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

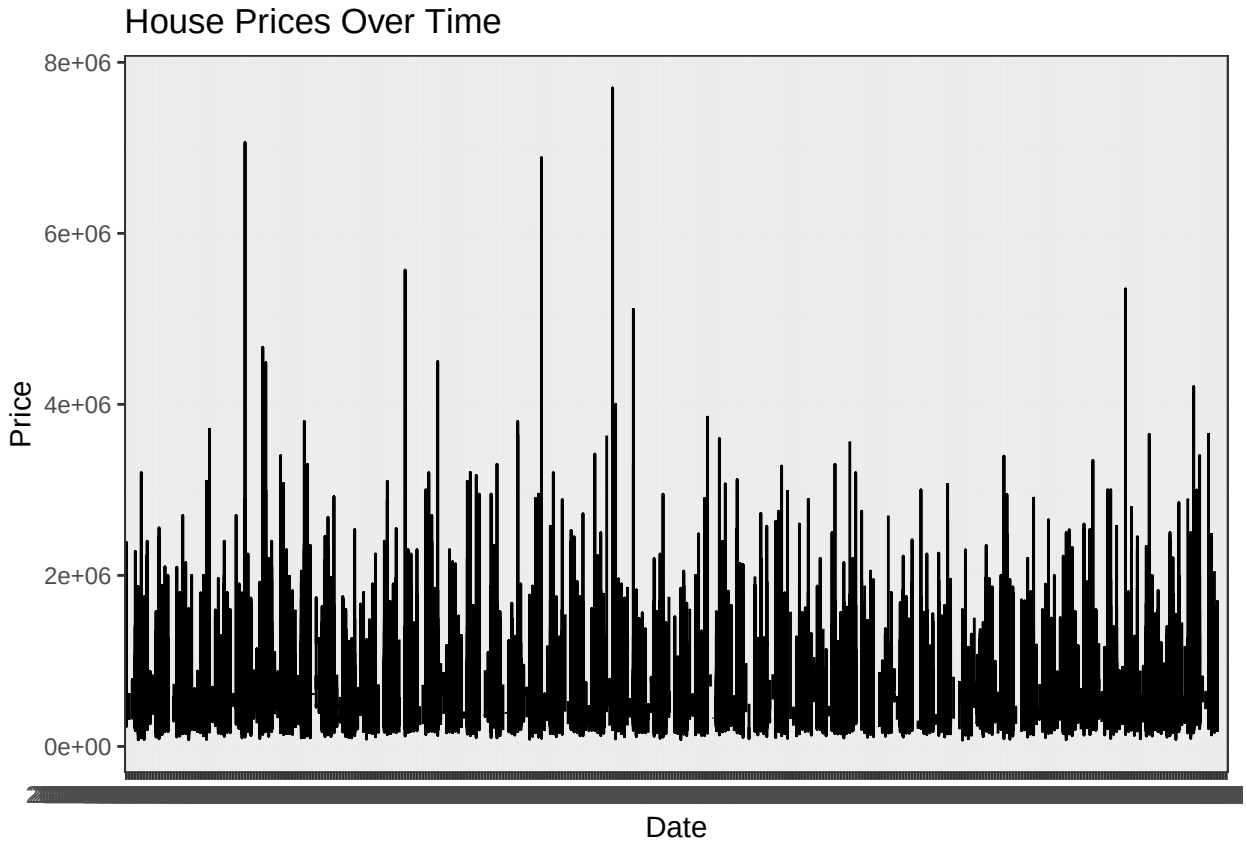
```
set.seed(1)

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
```

