DS2 Midterm Project

Siying Li

3/31/2020

1 Data Visualization

```
house = read.csv(file = "train.csv", stringsAsFactors = FALSE)
# Do not import strings as factors, since the ultimate goal is to transfer all variables to numeric.

dim(house)
```

```
## [1] 1460   81
```

```
house = house %>%
  dplyr::select(-Id) %>%
  janitor::clean_names()
```

The house dataset consists of both integer and character variables. Most of the categorical variables are ordinal. There is a total of 81 variables, and the last column is our response.

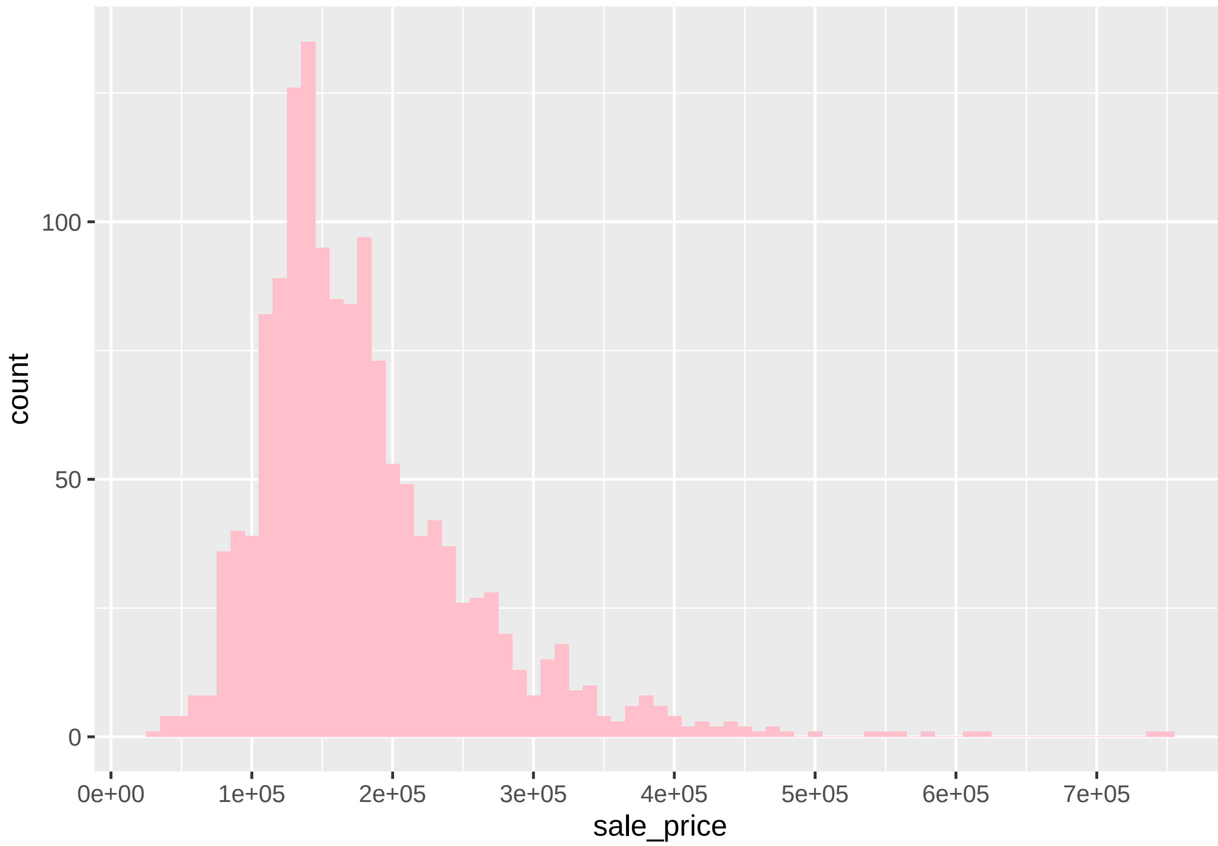
1.1 Data Cleaning

```
# Get rid of virables with 500 plus missing values
missing =
  colSums(sapply(house, is.na)) %>%
  as.data.frame() %>%
  mutate(variable = colnames(house)) %>%
  filter(. > 500) %>%
  pull(variable)

house =
  house %>%
  select(-missing) %>%
  filter(lot_area < 100000) # filter out house with extreme lot size to prevent outliers.
```

