STATS_412_Project_EDA

Lisa Kaunitz, TianShu Fan, Jaehee Jeong

11/20/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(DataExplorer) # eda
library(pls) # pcr
```

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

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
library(glmnet) #cv.glmnet
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-3
```

```
df <- read.csv(file ='kc_house_data.csv')
head(df)
```

```
##           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
```

```
colnames(df)
```

```
##  [1] "id"            "date"          "price"         "bedrooms"
##  [5] "bathrooms"     "sqft_living"   "sqft_lot"      "floors"
##  [9] "waterfront"    "view"          "condition"     "grade"
## [13] "sqft_above"    "sqft_basement" "yr_built"      "yr_renovated"
## [17] "zipcode"       "lat"           "long"          "sqft_living15"
## [21] "sqft_lot15"
```

```
dim(df)
```

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

```
# Check missing values
sum(is.na(df))
```

```
## [1] 0
```

```
str(df)
```

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : chr  "20141013T000000" "20141209T000000" "20150225T000000" "20141209T000000" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

```
# Change date from chr to date format
```

```r
df$date <- str_sub(df$date,1,8)
df$date<-as.Date(df$date, format =  "%Y%m%d")

str(df)
```

```
## 'data.frame':    21613 obs. of  21 variables:
##  $ id           : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date         : Date, format: "2014-10-13" "2014-12-09" ...
##  $ price        : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms     : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms    : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living  : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot     : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors       : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition    : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade        : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above   : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement: int  0 400 0 910 0 1530 0 0 730 0 ...
##  $ yr_built     : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode      : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat          : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long         : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15: int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15   : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
```

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

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

str(df)
```

```
## 'data.frame':    21613 obs. of  22 variables:
##  $ id                 : num  7.13e+09 6.41e+09 5.63e+09 2.49e+09 1.95e+09 ...
##  $ date               : Date, format: "2014-10-13" "2014-12-09" ...
##  $ price              : num  221900 538000 180000 604000 510000 ...
##  $ bedrooms           : int  3 3 2 4 3 4 3 3 3 3 ...
##  $ bathrooms          : num  1 2.25 1 3 2 4.5 2.25 1.5 1 2.5 ...
##  $ sqft_living        : int  1180 2570 770 1960 1680 5420 1715 1060 1780 1890 ...
##  $ sqft_lot           : int  5650 7242 10000 5000 8080 101930 6819 9711 7470 6560 ...
##  $ floors             : num  1 2 1 1 1 1 2 1 1 2 ...
##  $ waterfront         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ view               : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ condition          : int  3 3 3 5 3 3 3 3 3 3 ...
##  $ grade              : int  7 7 6 7 8 11 7 7 7 7 ...
##  $ sqft_above         : int  1180 2170 770 1050 1680 3890 1715 1060 1050 1890 ...
##  $ sqft_basement      : int  0 400 0 910 0 1530 0 0 730 0 ...
```

```
##  $ yr_built           : int  1955 1951 1933 1965 1987 2001 1995 1963 1960 2003 ...
##  $ yr_renovated       : int  0 1991 0 0 0 0 0 0 0 0 ...
##  $ zipcode            : int  98178 98125 98028 98136 98074 98053 98003 98198 98146 98038 ...
##  $ lat                : num  47.5 47.7 47.7 47.5 47.6 ...
##  $ long               : num  -122 -122 -122 -122 -122 ...
##  $ sqft_living15      : int  1340 1690 2720 1360 1800 4760 2238 1650 1780 2390 ...
##  $ sqft_lot15         : int  5650 7639 8062 5000 7503 101930 6819 9711 8113 7570 ...
##  $ sqft_basement_yesno: num  0 1 0 1 0 1 0 0 1 0 ...
```

```
#DataExplorer::create_report(df)
```

```
# split the data to a train set(80%) and a test set(20%)
set.seed(101)
train = sample(1:nrow(df), 4*nrow(df) / 5)

