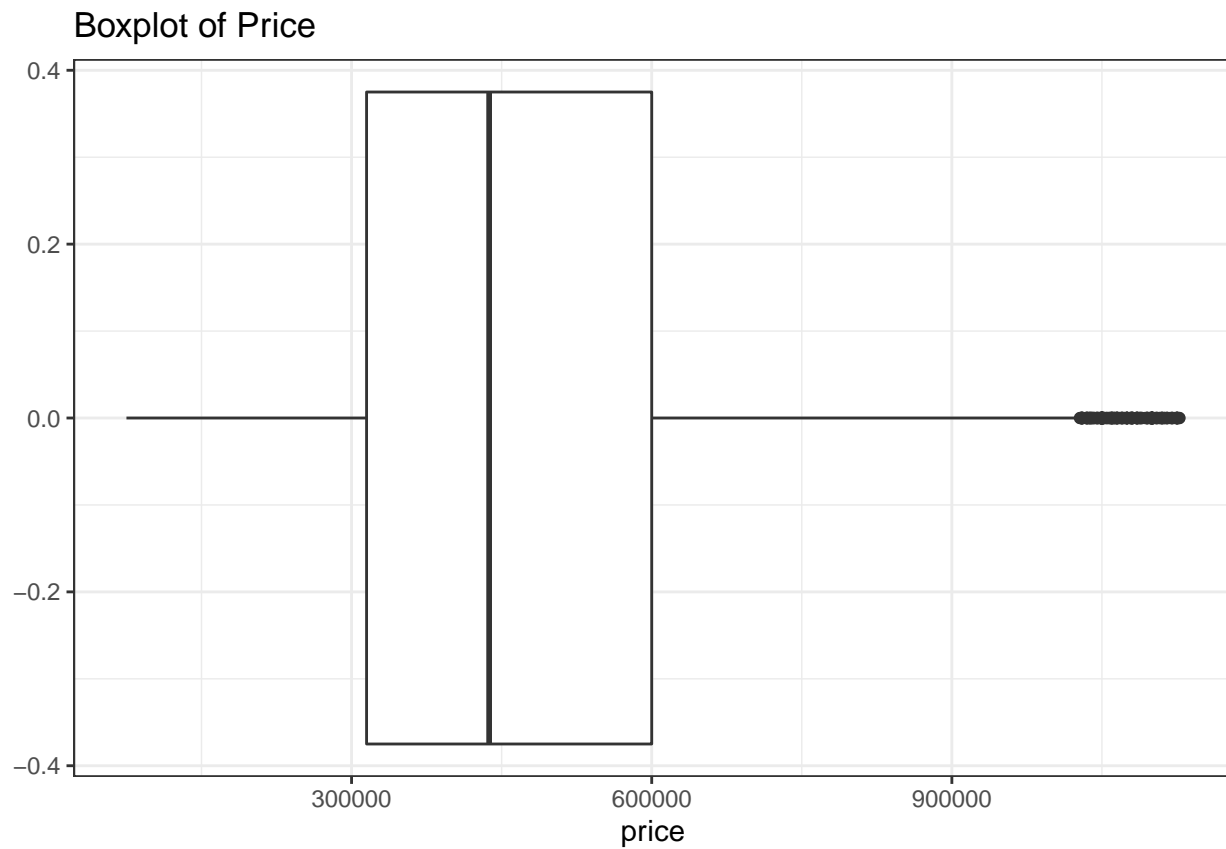
```
first_quartile <- summary(house$price)[[2]]
third_quartile <- summary(house$price)[[5]]
IQR <- third_quartile-first_quartile
Upper <- 1.5*IQR + third_quartile
Lower <- first_quartile - 1.5*IQR
house <- subset(house, price >= Lower & price <= Upper)
```

```
# Distribution of Date
ggplot(house, aes(x=date, y = price))+
  geom_line()+
  xlab('Date')+
  ylab('Price')+
  ggtitle('House Prices Over Time') +
  theme_bw()
```