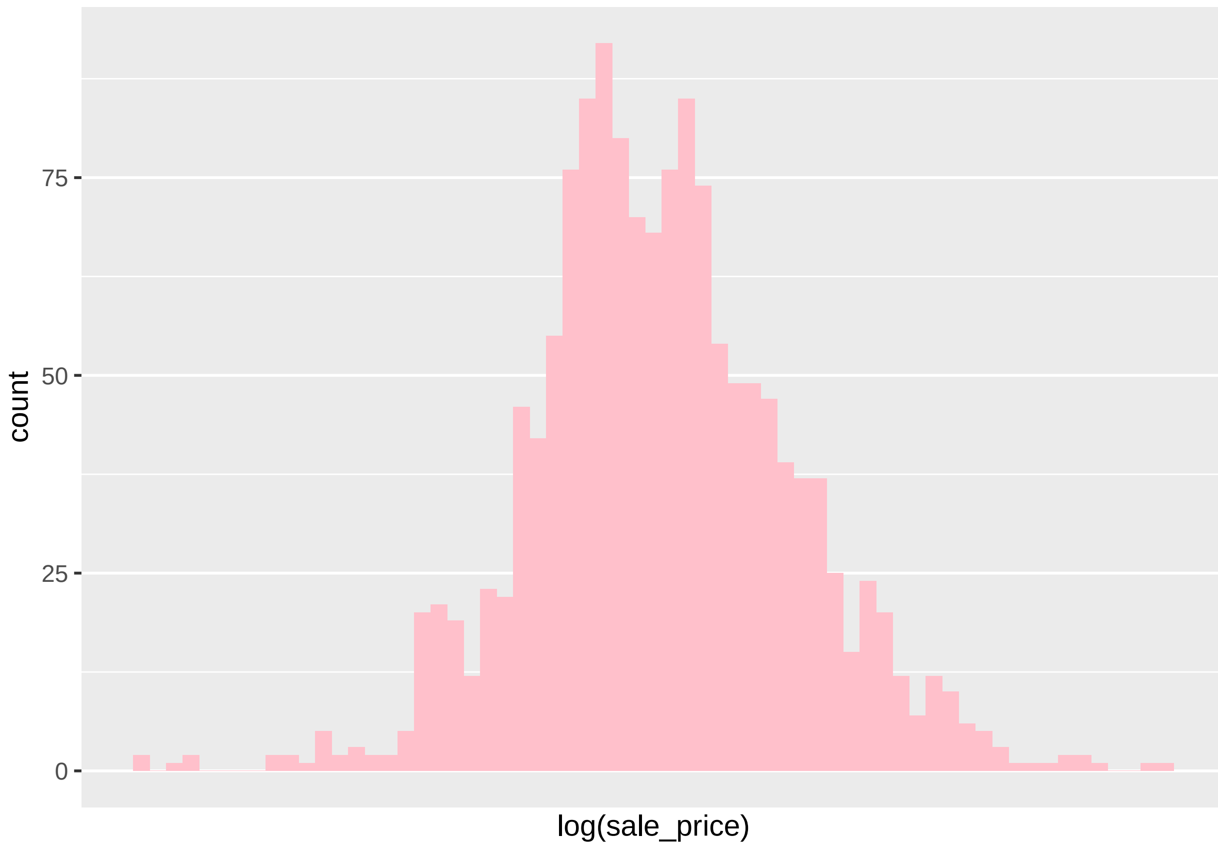
1.2 Response Variable

```
# Response variable distribution
ggplot(data=house, aes(x=sale_price)) +
        geom_histogram(fill="pink", binwidth = 10000) +
        scale_x_continuous(breaks= seq(0, 800000, by=100000))
```




```r
# Right skewed

# log transform on y
ggplot(data=house, aes(x=log(sale_price))) +
        geom_histogram(fill="pink", binwidth = 0.05) +
        scale_x_continuous(breaks= seq(0, 800000, by=100000))
```