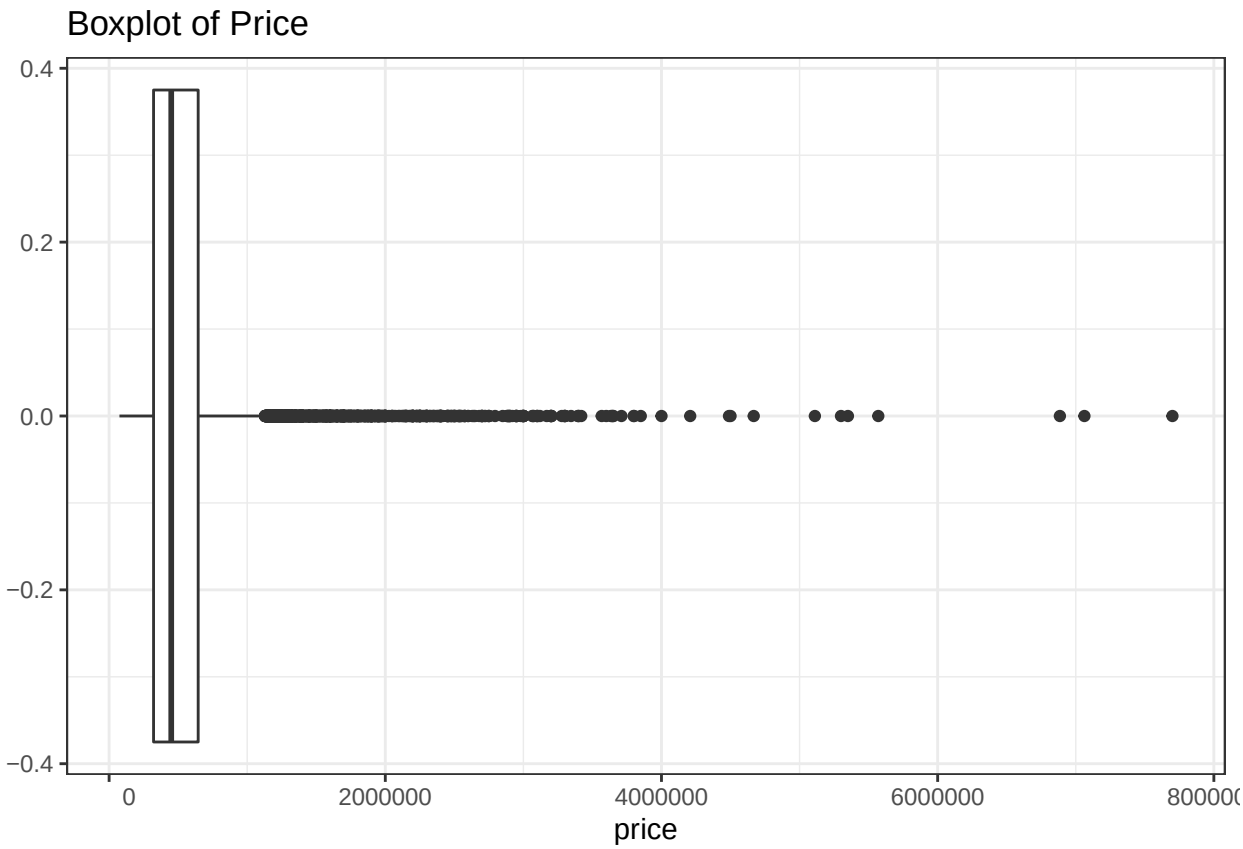
```
dim(house)
```

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

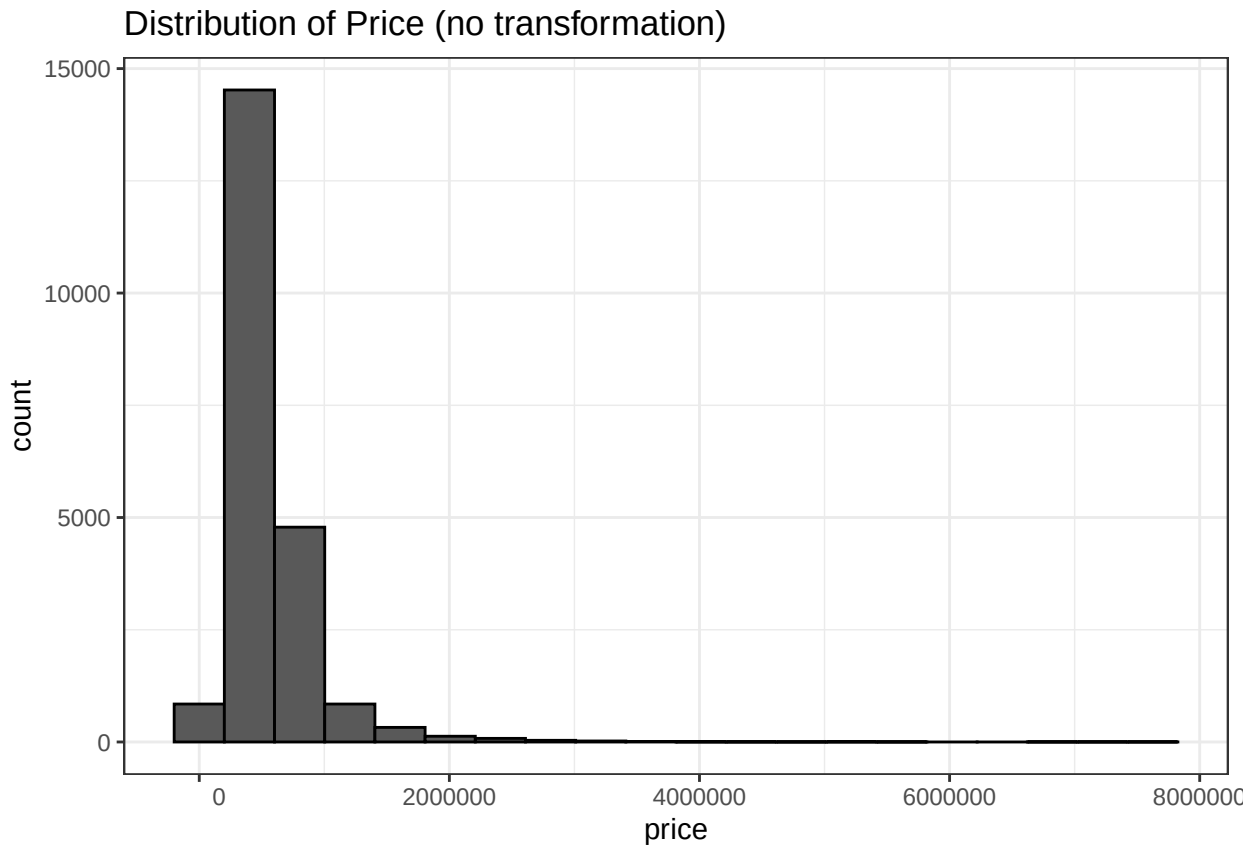
```
num_ob_bf_drop <- dim(house)[1]

# 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
  }
}

#DataExplorer::create_report(df)

# 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))
```