House Prices Over Time

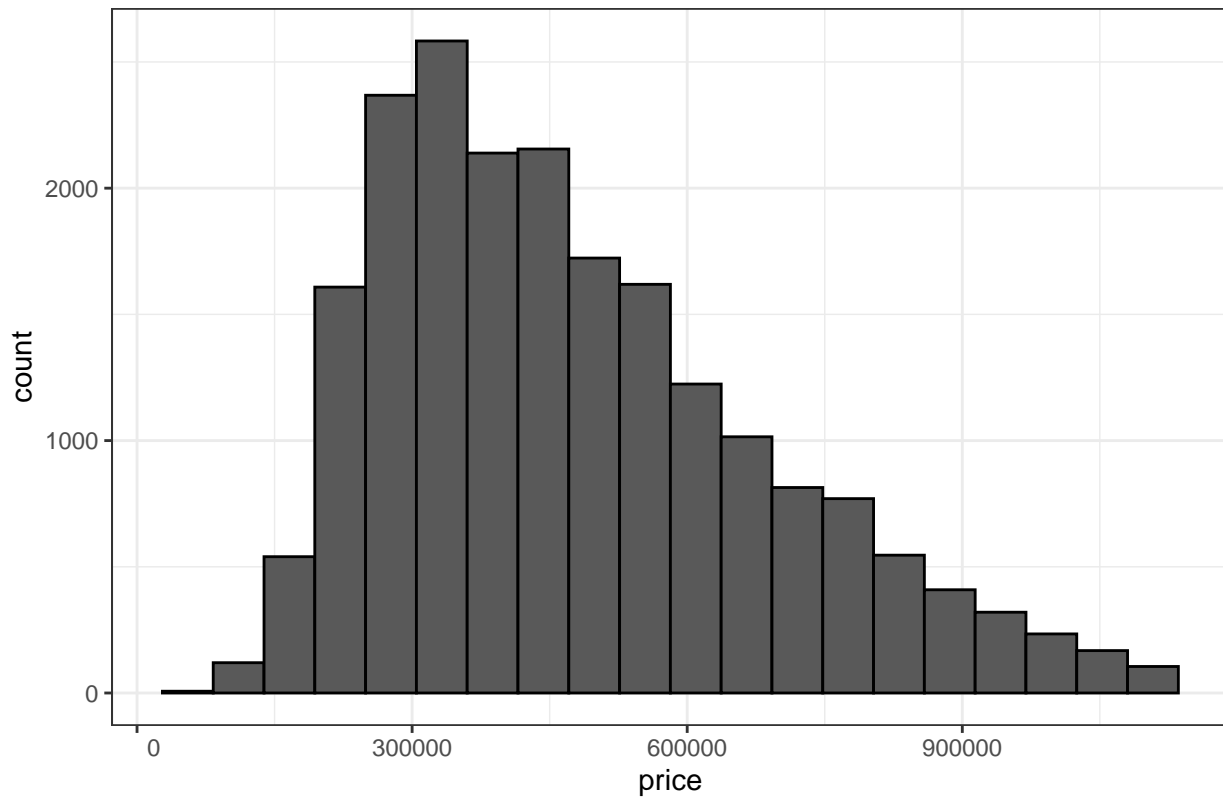


```
# Boxplot of prices
ggplot(house)+
  aes(x=price)+
  geom_boxplot() +
  ggtitle("Boxplot of Price") +
  theme_bw() +
  scale_x_continuous(labels = function(x) format(x, scientific = FALSE))
```



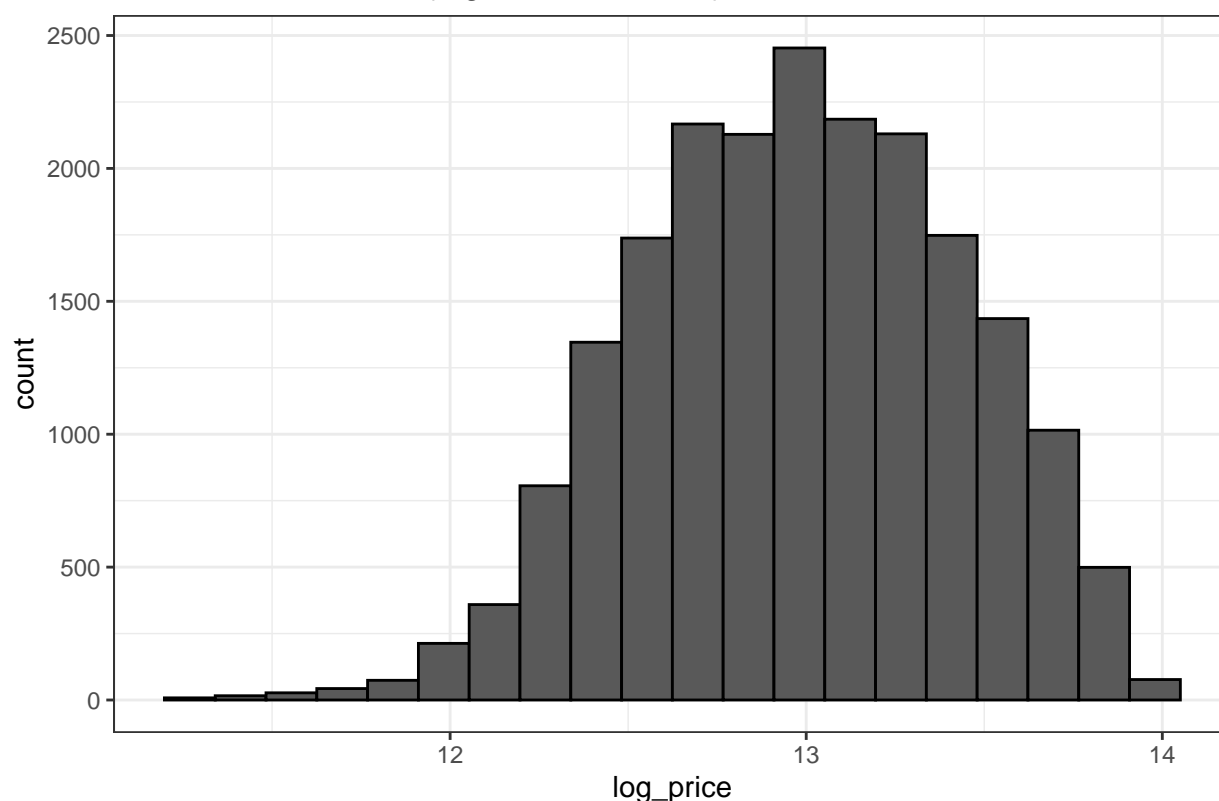
```
# Distribution of price
ggplot(house)+
  aes(x=price)+
  geom_histogram(col = 'black', bins = 20) +
  ggtitle("Distribution of Price (no transformation)") +
  theme_bw() +
  scale_x_continuous(labels = function(x) format(x, scientific = FALSE))
```