# We don't want to use ID as a factor
df_train <- df[train,c(2:22)]
df_test <- df[-train,c(2:22)]

# define a rmse function
rmse <- function(y_hat, y) sqrt(mean((y_hat - y)^2))

################################################################################
# Model1 - Linear regression
################################################################################

m1_lm <- lm(price ~ ., data=df_train)
# I am not sure why sqft_basement factor is not applicable
summary(m1_lm)
```

```
##
## Call:
## lm(formula = price ~ ., data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1309342   -99390    -9133    77774  4318824
##
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          4.969e+06  3.305e+06   1.503  0.13276
## date                 1.095e+02  1.362e+01   8.045 9.19e-16 ***
## bedrooms            -3.666e+04  2.103e+03 -17.431  < 2e-16 ***
## bathrooms            4.280e+04  3.666e+03  11.676  < 2e-16 ***
## sqft_living          1.568e+02  6.908e+00  22.699  < 2e-16 ***
## sqft_lot             9.569e-02  5.134e-02   1.864  0.06232 .
## floors               7.896e+03  4.024e+03   1.962  0.04976 *
## waterfront           5.943e+05  1.991e+04  29.856  < 2e-16 ***
## view                 5.140e+04  2.403e+03  21.392  < 2e-16 ***
## condition            3.108e+04  2.639e+03  11.776  < 2e-16 ***
## grade                9.380e+04  2.418e+03  38.794  < 2e-16 ***
## sqft_above           2.315e+01  7.645e+00   3.028  0.00247 **
## sqft_basement               NA         NA      NA       NA
## yr_built            -2.596e+03  8.128e+01 -31.934  < 2e-16 ***
## yr_renovated         2.142e+01  4.138e+00   5.177 2.28e-07 ***
```

```
## zipcode             -5.774e+02  3.690e+01 -15.646  < 2e-16 ***
## lat                  6.007e+05  1.199e+04  50.094  < 2e-16 ***
## long                -2.104e+05  1.473e+04 -14.285  < 2e-16 ***
## sqft_living15        2.688e+01  3.858e+00   6.967 3.35e-12 ***
## sqft_lot15          -4.041e-01  8.038e-02  -5.028 5.00e-07 ***
## sqft_basement_yesno -6.583e+03  5.888e+03  -1.118  0.26359
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201500 on 17270 degrees of freedom
## Multiple R-squared:  0.7001, Adjusted R-squared:  0.6998
## F-statistic:  2122 on 19 and 17270 DF,  p-value: < 2.2e-16
```

```
##############################################################################
# Model2 - Linear regression using step function
##############################################################################

m2_lm_step <- step(lm(price ~.,data=df_train),direction = "backward",trace = F)
summary(m2_lm_step)
```

```
##
## Call:
## lm(formula = price ~ date + bedrooms + bathrooms + sqft_living +
##     sqft_lot + floors + waterfront + view + condition + grade +
##     sqft_above + yr_built + yr_renovated + zipcode + lat + long +
##     sqft_living15 + sqft_lot15, data = df_train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1314252   -99171    -9125    77880  4328595
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.347e+06  3.288e+06   1.626   0.1039
## date           1.094e+02  1.362e+01   8.035 9.96e-16 ***
## bedrooms      -3.664e+04  2.103e+03 -17.424  < 2e-16 ***
## bathrooms      4.230e+04  3.638e+03  11.627  < 2e-16 ***
## sqft_living    1.514e+02  4.904e+00  30.866  < 2e-16 ***
## sqft_lot       9.502e-02  5.133e-02   1.851   0.0642 .
## floors         7.987e+03  4.023e+03   1.985   0.0471 *
## waterfront     5.949e+05  1.990e+04  29.892  < 2e-16 ***
## view           5.151e+04  2.401e+03  21.452  < 2e-16 ***
## condition      3.119e+04  2.637e+03  11.828  < 2e-16 ***
## grade          9.357e+04  2.409e+03  38.838  < 2e-16 ***
## sqft_above     2.969e+01  4.912e+00   6.045 1.52e-09 ***
## yr_built      -2.594e+03  8.126e+01 -31.917  < 2e-16 ***
## yr_renovated   2.155e+01  4.137e+00   5.210 1.91e-07 ***
## zipcode       -5.794e+02  3.686e+01 -15.719  < 2e-16 ***
## lat            6.001e+05  1.198e+04  50.093  < 2e-16 ***
## long          -2.092e+05  1.469e+04 -14.241  < 2e-16 ***
## sqft_living15  2.681e+01  3.857e+00   6.950 3.79e-12 ***
## sqft_lot15    -4.037e-01  8.038e-02  -5.022 5.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