1.3 Numeric Predictors

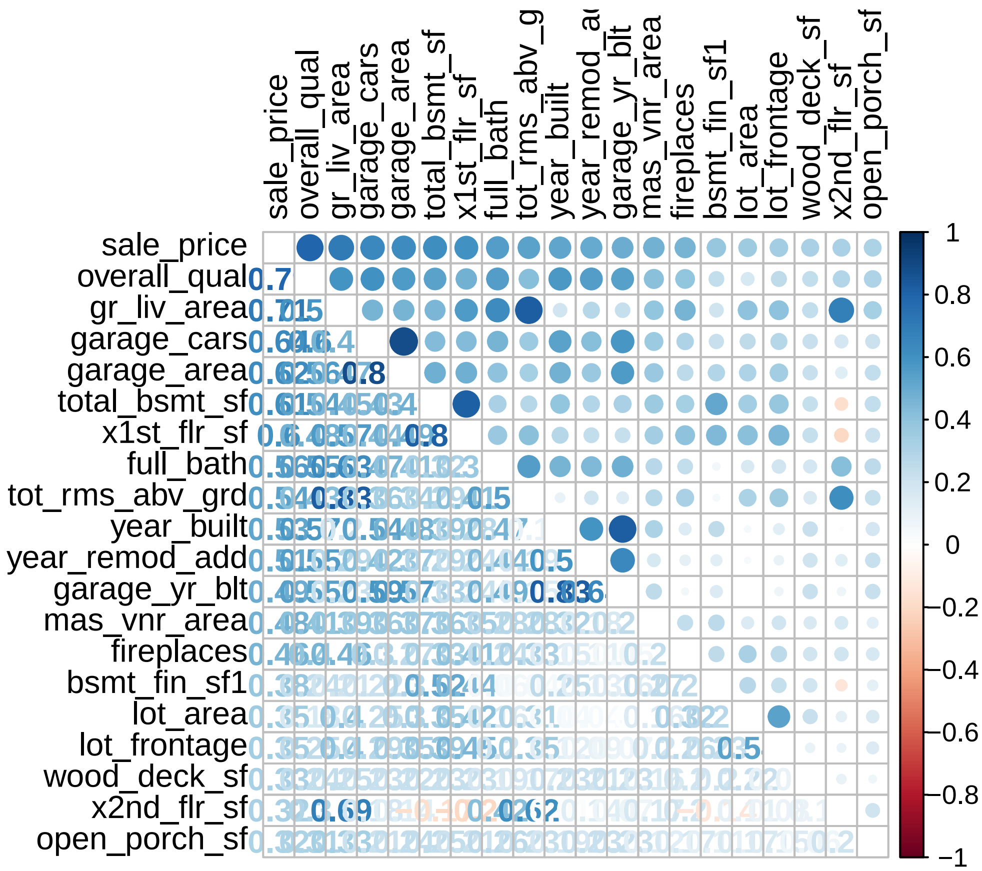
```
label_num <- sapply(house, is.numeric) #set numeric variables as TRUE
num_var <- house[, label_num] # get all the numeric variables
```

There is a total of 37 numeric variables.

```
# Test Correlation
corr_num <- cor(num_var, use = "pairwise.complete.obs") #correlations of all numeric variables

#sort on decreasing correlations with sale price
corr_sort <- as.matrix(sort(corr_num[,'sale_price'], decreasing = TRUE))
#eliminate low correlation variables
high_corr <- names(which(apply(corr_sort, 1, function(x) abs(x)>0.3)))
corr_num <- corr_num[high_corr, high_corr]

corrplot.mixed(corr_num, tl.col="black", tl.pos = "lt")
```




```r
# 'Lot_frontage' variable represent linear feet of street connected to property. It contains NAs, which
house$lot_frontage[is.na(house$lot_frontage)] = 0

# There is a "1st floor sq footage" variable and a "2nd floor square footage" variable.
# It would be better to sum them up to total square footage for better pridiction.
house =
  house %>%
  mutate(total_sq = x1st_flr_sf + x2nd_flr_sf) %>%
  select(-x1st_flr_sf, -x2nd_flr_sf)

# Convert year and month sold to factor variable
house$mo_sold = as.factor(house$mo_sold)
house$yr_sold = as.factor(house$yr_sold)
```

