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

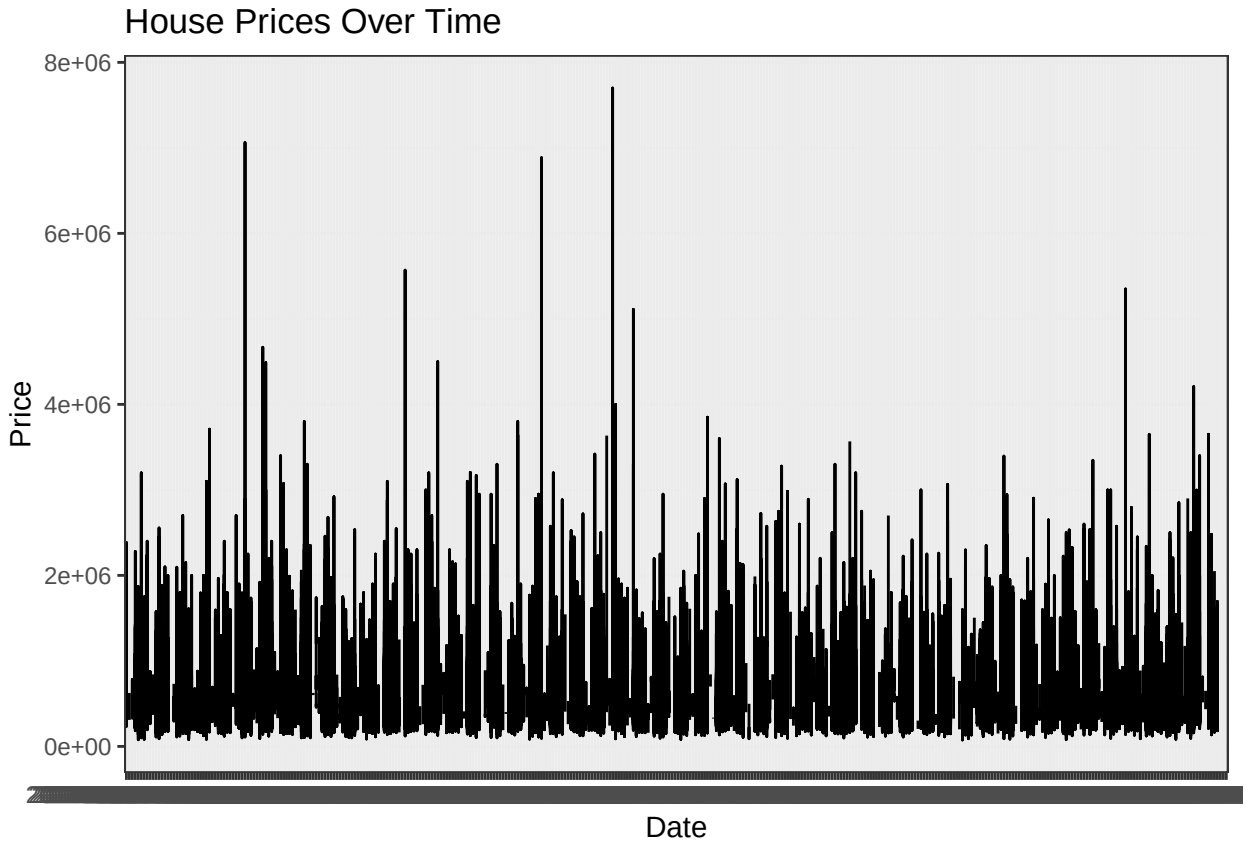
```
set.seed(1)

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
```