Distribution of Price (no transformation)



```
# Distribution of price using log transform
house$log_price <- log(house$price)
ggplot(house)+
  aes(x=log_price)+
  geom_histogram(col = 'black', bins = 20) +
  ggtitle("Distribution of Price (log transformation)") +
  theme_bw()
```

Distribution of Price (log transformation)



```
# Drop date: No relationship is detected
# Drop id: No meaning
# Drop zipcode: We have latitude and longitude info
# Drop sqft_basement: I have sqft_basement_yesno feature
drop <- c('date', 'id', 'zipcode', 'sqft_basement', 'log_price')
house <- house[!names(house) %in% drop]
summary(house)
```

```
##      price      bedrooms      bathrooms      sqft_living
## Min.   : 75000   Min.   : 0.00   Min.   :0.0000   Min.   : 290
## 1st Qu.: 315000 1st Qu.: 3.00   1st Qu.:1.500   1st Qu.:1400
## Median : 437500 Median : 3.00   Median :2.000   Median :1860
## Mean   : 476985 Mean   : 3.33   Mean   :2.052   Mean   :1976
## 3rd Qu.: 600000 3rd Qu.: 4.00   3rd Qu.:2.500   3rd Qu.:2431
## Max.   :1127500 Max.   :33.00   Max.   :7.500   Max.   :7480
##      sqft_lot      floors      waterfront      view
## Min.   : 520   Min.   :1.000   Min.   :0.00000   Min.   :0.0000
## 1st Qu.: 5000 1st Qu.:1.000   1st Qu.:0.00000   1st Qu.:0.0000
## Median : 7500 Median :1.000   Median :0.00000   Median :0.0000
## Mean   : 14610 Mean :1.476   Mean :0.00298   Mean :0.1727
## 3rd Qu.: 10319 3rd Qu.:2.000   3rd Qu.:0.00000   3rd Qu.:0.0000
## Max.   :1651359 Max. :3.500   Max. :1.00000   Max. :4.0000
##      condition      grade      sqft_above      yr_built
## Min.   :1.000   Min.   : 1.000   Min.   : 290   Min.   :1900
## 1st Qu.:3.000   1st Qu.: 7.000   1st Qu.:1170   1st Qu.:1951
## Median :3.000   Median : 7.000   Median :1520   Median :1974
## Mean   :3.406   Mean   : 7.531   Mean   :1708   Mean   :1971
```

```
## 3rd Qu.:4.000 3rd Qu.: 8.000 3rd Qu.:2100 3rd Qu.:1996
## Max. :5.000 Max. :12.000 Max. :5710 Max. :2015
## yr_renovated lat long sqft_living15
## Min. : 0.00 Min. :47.16 Min. :-122.5 Min. : 399
## 1st Qu.: 0.00 1st Qu.:47.46 1st Qu.: -122.3 1st Qu.:1470
## Median : 0.00 Median :47.57 Median : -122.2 Median :1800
## Mean : 74.68 Mean :47.56 Mean : -122.2 Mean :1922
## 3rd Qu.: 0.00 3rd Qu.:47.68 3rd Qu.: -122.1 3rd Qu.:2280
## Max. :2015.00 Max. :47.78 Max. : -121.3 Max. :5380
## sqft_lot15 sqft_basement_yesno
## Min. : 651 Min. :0.0000
## 1st Qu.: 5046 1st Qu.:0.0000
## Median : 7542 Median :0.0000
## Mean : 12447 Mean :0.3794
## 3rd Qu.: 9884 3rd Qu.:1.0000
## Max. :871200 Max. :1.0000
```

```
dim(house)
```

```
## [1] 20467 18
```

```
num_ob_af_drop <- dim(house)[1]
```

```
num_ob_af_drop/num_ob_bf_drop*100
```

```
## [1] 94.69764
```

```
#DataExplorer::create_report(house)
```