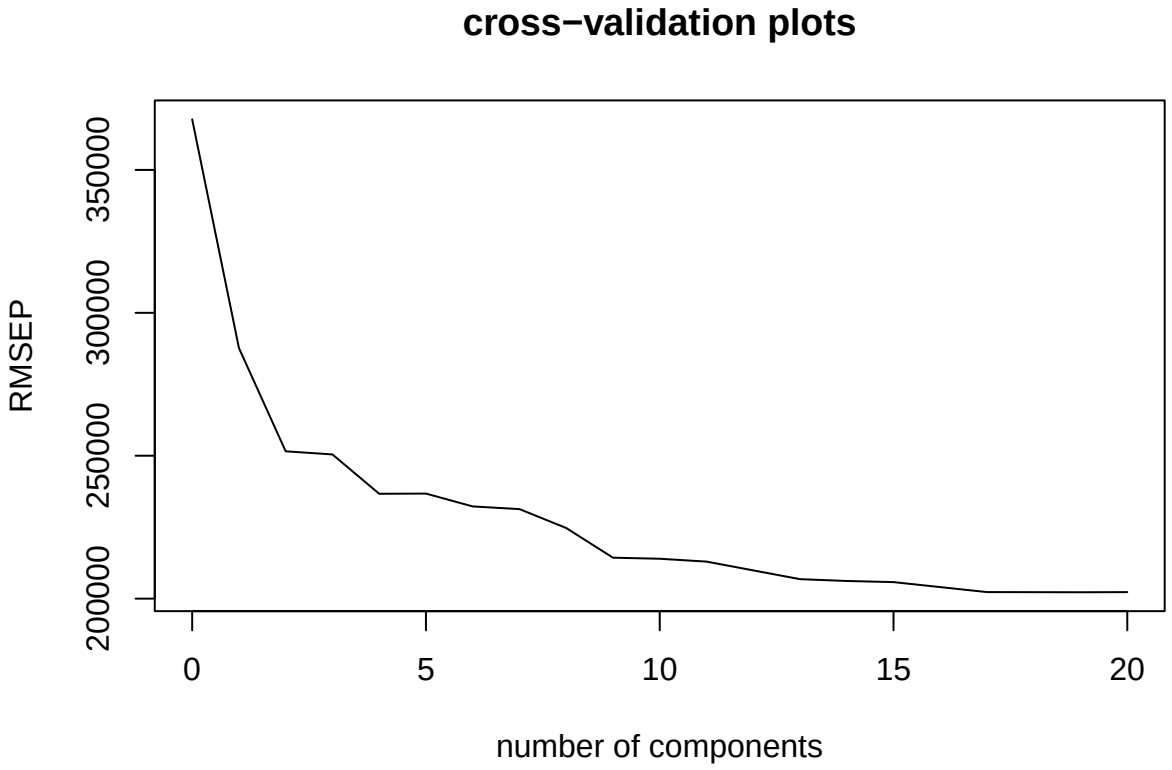
```
## Residual standard error: 201500 on 17271 degrees of freedom
## Multiple R-squared:  0.7001, Adjusted R-squared:  0.6998
## F-statistic:  2240 on 18 and 17271 DF,  p-value: < 2.2e-16
```

```
#############################################################################
# Model3 - PCR
#############################################################################

set.seed(101)
m3_pcr <- pcr(price ~ ., data=df_train,scale = T, validation="CV"
              , ncomp=20, segments = 10)
summary(m3_pcr)
```

```
## Data:    X dimension: 17290 20
##  Y dimension: 17290 1
## Fit method: svdpc
## Number of components considered: 20
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          367695   287689   251542   250449   236691   236755   232255
## adjCV       367695   287676   251523   250430   236672   236739   232243
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      231319   224706   214326    213947    212954    209879    206814
## adjCV   231315   224688   214300    213915    212927    209851    206783
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       206179    205778    204029    202288    202254    202233    202280
## adjCV    206154    205888    203994    202246    202209    202186    202224
##
## TRAINING: % variance explained
##        1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X        26.13    39.65    48.90    56.07    62.10    67.46    72.43    77.27
## price    38.87    53.28    53.71    58.65    58.65    60.21    60.55    62.81
##        9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X        81.58     84.92     88.09     90.62     92.65     94.33     95.82
## price    66.17     66.31     66.60     67.58     68.53     68.73     68.90
##        16 comps  17 comps  18 comps  19 comps  20 comps
## X         97.15     98.28     99.26    100.00    100.00
## price     69.46     69.97     70.00     70.01     70.01
```

```
# plot of CV error 1 computing manually using the class notes
pcrCV <- RMSEP(m3_pcr, estimate="CV")
plot(pcrCV, main = "cross-validation plots")
```