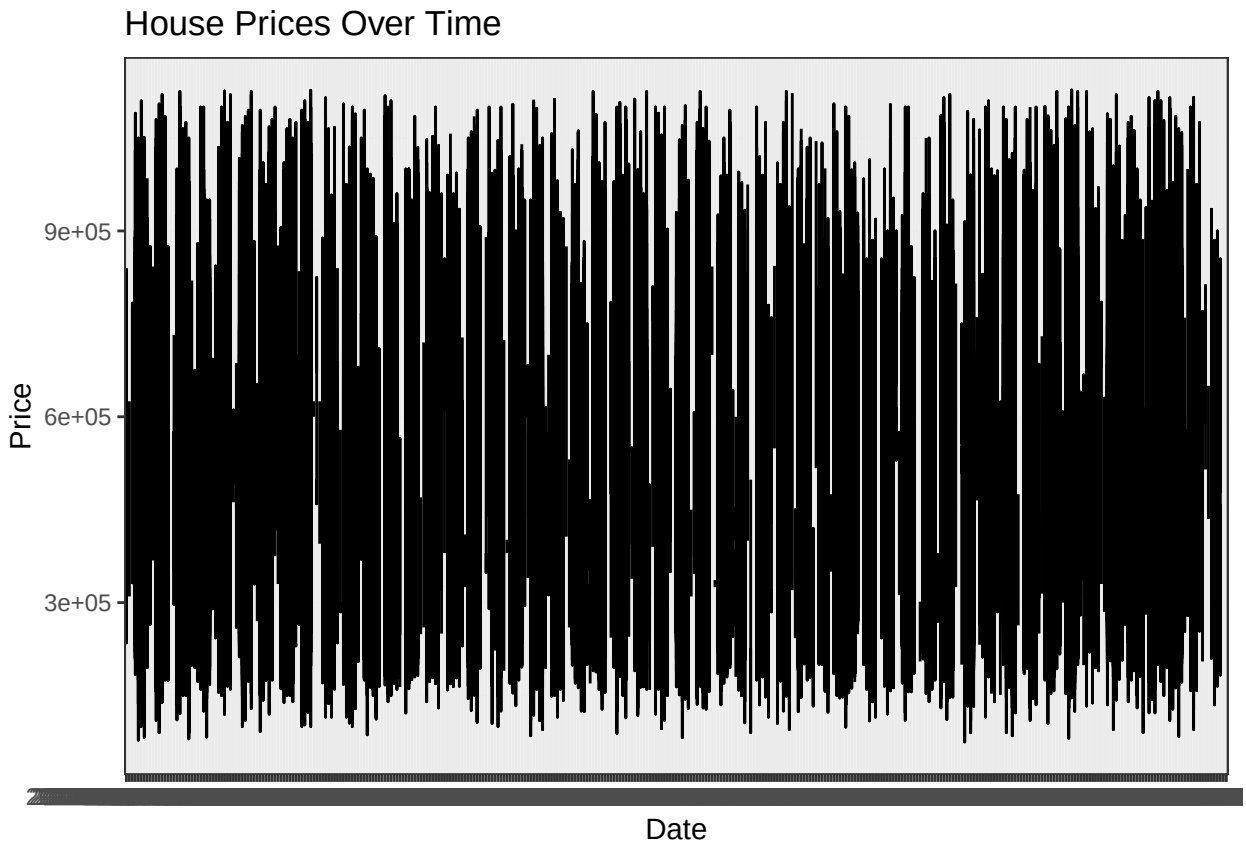

```
# Distribution of price
ggplot(house)+
  aes(x=price)+
  geom_histogram(col = 'black', bins = 20) +
  ggtitle("Distribution of Price (no transformation)") +
  theme_bw() +
  scale_x_continuous(labels = function(x) format(x, scientific = FALSE))
```



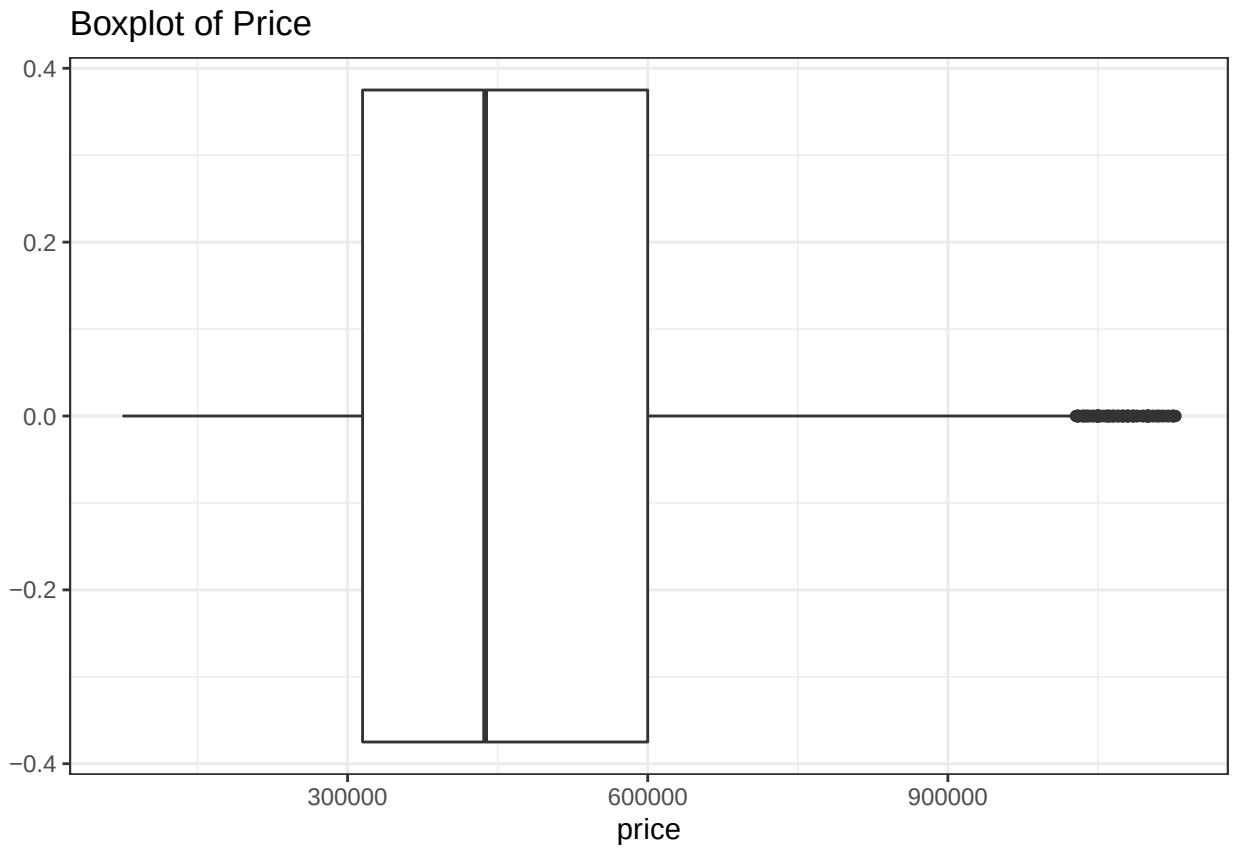

```
# Get rid of outliers (price-wise)
summary(house$price)
```