1.4 Categorical Predictors

```r
char_num <- names(house[,sapply(house, is.character)])
length(char_num)
```

```
## [1] 38
```

```r
# Oridinal Predictors
house[house == "Ex"] = 5
house[house == "Gd"] = 4
house[house == "TA"] = 3
house[house == "Fa"] = 2
house[house == "Po"] = 1
```

```r
house[house == "NA"] = 3

house$exter_qual = as.numeric(house$exter_qual)
house$exter_qual = as.numeric(house$exter_qual)
house$bsmt_qual = as.numeric(house$bsmt_qual)
house$bsmt_cond = as.numeric(house$bsmt_cond)
house$heating_qc= as.numeric(house$heating_qc)
house$kitchen_qual= as.numeric(house$kitchen_qual)
house$garage_qual= as.numeric(house$garage_qual)
house$garage_cond= as.numeric(house$garage_cond)


# LotShape: General shape of property
house$lot_shape[house$lot_shape == "Reg"] = 4 # Regular
house$lot_shape[house$lot_shape == "IR1"] = 3 # Slightly irregular
house$lot_shape[house$lot_shape == "IR2"] = 2 # Moderately Irregular
house$lot_shape[house$lot_shape == "IR3"] = 1 # Irregular
house$lot_shape = as.numeric(house$lot_shape)

# Utilities: Type of utilities available
house$utilities[house$utilities == "AllPub"] = 4 # All public Utilities (E,G,W,& S)
house$utilities[house$utilities == "NoSewr"] = 3 # Electricity, Gas, and Water (Septic Tank)
house$utilities[house$utilities == "NoSeWa"] = 2 # Electricity and Gas Only
house$utilities[house$utilities == "ELO"] = 1    # Electricity only
house$utilities = as.numeric(house$utilities)

# LandSlope: Slope of property
house$land_slope[house$land_slope == "Gtl"] = 3 # Gentle slope
house$land_slope[house$land_slope == "Mod"] = 2 # Moderate Slope
house$land_slope[house$land_slope == "Sev"] = 1 # Severe Slope
house$land_slope = as.numeric(house$land_slope)

# BsmtExposure: Refers to walkout or garden level walls
house$bsmt_exposure[house$bsmt_exposure == "Av"] = 3 # Average Exposure (split levels or foyers typical
house$bsmt_exposure[house$bsmt_exposure == "Mn"
                    ] = 2 # Mimimum Exposure
house$bsmt_exposure[house$bsmt_exposure == "No"] = 1 # No Exposure
house$bsmt_exposure[house$bsmt_exposure == 0] = 0 # No Exposure
house$bsmt_exposure = as.numeric(house$bsmt_exposure)

# BsmtFinType1: Rating of basement finished area
house$bsmt_fin_type1[house$bsmt_fin_type1 == "GLQ"] = 6 # Good Living Quarters
house$bsmt_fin_type1[house$bsmt_fin_type1 == "ALQ"] = 5 # Average Living Quarters
house$bsmt_fin_type1[house$bsmt_fin_type1 == "BLQ"] = 4 # Below Average Living Quarters
house$bsmt_fin_type1[house$bsmt_fin_type1 == "Rec"] = 3 # Average Rec Room
house$bsmt_fin_type1[house$bsmt_fin_type1 == "LwQ"] = 2 # Low Quality
house$bsmt_fin_type1[house$bsmt_fin_type1 == "Unf"] = 1 # Unfinshed
house$bsmt_fin_type1[house$bsmt_fin_type1 == "NA"] = 3 # No Basement
house$bsmt_fin_type1 = as.numeric(house$bsmt_fin_type1)

# CentralAir: Central air conditioning
house$central_air[house$central_air == "N"] = 0 # No
house$central_air[house$central_air == "Y"] = 1 # Yes
```

```r
house$central_air = as.numeric(house$central_air)

# Functional: Home functionality (Assume typical unless deductions are warranted)
house$functional[house$functional == "Typ"] = 8 # Typical Functionality
house$functional[house$functional == "Min1"] = 7 # Minor Deductions 1
house$functional[house$functional == "Min2"] = 6 # Minor Deductions 2
house$functional[house$functional == "Mod"] = 5 # Moderate Deductions
house$functional[house$functional == "Maj1"] = 4 # Major Deductions 1
house$functional[house$functional == "Maj2"] = 3 # Major Deductions 2
house$functional[house$functional == "Sev"] = 2 # Severely Damaged
house$functional[house$functional == "Sal"] = 1 # Salvage only
house$functional = as.numeric(house$functional)

# GarageFinish: Interior finish of the garage
house$garage_finish[house$garage_finish == "Fin"] = 3 # Finished
house$garage_finish[house$garage_finish == "RFn"] = 2 # Rough Finished
house$garage_finish[house$garage_finish == "Unf"] = 1 # Unfinished
house$garage_finish[house$garage_finish == 3 ] = 0 # No Garage
house$garage_finish = as.numeric(house$garage_finish)

# PavedDrive: Paved driveway
house$paved_drive[house$paved_drive == "Y"] = 3 # Paved
house$paved_drive[house$paved_drive == "P"] = 2 # Partial Pavement
house$paved_drive[house$paved_drive == "N"] = 1 # Dirt/Gravel
house$paved_drive = as.numeric(house$paved_drive)

# sapply(house, class)

# Select rest of the character variables and change them to factors.
house[sapply(house, is.character)] <- lapply(house[sapply(house, is.character)],
                                       as.factor)
# Select rest of the integer variables and change them to numeric.
# house[sapply(house, is.integer)] <- lapply(house[sapply(house, is.integer)],
#                                          as.numeric)
```