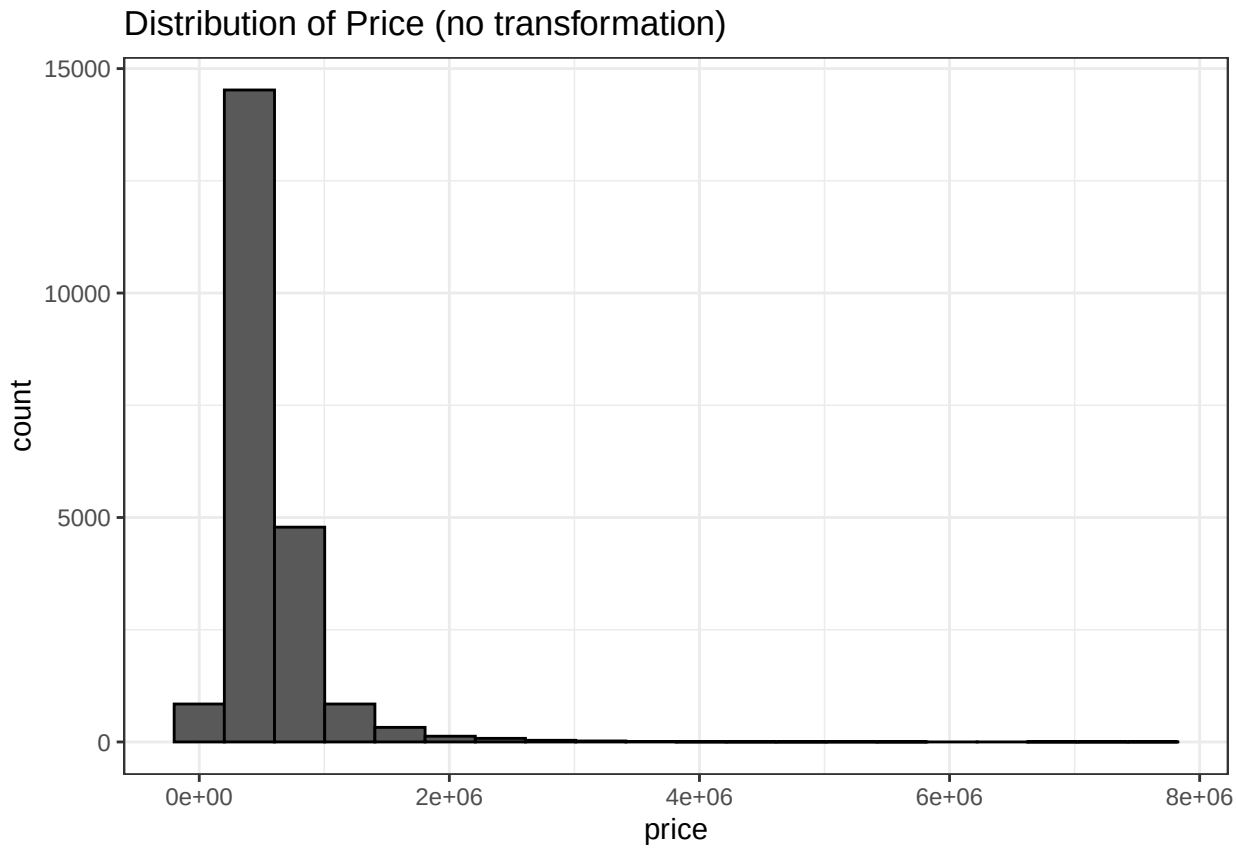
```
dim(house)
```

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

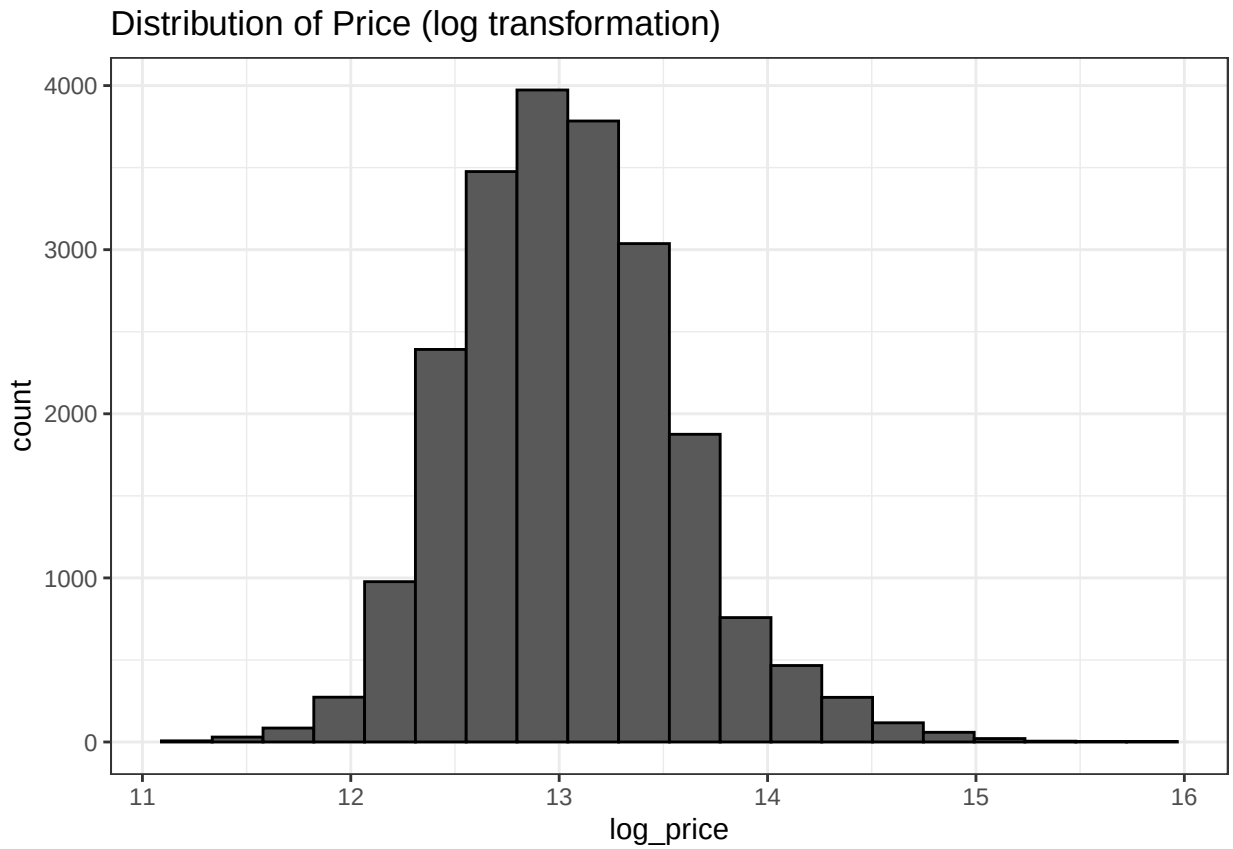
```
# Add a feature if there is a basement then 1 else 0
for(i in 1: nrow(house)){
    if (house$sqft_basement[i] >0) {
  house$sqft_basement_yesno[i] <- 1
  } else {
  house$sqft_basement_yesno[i] <- 0
  }
}

#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()
```




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




```
# 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()
```




```
# Get rid of outliers (price-wise)
IQR <- 645000-321950
Upper <- 1.5*IQR + 540088
Lower <- 540088 - 1.5*IQR
house <- subset(house, price >= Lower & price <= Upper)

# 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.   : 0.000   Min.   :0.00   Min.   : 290
##  1st Qu.: 314950   1st Qu.: 3.000   1st Qu.:1.50   1st Qu.:1390
##  Median : 435000   Median : 3.000   Median :2.00   Median :1840
##  Mean   : 468939   Mean   : 3.321   Mean   :2.04   Mean   :1957
##  3rd Qu.: 595000   3rd Qu.: 4.000   3rd Qu.:2.50   3rd Qu.:2410
##  Max.   :1020000   Max.   :33.000   Max.   :7.50   Max.   :7480
##     sqft_lot           floors        waterfront            view
##  Min.   :    520   Min.   :1.000   Min.   :0.000000   Min.   :0.0000
##  1st Qu.:   5000   1st Qu.:1.000   1st Qu.:0.000000   1st Qu.:0.0000
##  Median :   7500   Median :1.000   Median :0.000000   Median :0.0000
##  Mean   :  14526   Mean   :1.472   Mean   :0.002922   Mean   :0.1633
##  3rd Qu.:  10267   3rd Qu.:2.000   3rd Qu.:0.000000   3rd Qu.:0.0000
```

```
##  Max.   :1651359   Max.   :3.500   Max.   :1.000000   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 :1510   Median :1974
##  Mean   :3.405   Mean   : 7.507   Mean   :1694   Mean   :1971
##  3rd Qu.:4.000   3rd Qu.: 8.000   3rd Qu.:2080   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.:1461
##  Median :   0.00   Median :47.57   Median :-122.2   Median :1790
##  Mean   :  73.21   Mean   :47.56   Mean   :-122.2   Mean   :1909
##  3rd Qu.:   0.00   3rd Qu.:47.68   3rd Qu.:-122.1   3rd Qu.:2260
##  Max.   :2015.00   Max.   :47.78   Max.   :-121.3   Max.   :4950
##    sqft_lot15     sqft_basement_yesno   log_price
##  Min.   :   651   Min.   :0.0000      Min.   :11.23
##  1st Qu.:  5040   1st Qu.:0.0000      1st Qu.:12.66
##  Median :  7528   Median :0.0000      Median :12.98
##  Mean   : 12386   Mean   :0.3767      Mean   :12.97
##  3rd Qu.:  9840   3rd Qu.:1.0000      3rd Qu.:13.30
##  Max.   :871200   Max.   :1.0000      Max.   :13.84
```

```
dim(house)
```

```
## [1] 20194    19
```