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

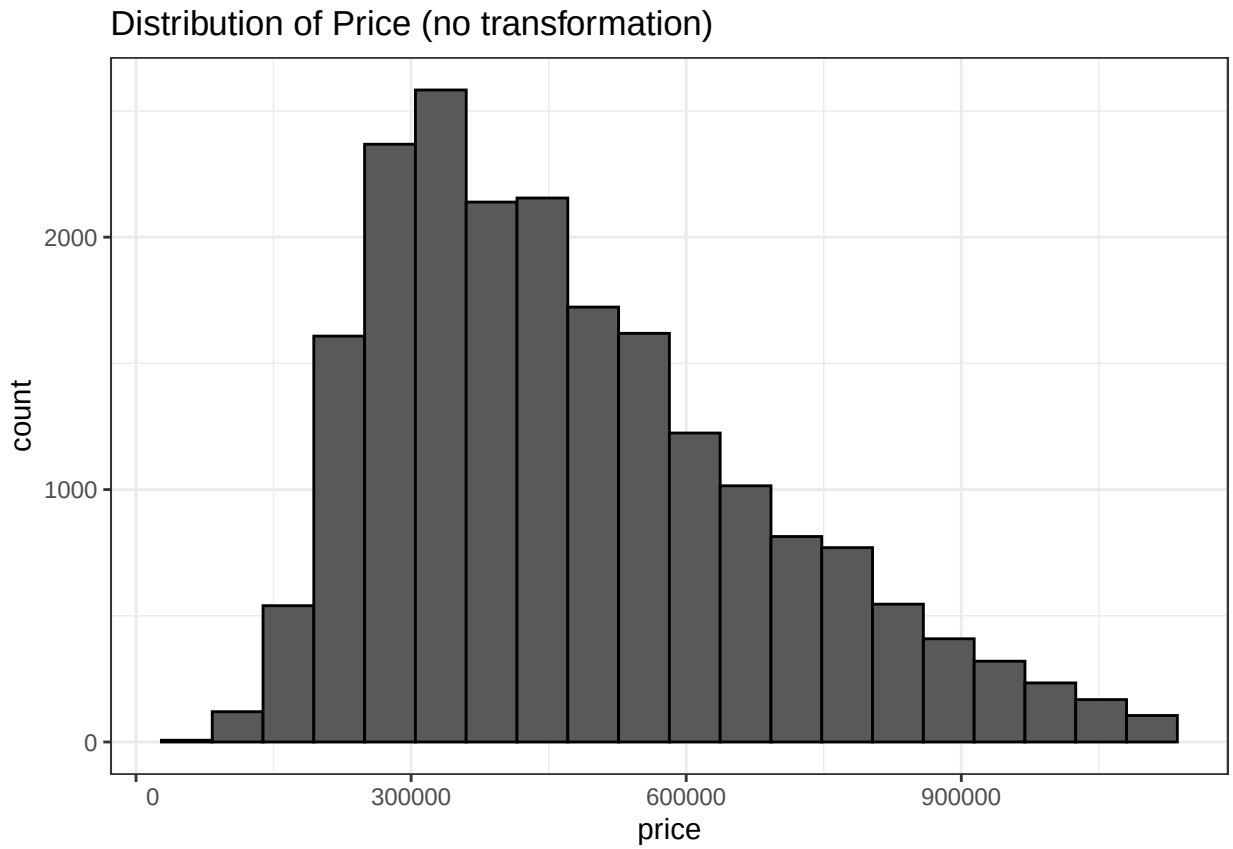
```
first_quartile <- summary(house$price)[[2]]
third_quartile <- summary(house$price)[[5]]
IQR <- third_quartile-first_quartile
Upper <- 1.5*IQR + third_quartile
Lower <- first_quartile - 1.5*IQR
house <- subset(house, price >= Lower & price <= Upper)

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



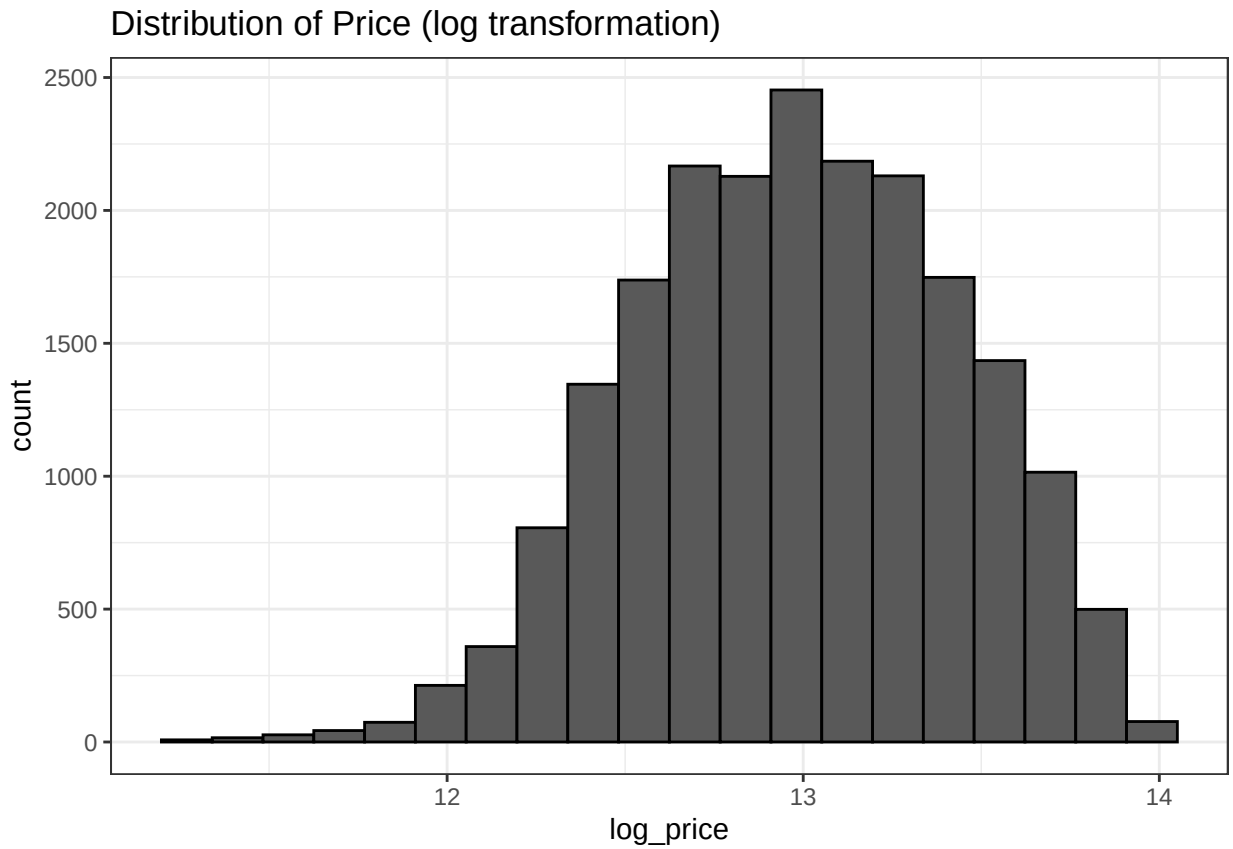