Creating randomForest Model to know important features

```
house.rf <- randomForest(price ~ ., data = house,
                          importance = TRUE)
print(house.rf)
```

```
##
## Call:
## randomForest(formula = price ~ ., data = house, importance = TRUE)
##                Type of random forest: regression
##                Number of trees: 500
## No. of variables tried at each split: 5
##
##                Mean of squared residuals: 6091072661
##                % Var explained: 85.97
```

```
import <- house.rf$importance
import
```

```
##                %IncMSE IncNodePurity
## bedrooms      444355257 9.277752e+12
## bathrooms     1250016759 2.381699e+13
## sqft_living    10099930426 1.538718e+14
## sqft_lot       2220350443 2.483548e+13
## floors        660008739 6.595819e+12
## waterfront     62262808 1.651742e+12
## view          633386988 1.207915e+13
## condition     568863470 7.562389e+12
```

```
## grade                8855561477 1.276044e+14
## sqft_above           4119744234 5.503862e+13
## yr_built             4004233976 3.570054e+13
## yr_renovated         71132347 2.982924e+12
## lat                  28848985617 2.686521e+14
## long                 4840933511 3.976165e+13
## sqft_living15        4640192179 7.223977e+13
## sqft_lot15           2143322463 2.668528e+13
## sqft_basement_yesno 530044469 5.866989e+12
```

Save only important features

```
keep <- c('price', 'lat', 'sqft_living', 'grade', 'sqft_living15', 'sqft_above', 'long', 'yr_built', 'sqft_lot15')
house <- house[names(house) %in% keep]
summary(house)
```

```
##      price      bathrooms      sqft_living      sqft_lot
## Min.   : 75000   Min.   :0.000   Min.   : 290   Min.   :  520
## 1st Qu.: 315000 1st Qu.:1.500   1st Qu.:1400 1st Qu.:  5000
## Median : 437500 Median :2.000   Median :1860 Median :  7500
## Mean   : 476985 Mean   :2.052   Mean   :1976 Mean   : 14610
## 3rd Qu.: 600000 3rd Qu.:2.500   3rd Qu.:2431 3rd Qu.: 10319
## Max.   :1127500 Max.   :7.500   Max.   :7480 Max.   :1651359
##      grade      sqft_above      yr_built      lat
## Min.   : 1.000   Min.   : 290   Min.   :1900   Min.   :47.16
## 1st Qu.: 7.000   1st Qu.:1170   1st Qu.:1951   1st Qu.:47.46
## Median : 7.000   Median :1520   Median :1974   Median :47.57
## Mean   : 7.531   Mean   :1708   Mean   :1971   Mean   :47.56
## 3rd Qu.: 8.000   3rd Qu.:2100   3rd Qu.:1996   3rd Qu.:47.68
## Max.   :12.000   Max.   :5710   Max.   :2015   Max.   :47.78
##      long      sqft_living15      sqft_lot15
## Min.   :-122.5   Min.   : 399   Min.   :  651
## 1st Qu.: -122.3 1st Qu.:1470   1st Qu.:  5046
## Median : -122.2 Median :1800   Median :  7542
## Mean   : -122.2 Mean   :1922   Mean   : 12447
## 3rd Qu.: -122.1 3rd Qu.:2280   3rd Qu.:  9884
## Max.   : -121.3 Max.   :5380   Max.   :871200
```

We decided not to convert the numerical variables to a factor

```
# house$bathrooms = as.factor(house$bathrooms)
# house$grade = as.factor(house$grade)
```

Split dataset to a train set and a test set

```
s = sort(sample(nrow(house), nrow(house)*.7))
train <- house[s,]
test <- house[-s,]

# Create a rmse function to test results
rmse <- function(y_hat, y) sqrt(mean((y_hat - y)^2))
```