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: 65067734
##                     % Var explained: 99.83
```

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

```
##                         %IncMSE IncNodePurity
## bedrooms               30758187  1.884660e+12
## bathrooms             103895168  6.368203e+12
## sqft_living           851653080  6.072275e+13
## sqft_lot              142147623  2.911063e+12
## floors                 44190477  1.285230e+12
## waterfront              2695912  3.276784e+11
## view                   22574262  1.505540e+12
## condition              17757073  6.171916e+11
## grade                 730798317  5.188177e+13
```

```
## sqft_above             326987710  1.538648e+13
## yr_built               240549054  4.983614e+12
## yr_renovated             1745448  2.681514e+11
## lat                   1879833213  1.138526e+14
## long                   292884364  4.852294e+12
## sqft_living15          299561013  2.472030e+13
## sqft_lot15             145860854  3.945836e+12
## sqft_basement_yesno     22010168  8.152095e+11
## log_price           61595994860  4.934767e+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.   :0.00   Min.   : 290   Min.   :    520
##  1st Qu.: 314950   1st Qu.:1.50   1st Qu.:1390   1st Qu.:   5000
##  Median : 435000   Median :2.00   Median :1840   Median :   7500
##  Mean   : 468939   Mean   :2.04   Mean   :1957   Mean   :  14526
##  3rd Qu.: 595000   3rd Qu.:2.50   3rd Qu.:2410   3rd Qu.:  10267
##  Max.   :1020000   Max.   :7.50   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 :1510   Median :1974   Median :47.57
##  Mean   : 7.507   Mean   :1694   Mean   :1971   Mean   :47.56
##  3rd Qu.: 8.000   3rd Qu.:2080   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.:1461   1st Qu.:  5040
##  Median :-122.2   Median :1790   Median :  7528
##  Mean   :-122.2   Mean   :1909   Mean   : 12386
##  3rd Qu.:-122.1   3rd Qu.:2260   3rd Qu.:  9840
##  Max.   :-121.3   Max.   :4950   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
## -534457  -75071   -8059   63676  605847
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.415e+07  1.059e+06 -22.796  < 2e-16 ***
## bathrooms      3.435e+04  2.192e+03  15.669  < 2e-16 ***
## sqft_living    5.208e+01  2.977e+00  17.493  < 2e-16 ***
## sqft_lot       2.250e-01  3.512e-02   6.406 1.54e-10 ***
## grade          7.677e+04  1.537e+03  49.950  < 2e-16 ***
## sqft_above     9.732e+00  2.857e+00   3.407 0.000659 ***
## yr_built      -1.961e+03  4.565e+01 -42.943  < 2e-16 ***
## lat            5.119e+05  7.027e+03  72.839  < 2e-16 ***
## long          -2.676e+04  8.032e+03  -3.332 0.000864 ***
## sqft_living15  5.154e+01  2.585e+00  19.940  < 2e-16 ***
## sqft_lot15    -4.237e-02  5.103e-02  -0.830 0.406313
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 113400 on 14124 degrees of freedom
## Multiple R-squared:  0.672,  Adjusted R-squared:  0.6717
## F-statistic:  2893 on 10 and 14124 DF,  p-value: < 2.2e-16
```

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