```
# Boxplot of prices
ggplot(house)+
  aes(x=price)+
  geom_boxplot() +
  ggtitle("Boxplot of Price") +
  theme_bw() +
  scale_x_continuous(labels = function(x) format(x, scientific = FALSE))
```




```
# 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 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()
```




```
# Drop date: No relationshop 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')
house <- house[!names(house) %in% drop]
summary(house)
```

```
##      price            bedrooms       bathrooms      sqft_living
##  Min.   :  75000   Min.   : 0.00   Min.   :0.000   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   log_price
##  Min.   :   651   Min.   :0.0000      Min.   :11.23
##  1st Qu.:  5046   1st Qu.:0.0000      1st Qu.:12.66
##  Median :  7542   Median :0.0000      Median :12.99
##  Mean   : 12447   Mean   :0.3794      Mean   :12.98
##  3rd Qu.:  9884   3rd Qu.:1.0000      3rd Qu.:13.30
##  Max.   :871200   Max.   :1.0000      Max.   :13.94
```

```
dim(house)
```

```
## [1] 20467    19
```

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

num_ob_af_drop/num_ob_bf_drop*100
```

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

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: 6
##
##           Mean of squared residuals: 79114232
##                     % Var explained: 99.82
```

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

```
##                          %IncMSE IncNodePurity
## bedrooms            3.696193e+07  2.077459e+12
## bathrooms           1.422591e+08  8.146033e+12
## sqft_living         1.166858e+09  8.291708e+13
## sqft_lot            1.637426e+08  3.088760e+12
## floors              4.614741e+07  1.168794e+12
## waterfront          4.031845e+06  2.828339e+11
## view                2.439634e+07  1.816970e+12
## condition           2.221408e+07  7.470888e+11
## grade               8.972737e+08  6.197275e+13
```

```
## sqft_above          3.271938e+08  1.567614e+13
## yr_built            3.071369e+08  5.539002e+12
## yr_renovated        1.175449e+05  3.536349e+11
## lat                 1.922974e+09  1.053769e+14
## long                3.369882e+08  5.580133e+12
## sqft_living15       3.471413e+08  2.418153e+13
## sqft_lot15          1.596229e+08  3.923662e+12
## sqft_basement_yesno 3.405284e+07  1.192210e+12
## log_price           6.808462e+10  5.630963e+14
```

Save only important feaatures