1.5 Remove near zero variance pridictors and rows contaning NA

```r
# There are many variables that contains many zeros. Use a 95% cutoff for the percentage of distinct va
near_zero =
  house %>%
  nearZeroVar(names = TRUE, freqCut = 75/25)

house =
  house %>%
  select(-near_zero) %>%
  drop_na()
```

Finally, numer of numeric variable is 31. And the numer of factor variable is 9.

2 Model Fitting

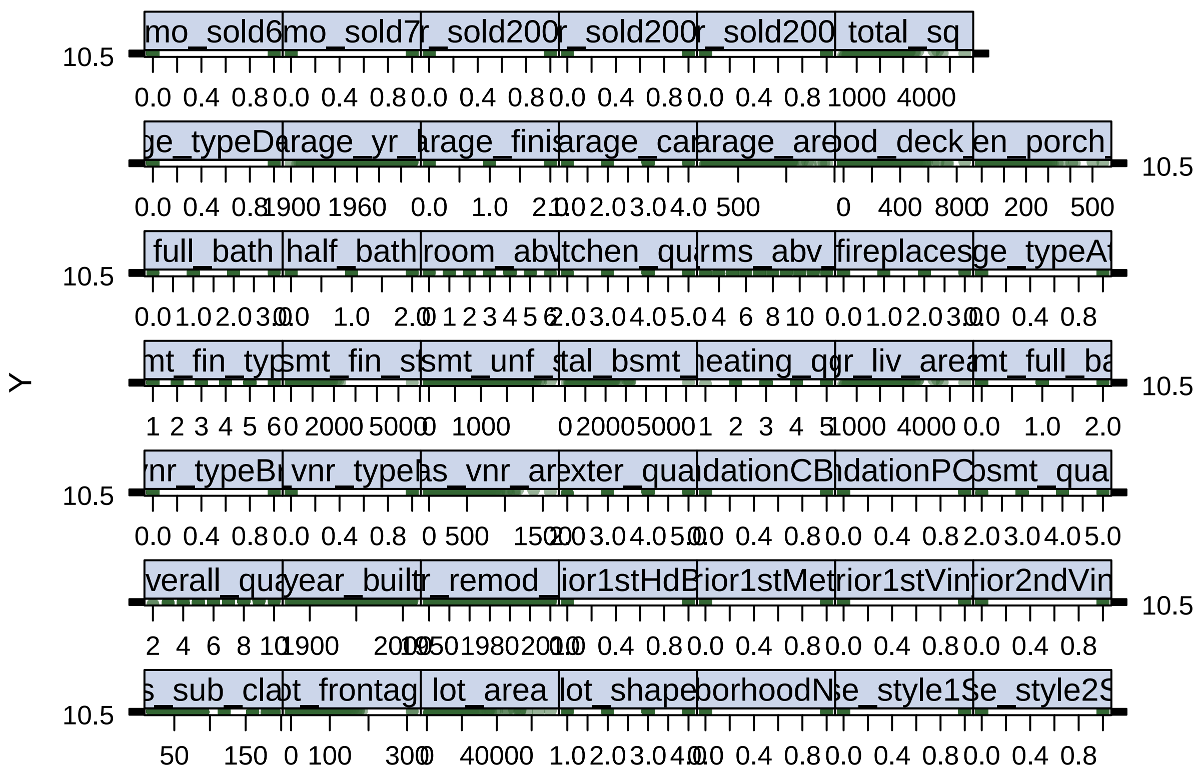
```r
x = model.matrix(sale_price~., house) [,-1]
y = log(house$sale_price)
```

```r
# remove colinear
linear_combo = findLinearCombos(x)
x = x[, -linear_combo$remove]

# remove near zero variance
near_zero_x = x %>% nearZeroVar(names = TRUE, freqCut = 85/15)
x = as.data.frame(x)
x = x %>% select(-near_zero_x)
x = data.matrix(x)

ctrl1 = trainControl(method = "repeatedcv", repeats = 5)
```

```r
theme1 <- trellis.par.get()
theme1$plot.symbol$col <- rgb(.2, .4, .2, .5)
theme1$plot.symbol$pch <- 16
theme1$plot.line$col <- rgb(.8, .1, .1, 1)
theme1$plot.line$lwd <- 2
theme1$strip.background$col <- rgb(.0, .2, .6, .2)
trellis.par.set(theme1)
featurePlot(x, y, plot = "scatter", labels = c("","Y"),
            type = c("p"), layout = c(7, 7))
```