Create linear Models

```
lMod <- lm(price~., data=train)
summary(lMod)
```

```
##
## Call:
## lm(formula = price ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -545533  -77685   -8947   64083  645268
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.548e+07  1.101e+06 -23.150 < 2e-16 ***
## bathrooms    3.302e+04  2.264e+03  14.587 < 2e-16 ***
## sqft_living   6.087e+01  3.055e+00  19.922 < 2e-16 ***
## sqft_lot      1.946e-01  3.747e-02   5.194 2.09e-07 ***
## grade        7.766e+04  1.584e+03  49.025 < 2e-16 ***
## sqft_above    8.392e+00  2.933e+00   2.862  0.00422 **
## yr_built     -2.002e+03  4.729e+01 -42.339 < 2e-16 ***
## lat          5.132e+05  7.343e+03  69.889 < 2e-16 ***
## long         -3.771e+04  8.383e+03  -4.499 6.89e-06 ***
## sqft_living15  5.353e+01  2.645e+00  20.241 < 2e-16 ***
## sqft_lot15    -9.237e-02  5.567e-02  -1.659  0.09712 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 118400 on 14315 degrees of freedom
## Multiple R-squared:  0.674, Adjusted R-squared:  0.6738
## F-statistic: 2960 on 10 and 14315 DF, p-value: < 2.2e-16
rmse(test$price, predict(lMod,test[-1]))
```

```
## [1] 120550.7
```

```
# Use step function
```

```
lstepMod <- step(lMod)
```

```
## Start: AIC=334709
## price ~ bathrooms + sqft_living + sqft_lot + grade + sqft_above +
##      yr_built + lat + long + sqft_living15 + sqft_lot15
##
##              Df Sum of Sq      RSS      AIC
## <none>                2.0054e+14 334709
## - sqft_lot15          1 3.8561e+10 2.0058e+14 334710
## - sqft_above          1 1.1474e+11 2.0066e+14 334715
## - long                1 2.8351e+11 2.0082e+14 334727
## - sqft_lot            1 3.7790e+11 2.0092e+14 334734
## - bathrooms           1 2.9810e+12 2.0352e+14 334918
## - sqft_living          1 5.5601e+12 2.0610e+14 335099
## - sqft_living15        1 5.7395e+12 2.0628e+14 335111
## - yr_built            1 2.5112e+13 2.2565e+14 336397
## - grade               1 3.3670e+13 2.3421e+14 336930
## - lat                 1 6.8427e+13 2.6897e+14 338913
```

```
summary(lstepMod)
```

```
##
## Call:
## lm(formula = price ~ bathrooms + sqft_living + sqft_lot + grade +
##      sqft_above + yr_built + lat + long + sqft_living15 + sqft_lot15,
##      data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -545533  -77685   -8947   64083  645268
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.548e+07  1.101e+06 -23.150 < 2e-16 ***
## bathrooms    3.302e+04  2.264e+03  14.587 < 2e-16 ***
## sqft_living   6.087e+01  3.055e+00  19.922 < 2e-16 ***
## sqft_lot      1.946e-01  3.747e-02   5.194 2.09e-07 ***
## grade         7.766e+04  1.584e+03  49.025 < 2e-16 ***
## sqft_above    8.392e+00  2.933e+00   2.862 0.00422 **
## yr_built     -2.002e+03  4.729e+01 -42.339 < 2e-16 ***
## lat           5.132e+05  7.343e+03  69.889 < 2e-16 ***
## long          -3.771e+04  8.383e+03  -4.499 6.89e-06 ***
## sqft_living15  5.353e+01  2.645e+00  20.241 < 2e-16 ***
## sqft_lot15    -9.237e-02  5.567e-02  -1.659 0.09712 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 118400 on 14315 degrees of freedom
## Multiple R-squared:  0.674, Adjusted R-squared:  0.6738
## F-statistic: 2960 on 10 and 14315 DF, p-value: < 2.2e-16
```

```
rmse(test$price, predict(lstepMod, test[-1]))
```

```
## [1] 120550.7
```

Create a randomforest model

```
rfMod <- randomForest(price ~ ., data = train,
                      importance = TRUE)
print(rfMod)
```

```
##
## Call:
## randomForest(formula = price ~ ., data = train, importance = TRUE)
##              Type of random forest: regression
##              Number of trees: 500
## No. of variables tried at each split: 3
##
##              Mean of squared residuals: 6398278541
##              % Var explained: 85.1
```

```
rmse(test$price, predict(rfMod, test[-1]))
```

```
## [1] 84023.75
```

```
rfMod$importance
```

```
##              %IncMSE IncNodePurity
## bathrooms      1431047068 1.707596e+13
## sqft_living  10456928305 1.074197e+14
## sqft_lot    2608869331 1.938115e+13
## grade        9208002588 9.213277e+13
## sqft_above   4161959456 4.433853e+13
## yr_built     4593475321 2.953932e+13
## lat          28122805683 1.882928e+14
## long         5113331231 3.065476e+13
## sqft_living15 5504222443 5.489101e+13
## sqft_lot15    2485211484 2.189853e+13
```

Create PCR models

```
set.seed(27)
pc <- prcomp(house, scale = T)
summary(pc)
```

```
## Importance of components:
##              PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation  2.1783 1.3425 1.1952 0.88863 0.81081 0.64409 0.55685
## Proportion of Variance 0.4314 0.1638 0.1299 0.07179 0.05977 0.03771 0.02819
## Cumulative Proportion 0.4314 0.5952 0.7251 0.79686 0.85662 0.89434 0.92253
##              PC8      PC9      PC10     PC11
## Standard deviation  0.53154 0.51950 0.43483 0.33276
## Proportion of Variance 0.02568 0.02453 0.01719 0.01007
## Cumulative Proportion 0.94821 0.97275 0.98993 1.00000
```

```
sort(round(pc$rotation[,1], 2))
```

```
##      lat      sqft_lot      sqft_lot15      long      yr_built
##      0.02      0.11      0.12      0.21      0.26
##      price      bathrooms sqft_living15      grade      sqft_living
##      0.31      0.36      0.38      0.39      0.41
##      sqft_above
##      0.41
```

```
# PCR
```

```
pcrMod <- pcr(price ~ ., data = train, ncomp = 5)
```

```
rmse(predict(pcrMod, nncomp = 5), train$price) # RMSE = 173611.4
```