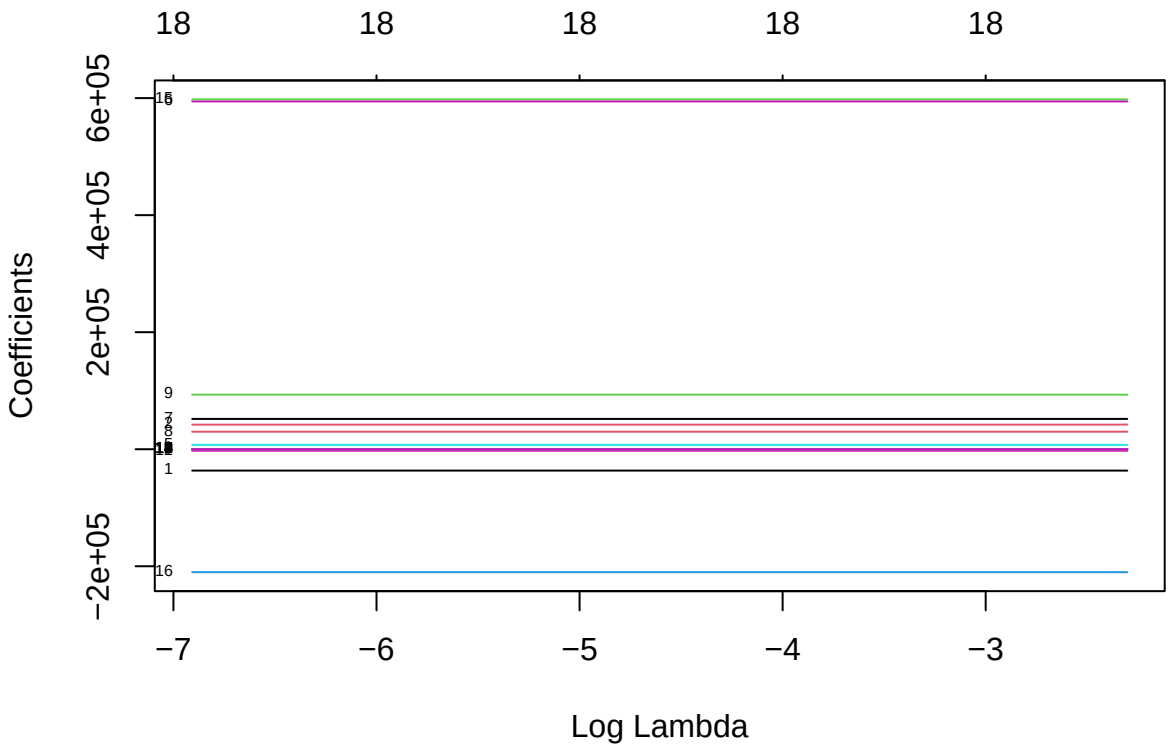



```
which.min(pcrCV$val) -1 # remove the intercept
```