2.1 Multiple Linear Regression

```r
set.seed(2)

lm.fit = train(x, y, method = "lm", trControl = ctrl1, preProcess = c("center", "scale"))
```

```r
# MSE
lm.fit$results$RMSE
```

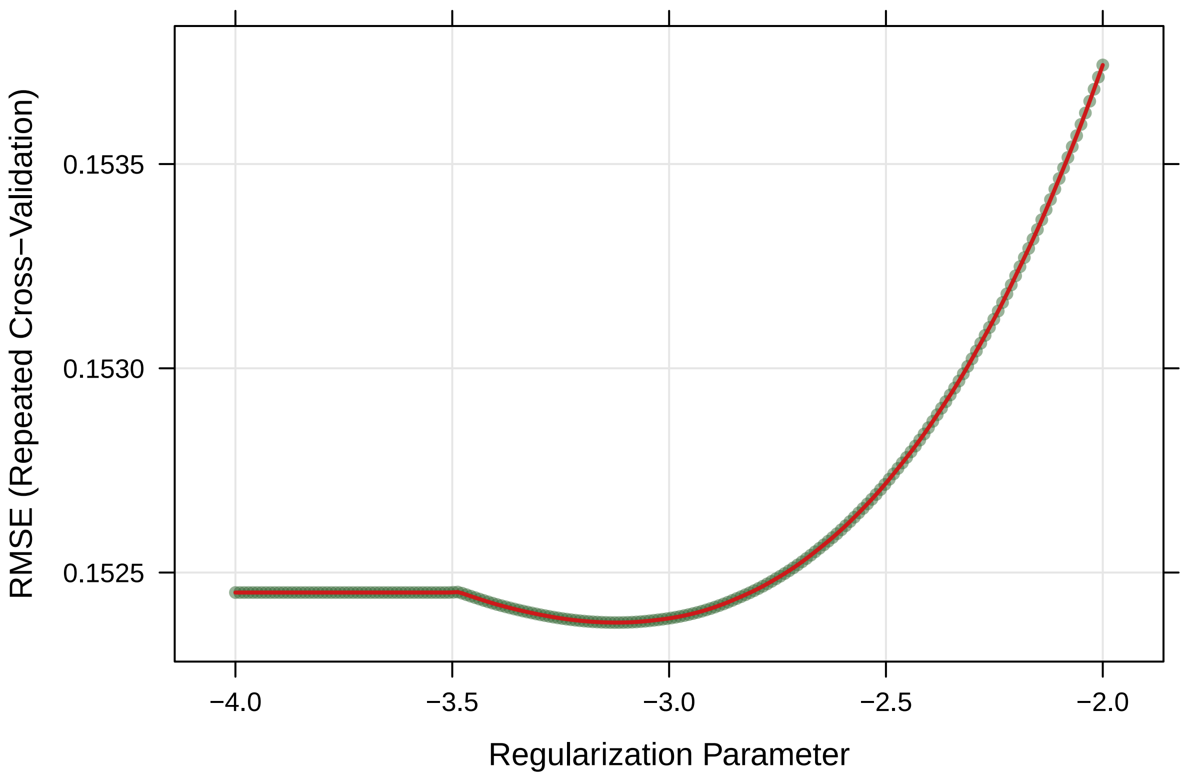
```
## [1] 0.1539649
```

2.2 Ridge Regression

```r
set.seed(2)

ridge.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(-4, -2, length=200))),