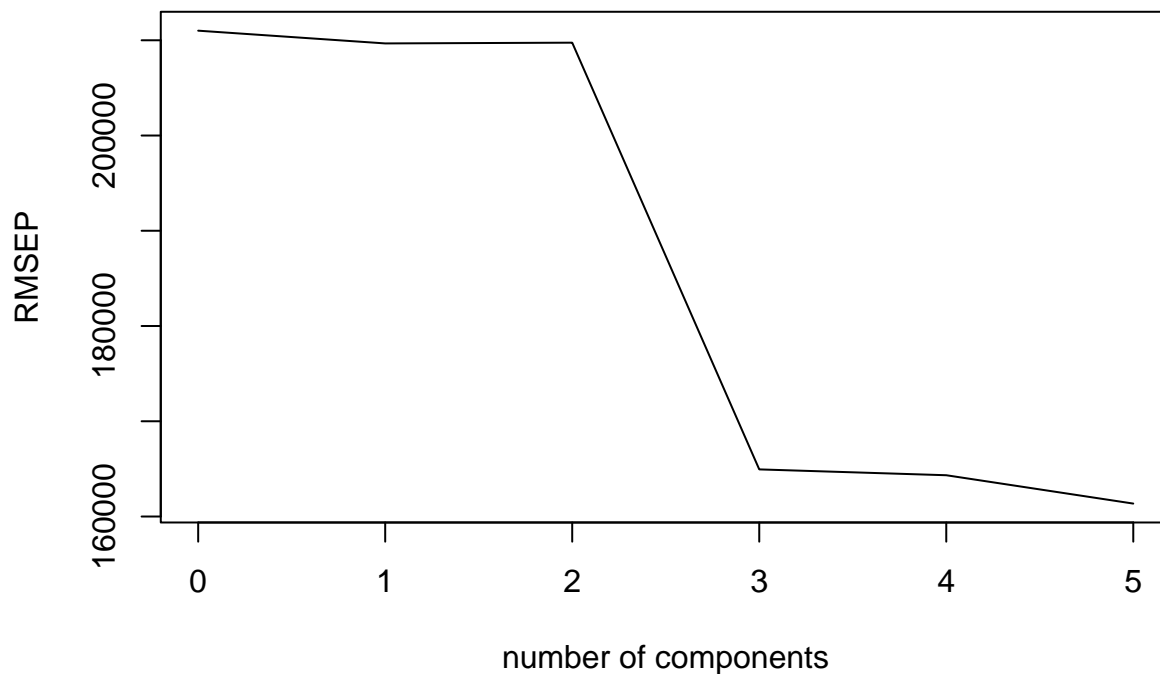
```
## [1] 180528.5
```

```
rmse(predict(pcrMod, nncomp = 5), test$price) # RMSE = 219128.2
```

```
## Warning in y_hat - y: longer object length is not a multiple of shorter object
## length
```

```
## [1] 234244.4
```

```
pcrmse <- RMSEP(pcrMod, newdata = test)
plot(pcrmse, main = "")
```



```
which.min(pcrmse$val) # 6 pc
```

```
## [1] 6
```

```
pcrmse$val[6] # 153961.9
```