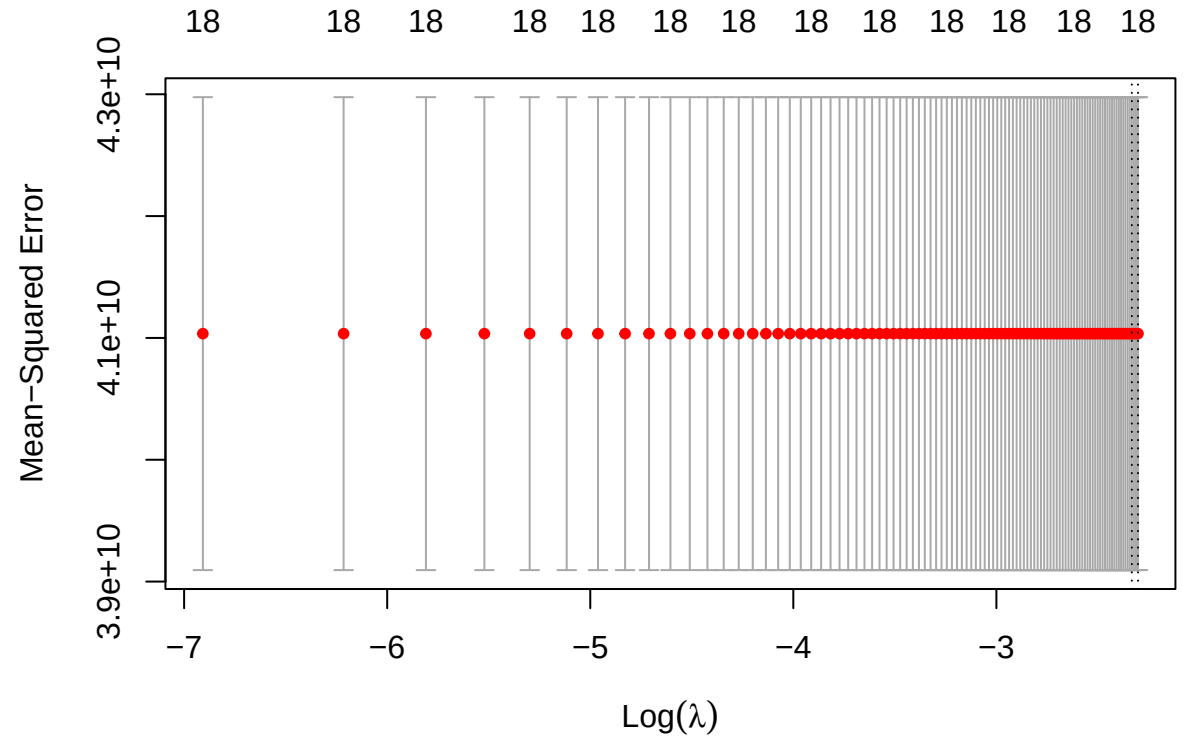
```
## [1] 19
```

```
################################################################################
# Model4 - Ridge regression
################################################################################
lambdas_to_try = lambda=seq(0.001,0.1, by=0.001)
set.seed(101)

# There is an error because of the date format. We need to figure this out.
# For now, I didn't include date factor.
m4_ridge <- cv.glmnet(as.matrix(df_train[,c(3:20)]), df_train$price, alpha = 0
                      , lambda = lambdas_to_try
                      ,standardize = TRUE, nfolds =10)
plot(m4_ridge$glmnet.fit,xvar = "lambda", label = T)
```




```
plot(m4_ridge)
```