```
## [1] 115315.2
```

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

```
## Start:  AIC=329031.4
## price ~ bathrooms + sqft_living + sqft_lot + grade + sqft_above +
##     yr_built + lat + long + sqft_living15 + sqft_lot15
##
##                 Df  Sum of Sq        RSS    AIC
## - sqft_lot15     1 8.8649e+09 1.8157e+14 329030
## <none>                        1.8156e+14 329031
## - long           1 1.4273e+11 1.8171e+14 329041
## - sqft_above     1 1.4922e+11 1.8171e+14 329041
## - sqft_lot       1 5.2756e+11 1.8209e+14 329070
## - bathrooms      1 3.1563e+12 1.8472e+14 329273
## - sqft_living    1 3.9338e+12 1.8550e+14 329332
## - sqft_living15  1 5.1110e+12 1.8668e+14 329422
## - yr_built       1 2.3707e+13 2.0527e+14 330764
## - grade          1 3.2073e+13 2.1364e+14 331329
## - lat            1 6.8203e+13 2.4977e+14 333537
```

```
##
## Step:  AIC=329030.1
## price ~ bathrooms + sqft_living + sqft_lot + grade + sqft_above +
##     yr_built + lat + long + sqft_living15
##
##                 Df  Sum of Sq        RSS    AIC
## <none>                        1.8157e+14 329030
## - sqft_above     1 1.4995e+11 1.8172e+14 329040
## - long           1 1.5095e+11 1.8172e+14 329040
## - sqft_lot       1 7.9458e+11 1.8237e+14 329090
## - bathrooms      1 3.1819e+12 1.8476e+14 329274
## - sqft_living    1 3.9253e+12 1.8550e+14 329330
## - sqft_living15  1 5.1021e+12 1.8668e+14 329420
## - yr_built       1 2.3705e+13 2.0528e+14 330763
## - grade          1 3.2085e+13 2.1366e+14 331328
## - lat            1 6.8378e+13 2.4995e+14 333546
```

```
summary(lstepMod)
```

```
##
## Call:
## lm(formula = price ~ bathrooms + sqft_living + sqft_lot + grade +
##     sqft_above + yr_built + lat + long + sqft_living15, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -534710  -75036   -8113   63627  605871
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.424e+07  1.054e+06 -22.998  < 2e-16 ***
## bathrooms      3.444e+04  2.189e+03  15.733  < 2e-16 ***
## sqft_living    5.198e+01  2.975e+00  17.475  < 2e-16 ***
## sqft_lot       2.055e-01  2.614e-02   7.862 4.05e-15 ***
## grade          7.678e+04  1.537e+03  49.959  < 2e-16 ***
## sqft_above     9.756e+00  2.856e+00   3.415 0.000639 ***
## yr_built      -1.961e+03  4.565e+01 -42.943  < 2e-16 ***
## lat            5.121e+05  7.021e+03  72.933  < 2e-16 ***
## long          -2.740e+04  7.996e+03  -3.427 0.000613 ***
## sqft_living15  5.145e+01  2.582e+00  19.922  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 113400 on 14125 degrees of freedom
## Multiple R-squared:  0.672,  Adjusted R-squared:  0.6717
## F-statistic:  3215 on 9 and 14125 DF,  p-value: < 2.2e-16
```

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

```
## [1] 115336.4
```

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: 5913983919
##                     % Var explained: 84.9
```

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

```
## [1] 79267.93
```

```
rfMod$importance
```

```
##                   %IncMSE IncNodePurity
## bathrooms      1186616895  1.457086e+13
## sqft_living    9066440347  9.416361e+13
## sqft_lot       2025745094  1.787292e+13
## grade          7829623456  7.621103e+13
## sqft_above     3496104829  3.629273e+13
## yr_built       3979024243  2.605115e+13
## lat           27446903499  1.830042e+14
## long           4516429149  2.768532e+13
## sqft_living15  4678797326  4.917176e+13
## sqft_lot15     2223759364  1.980714e+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     2.170 1.3396 1.1997 0.8893 0.81396 0.65088 0.56212
## Proportion of Variance 0.428 0.1631 0.1308 0.0719 0.06023 0.03851 0.02873
## Cumulative Proportion  0.428 0.5911 0.7220 0.7939 0.85412 0.89263 0.92135
##                           PC8     PC9    PC10    PC11
## Standard deviation     0.53460 0.52462 0.43659 0.33685
## Proportion of Variance 0.02598 0.02502 0.01733 0.01032
## Cumulative Proportion  0.94734 0.97236 0.98968 1.00000
```