                   preProc = c("center", "scale"),
                   trControl = ctrl1)

plot(ridge.fit, xTrans = function(x) log(x))
```




```r
# ridge.fit$results$RMSE

ridge.fit$bestTune
```

```
##    alpha     lambda
## 89     0 0.04435287
```

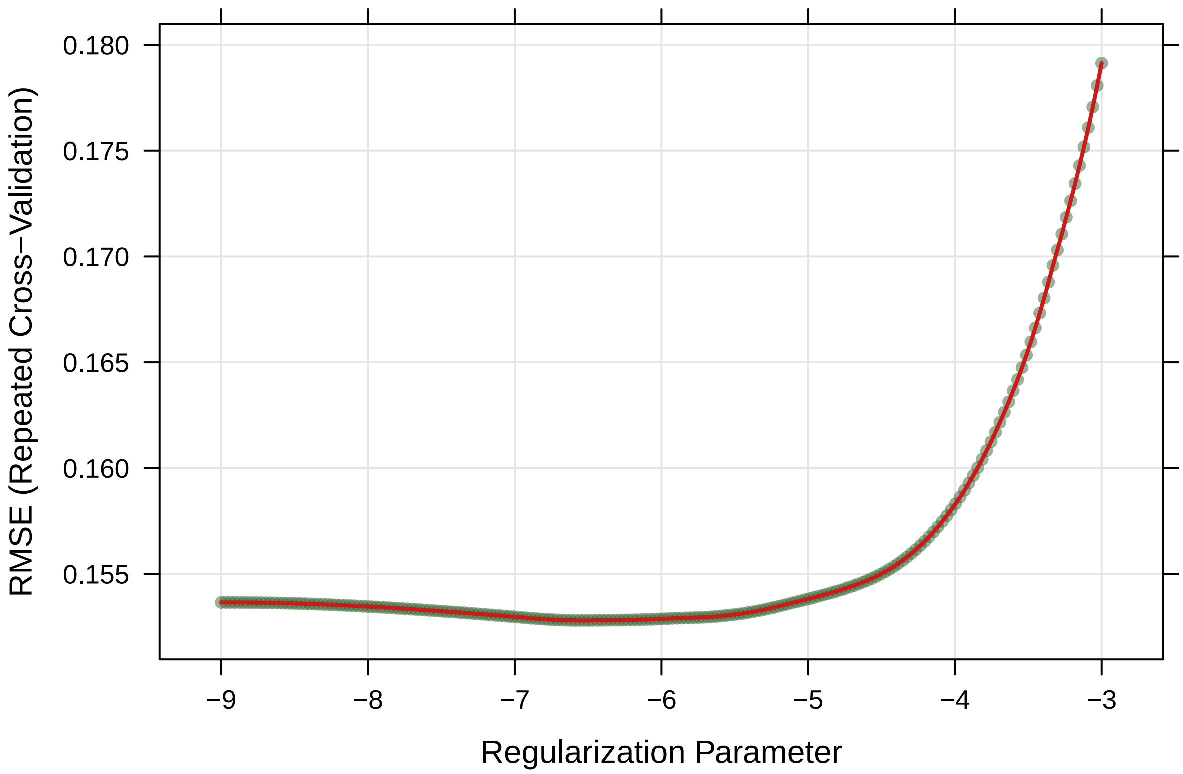
```r
# coef(ridge.fit$finalModel,ridge.fit$bestTune$lambda)
```