```
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 dateset 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
## -543457  -76600   -8994   64045  661546
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.657e+07  1.100e+06 -24.149  < 2e-16 ***
## bathrooms      3.311e+04  2.255e+03  14.683  < 2e-16 ***
## sqft_living    6.215e+01  3.052e+00  20.364  < 2e-16 ***
## sqft_lot       2.693e-01  3.392e-02   7.938 2.21e-15 ***
## grade          7.624e+04  1.578e+03  48.317  < 2e-16 ***
## sqft_above     6.610e+00  2.932e+00   2.255   0.0242 *
## yr_built      -1.900e+03  4.729e+01 -40.171  < 2e-16 ***
## lat            5.259e+05  7.333e+03  71.715  < 2e-16 ***
## long          -4.011e+04  8.340e+03  -4.810 1.53e-06 ***
## sqft_living15  5.033e+01  2.646e+00  19.017  < 2e-16 ***
## sqft_lot15    -1.067e-01  5.391e-02  -1.980   0.0478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 118500 on 14315 degrees of freedom
## Multiple R-squared:  0.6738, Adjusted R-squared:  0.6735
## F-statistic:  2956 on 10 and 14315 DF,  p-value: < 2.2e-16
```

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

```
## [1] 120323.7
```

```
# Use step function
lstepMod <- step(lMod)
```

```
## Start:  AIC=334734.6
## price ~ bathrooms + sqft_living + sqft_lot + grade + sqft_above +
##     yr_built + lat + long + sqft_living15 + sqft_lot15
##
##                 Df  Sum of Sq        RSS    AIC
## <none>                        2.0090e+14 334735
## - sqft_lot15     1 5.4997e+10 2.0095e+14 334736
## - sqft_above     1 7.1344e+10 2.0097e+14 334738
## - long           1 3.2463e+11 2.0122e+14 334756
## - sqft_lot       1 8.8422e+11 2.0178e+14 334795
## - bathrooms      1 3.0257e+12 2.0392e+14 334947
## - sqft_living15  1 5.0755e+12 2.0597e+14 335090
## - sqft_living    1 5.8199e+12 2.0672e+14 335142
## - yr_built       1 2.2647e+13 2.2355e+14 336263
## - grade          1 3.2763e+13 2.3366e+14 336897
## - lat            1 7.2179e+13 2.7308e+14 339130
```

```
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
## -543457  -76600   -8994   64045  661546
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.657e+07  1.100e+06 -24.149  < 2e-16 ***
## bathrooms      3.311e+04  2.255e+03  14.683  < 2e-16 ***
## sqft_living    6.215e+01  3.052e+00  20.364  < 2e-16 ***
## sqft_lot       2.693e-01  3.392e-02   7.938 2.21e-15 ***
## grade          7.624e+04  1.578e+03  48.317  < 2e-16 ***
## sqft_above     6.610e+00  2.932e+00   2.255   0.0242 *
## yr_built      -1.900e+03  4.729e+01 -40.171  < 2e-16 ***
## lat            5.259e+05  7.333e+03  71.715  < 2e-16 ***
## long          -4.011e+04  8.340e+03  -4.810 1.53e-06 ***
## sqft_living15  5.033e+01  2.646e+00  19.017  < 2e-16 ***
## sqft_lot15    -1.067e-01  5.391e-02  -1.980   0.0478 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 118500 on 14315 degrees of freedom
## Multiple R-squared:  0.6738, Adjusted R-squared:  0.6735
## F-statistic:  2956 on 10 and 14315 DF,  p-value: < 2.2e-16
```

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

```
## [1] 120323.7
```

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: 6579044230
##                     % Var explained: 84.69
```

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

```
## [1] 82149.41
```

```
rfMod$importance
```

```
##                   %IncMSE IncNodePurity
## bathrooms      1403976690  1.800940e+13
## sqft_living   10846782405  1.143602e+14
## sqft_lot       2652025409  1.962410e+13
## grade          9093530901  8.853824e+13
## sqft_above     3814270098  3.765388e+13
## yr_built       4284864009  2.751343e+13
## lat           28483172914  1.901026e+14
## long           5269346317  3.100389e+13
## sqft_living15  5634813237  5.576185e+13
## sqft_lot15     2590185203  2.264353e+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
```