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

```
##           lat      sqft_lot    sqft_lot15          long      yr_built
##          0.01          0.11          0.12          0.21          0.26
##         price     bathrooms sqft_living15         grade   sqft_living
##          0.30          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
```

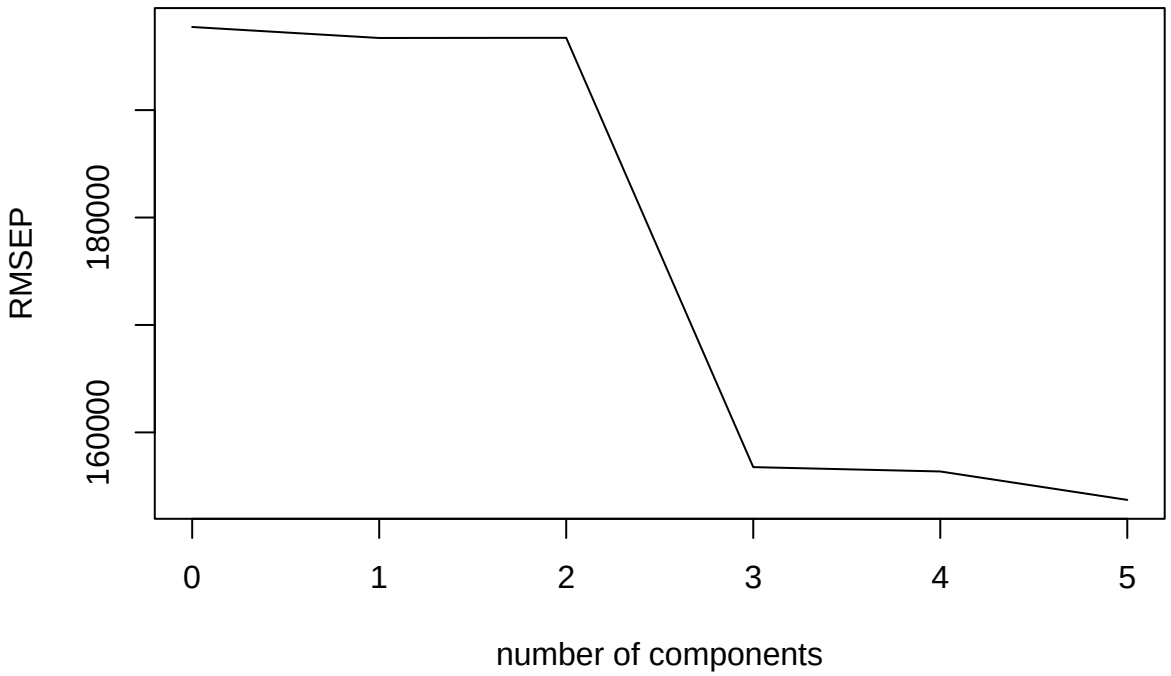
```
## [1] 173643.9
```

```
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] 219167.6
```