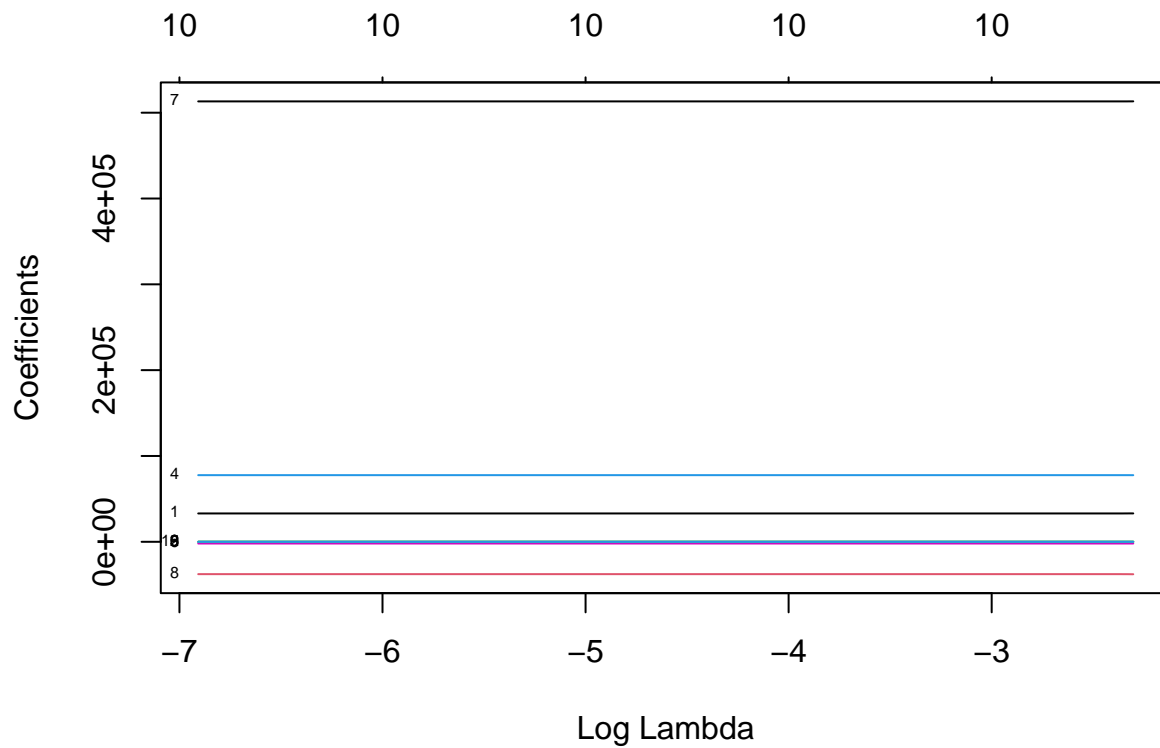
```
## [1] 161358.9
```

```
# I couldn't find pcrMod_2. Did I delete something?
#pcrCV <- RMSEP(pcrMod_2, estimate = "CV")
#plot(pcrmse, main = "PCR vs RMSE")
```

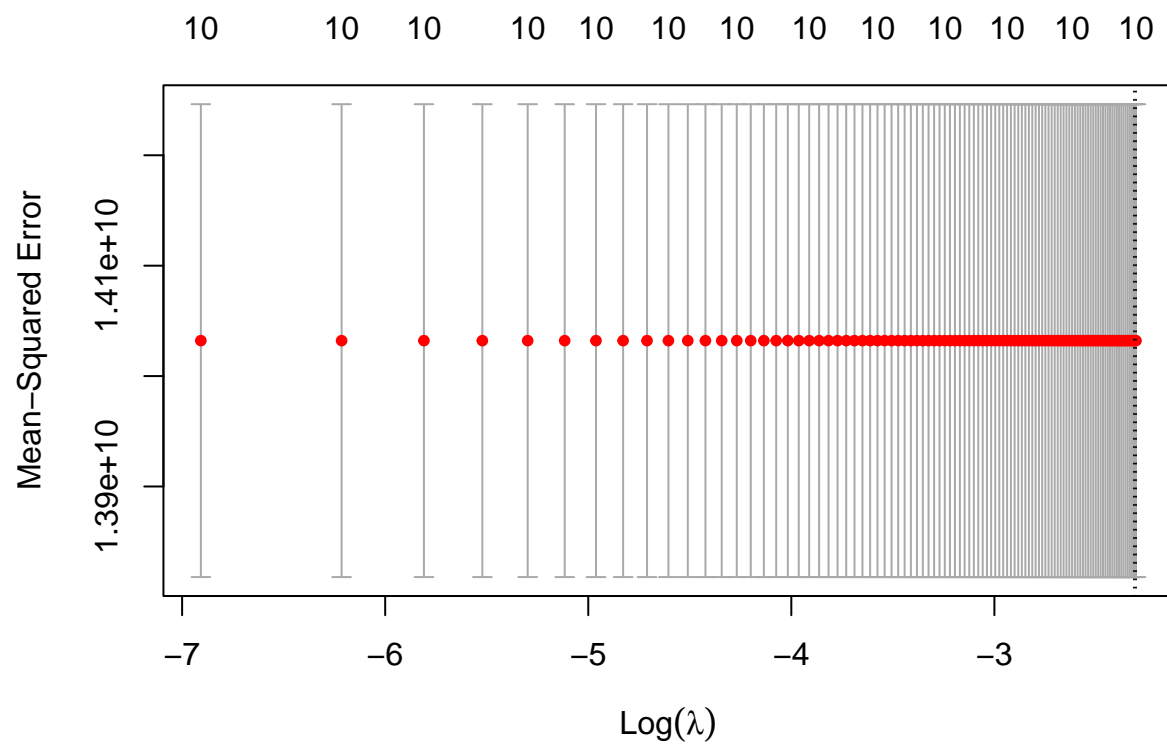
Create Ridge/LASSO models

```
set.seed(101)
# Create a ridge model
lambdas_to_try = lambda=seq(0.001,0.1, by=0.001)

ridgeMod <- cv.glmnet(as.matrix(train[,-1]), train$price, alpha = 0
                      , lambda = lambdas_to_try
                      ,standardize = TRUE, nfolds =10)
plot(ridgeMod$glmnet.fit,xvar = "lambda", label = T)
```

```
plot(ridgeMod)
```



```
ridgeMod$lambda.min
```

```
## [1] 0.099
```