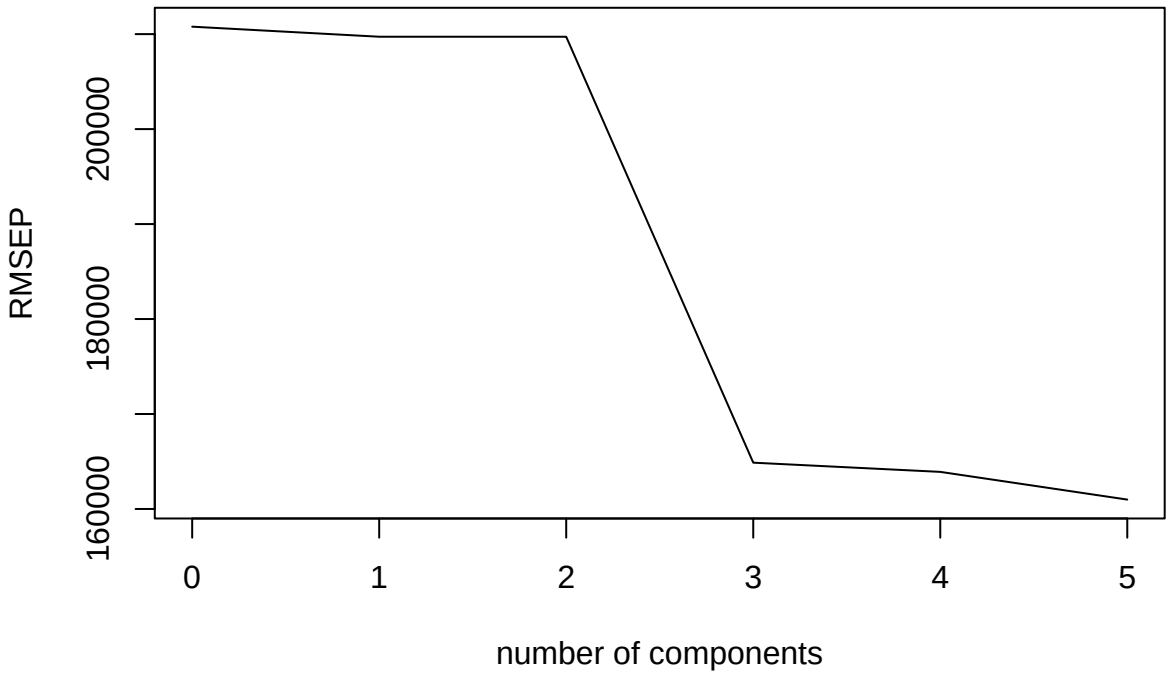
```
## [1] 180577.8
```

```
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] 234359.1
```

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




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

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

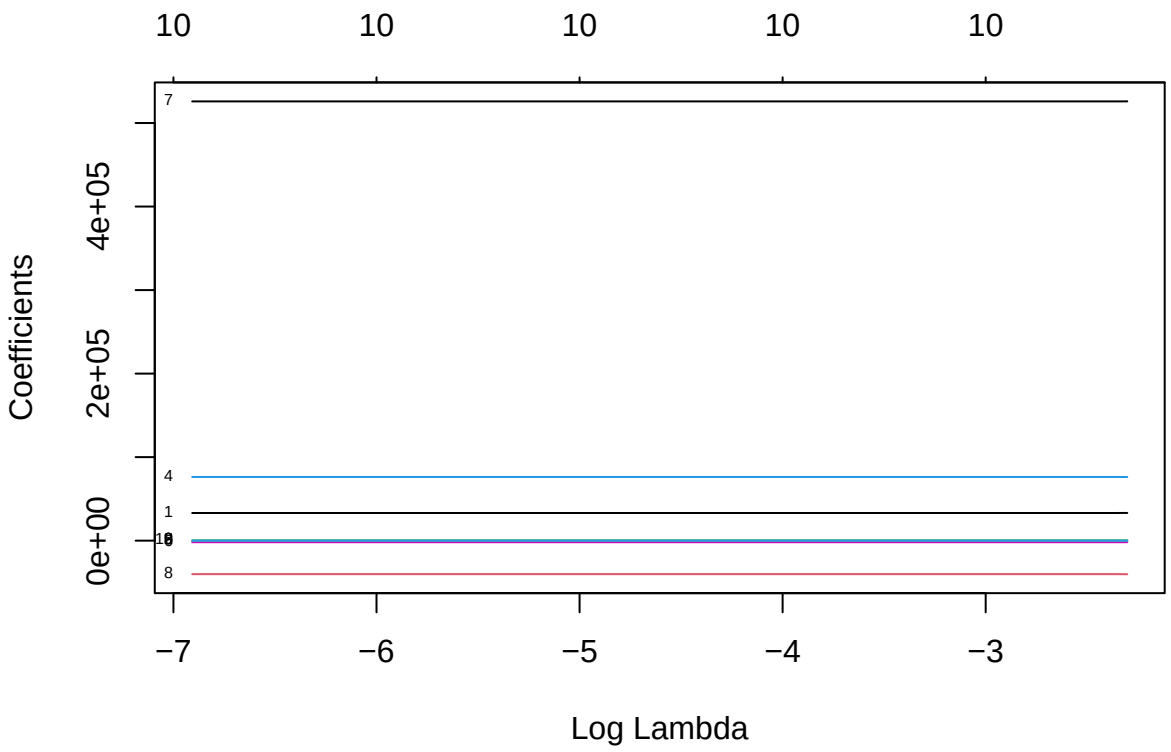
```
pcrmse$val[6] # 153961.9
```

```
## [1] 160990.4
```