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




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

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

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

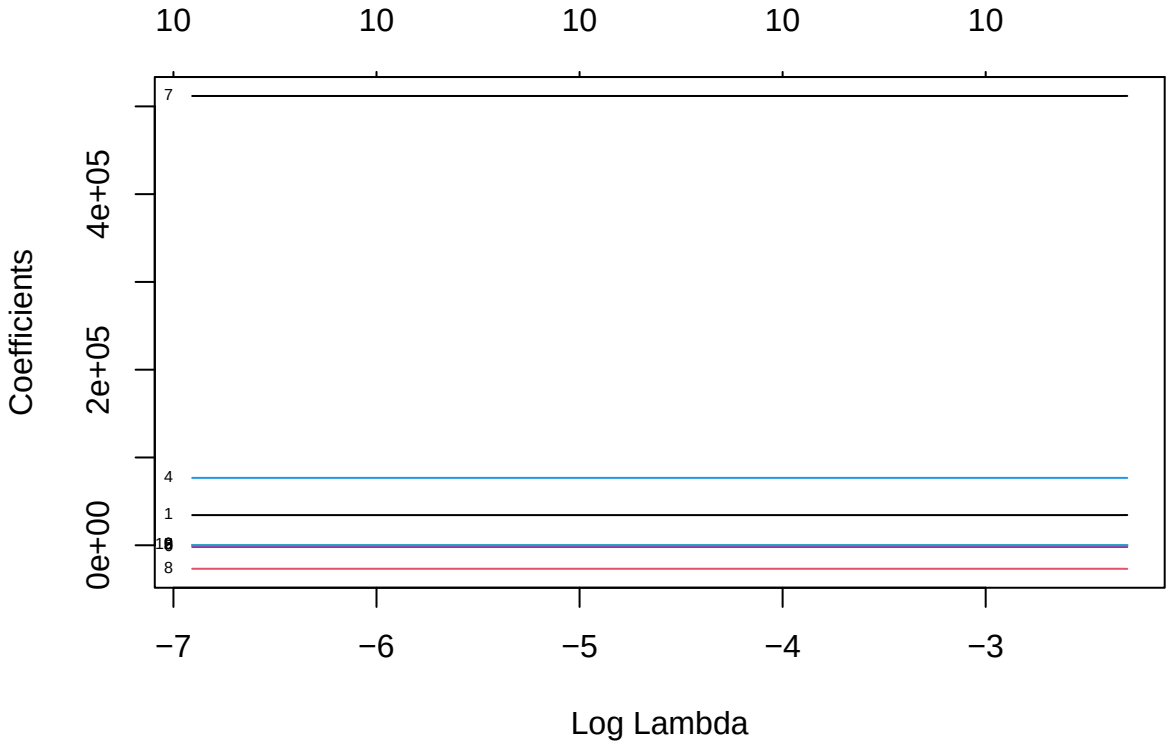
```
## [1] 153719.1
```

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

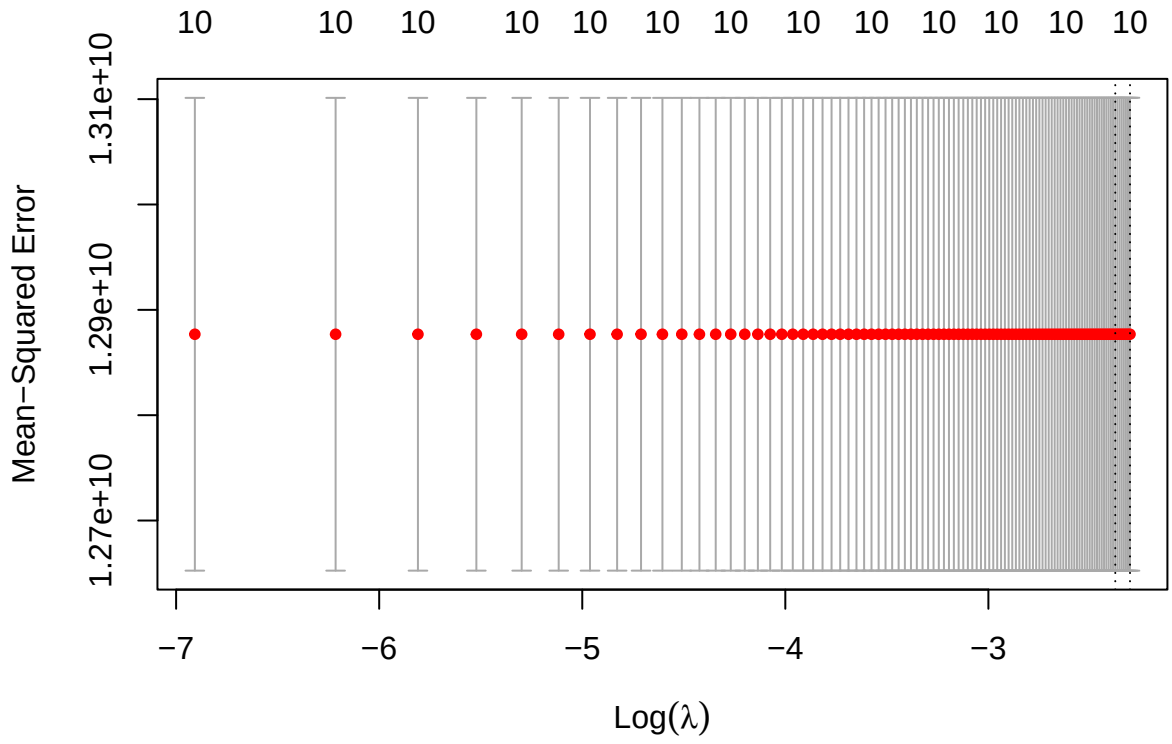
Create Ridge/LASSO models

```
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)
```