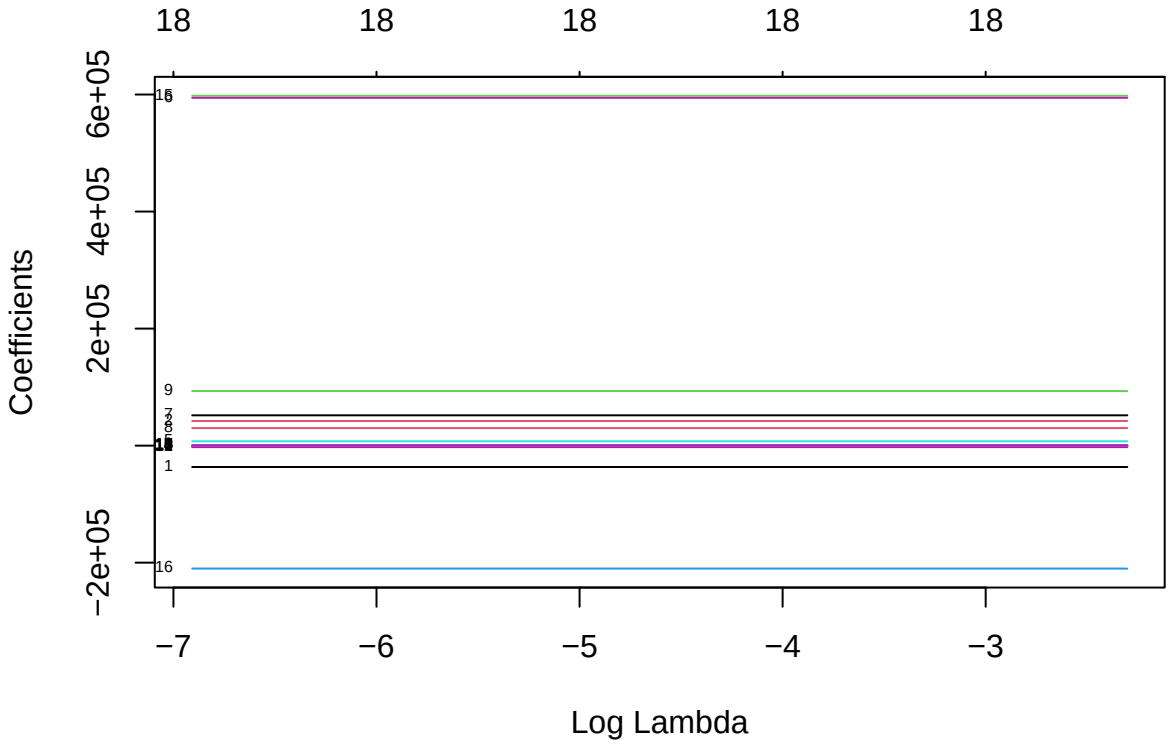



```
m4_ridge$lambda.min
```

```
## [1] 0.097
```

```
##############################################################################
# Model5 - Lasso
```

```
#############################################################################
set.seed(101)
m5_lasso <- cv.glmnet(as.matrix(df_train[,c(3:20)]), df_train$price, alpha = 1,
                      lambda = lambdas_to_try,
                      standardize = TRUE, nfolds = 10)
plot(m5_lasso$glmnet.fit,xvar = "lambda", label = T)
```




```
m5_lasso$lambda.min
```

```
## [1] 0.097
```

```
#############################################################################
# Create a df showing all the rmse values
#############################################################################
rmse_colnames<-c("Model1-lm","Model2-lm_step","Model3-pcr", "Model4-Ridge"
                 , "Model5-Lasso")
rmse_result <-c( rmse(predict(m1_lm, df_test), df_test$price)
             ,rmse(predict(m2_lm_step, df_test), df_test$price)
             ,rmse(predict(m3_pcr, df_test, ncomp = 15), df_test$price)
             ,rmse(predict(m4_ridge, newx = as.matrix(df_test[,c(3:20)])
                   , s=m4_ridge$lambda.min)
                   ,df_test$price)
             ,rmse(predict(m5_lasso, newx = as.matrix(df_test[,c(3:20)])
                   , s=m5_lasso$lambda.min)
                   ,df_test$price)
             )
```

```
## Warning in predict.lm(m1_lm, df_test): prediction from a rank-deficient fit may
## be misleading
```

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

```
##    rmse_colnames rmse_result
## 1      Model1-lm    198476.7
## 2 Model2-lm_step    198472.7
## 3     Model3-pcr    203629.0
## 4   Model4-Ridge    199106.9
## 5   Model5-Lasso    199106.9
```