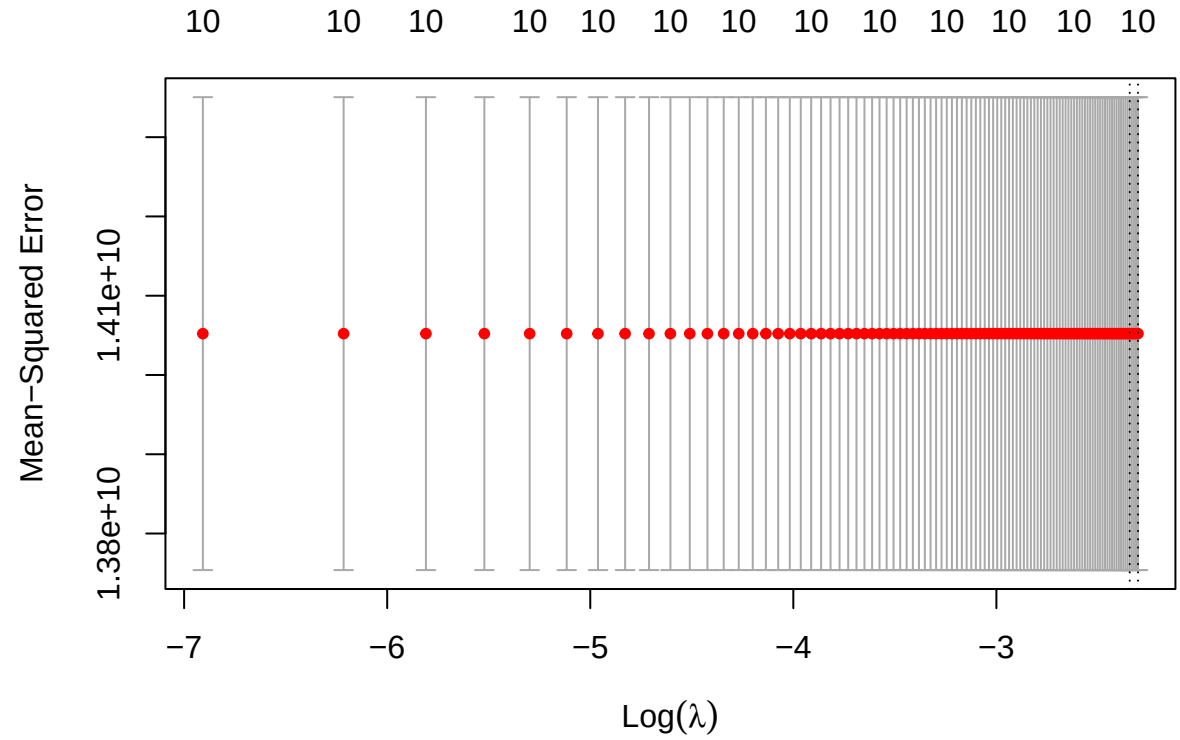
```
# I couldn't find pcrMod_2. Did I delete something?
#pcrCV <- RMSEP(pcrMod_2, estimate = "CV")
#plot(pcrmse, main = "PCR vs RMSE")
```

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)
```

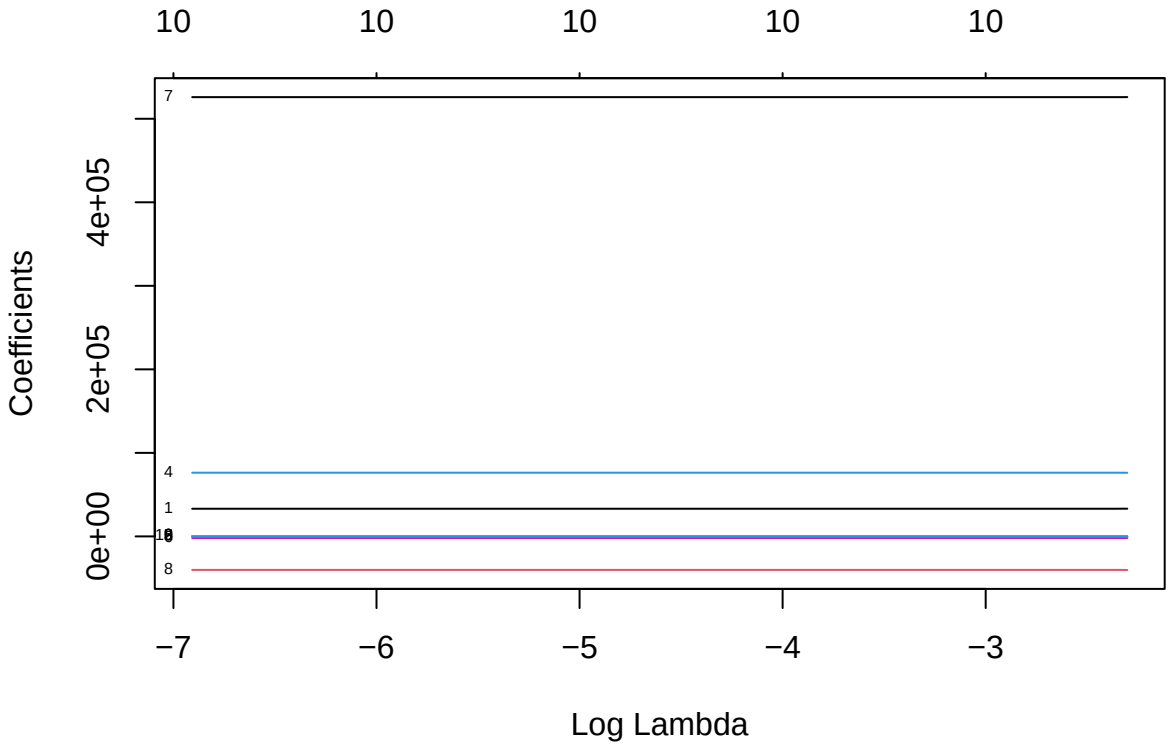



```
ridgeMod$lambda.min
```

```
## [1] 0.096
```

```
# 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] 0.1
```

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   120323.75
## 2 Model2-lstepMod   120323.75
## 3    Model3-rfMod    82149.41
## 4   Model4-pcrMod   234359.08
## 5    Model5-Ridge   120323.80
## 6    Model6-Lasso   120324.18
```