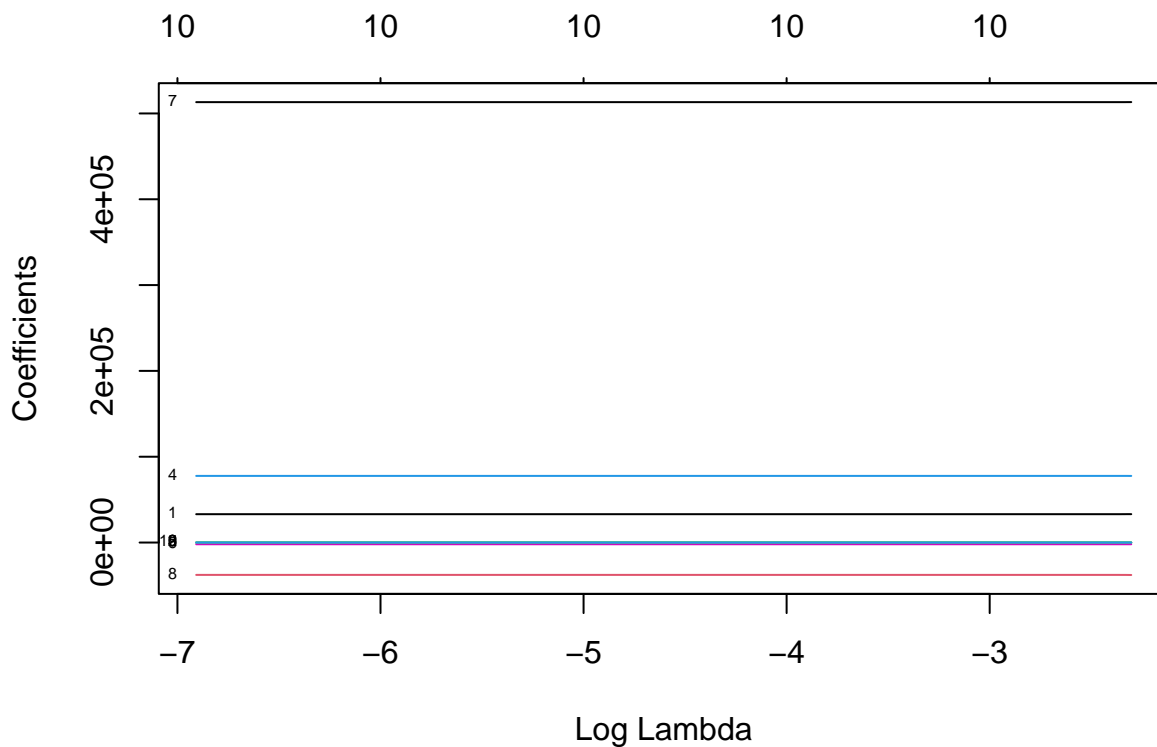
```
# Create a LASSO model
```

```
lassoMod <- cv.glmnet(as.matrix(train[,-1]), train$price, alpha = 1,
```

```

lambda = lambdas_to_try,
standardize = TRUE, nfolds = 10)
plot(lassoMod$glmnet.fit,xvar = "lambda", label = T)

```



```
lassoMod$lambda.min
```

```
## [1] 0.098
```

Create a df showing all the rmse values

```

rmse_colnames<-c("Model1-lMod","Model2-lstepMod","Model3-rfMod", "Model4-pcrMod"
, "Model5-Ridge", "Model6-Lasso")
rmse_result <-c( rmse(predict(lMod, test), test$price)
,rmse(predict(lstepMod, test), test$price)
,rmse(predict(rfMod, test), test$price)
,rmse(predict(pcrMod, nncomp = 5), test$price)
,rmse(predict(ridgeMod, newx = as.matrix(test[,-1])
, s=ridgeMod$lambda.min)
,test$price)
,rmse(predict(lassoMod, newx = as.matrix(test[,-1])
, s=lassoMod$lambda.min)
,test$price)
)

```

```

## Warning in y_hat - y: longer object length is not a multiple of shorter object
## length

```

```

result_df <- data.frame(rmse_colnames,rmse_result)
result_df

```

```
##      rmse_colnames rmse_result
```

## 1	Model1-lMod	120550.69
## 2	Model2-lstepMod	120550.69
## 3	Model3-rfMod	84023.75
## 4	Model4-pcrMod	234244.44
## 5	Model5-Ridge	120550.89
## 6	Model6-Lasso	120550.78