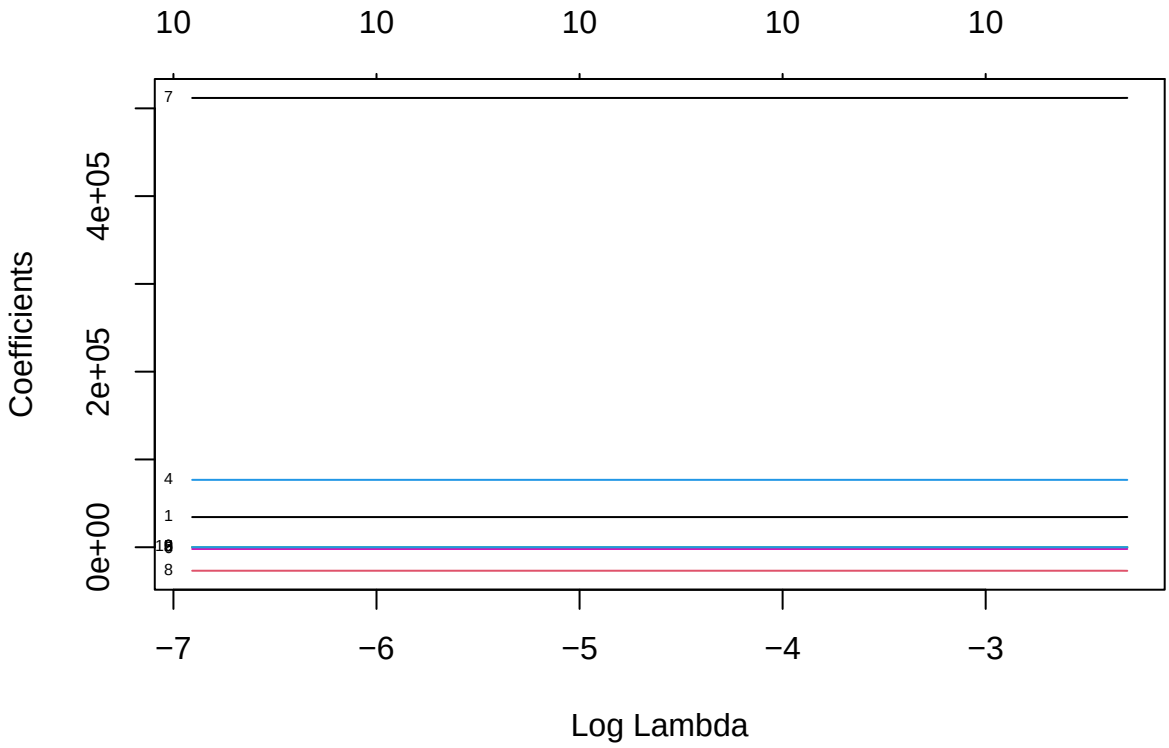

```
plot(ridgeMod)
```



```
ridgeMod$lambda.min
```

```
## [1] 0.093
```

```
# 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   115315.17
## 2 Model2-lstepMod   115336.44
## 3    Model3-rfMod    79267.93
## 4   Model4-pcrMod   219167.56
## 5    Model5-Ridge   115315.16
## 6    Model6-Lasso   115315.42
```