2.3 Lasso

```r
set.seed(2)

lasso.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-9,-3, length=200))),
                   preProc = c("center", "scale"),
                   trControl = ctrl1)

plot(lasso.fit, xTrans = function(x) log(x))
```




```r
lasso.fit$bestTune
```

```
##    alpha      lambda
## 83     1 0.001462456
```

```r
# number of non-zero coefficient
coef = coef(lasso.fit$finalModel,lasso.fit$bestTune$lambda)
nnzero(coef)
```

```
## [1] 39
```

coef(lasso.fit$finalModel,lasso.fit$bestTune$lambda)
```

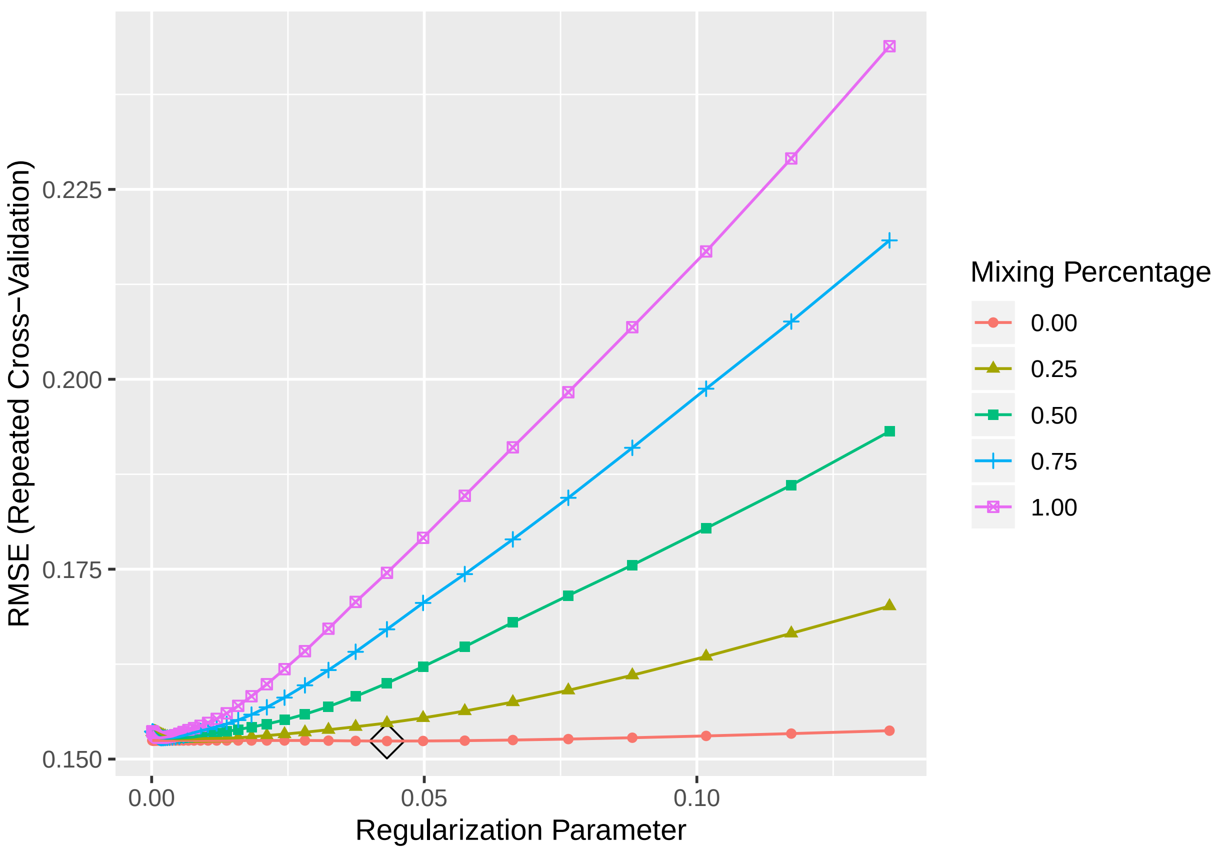
### 2.4 Elastic Net

```r
set.seed(2)

enet.fit <- train(x, y,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 5),
                                         lambda = exp(seq(-9, -2, length = 50))),
                  preProc = c("center", "scale"),
                  trControl = ctrl1)
enet.fit$bestTune
```

```
##    alpha     lambda
## 42     0 0.04315931
```

```r
ggplot(enet.fit, highlight = TRUE)
```



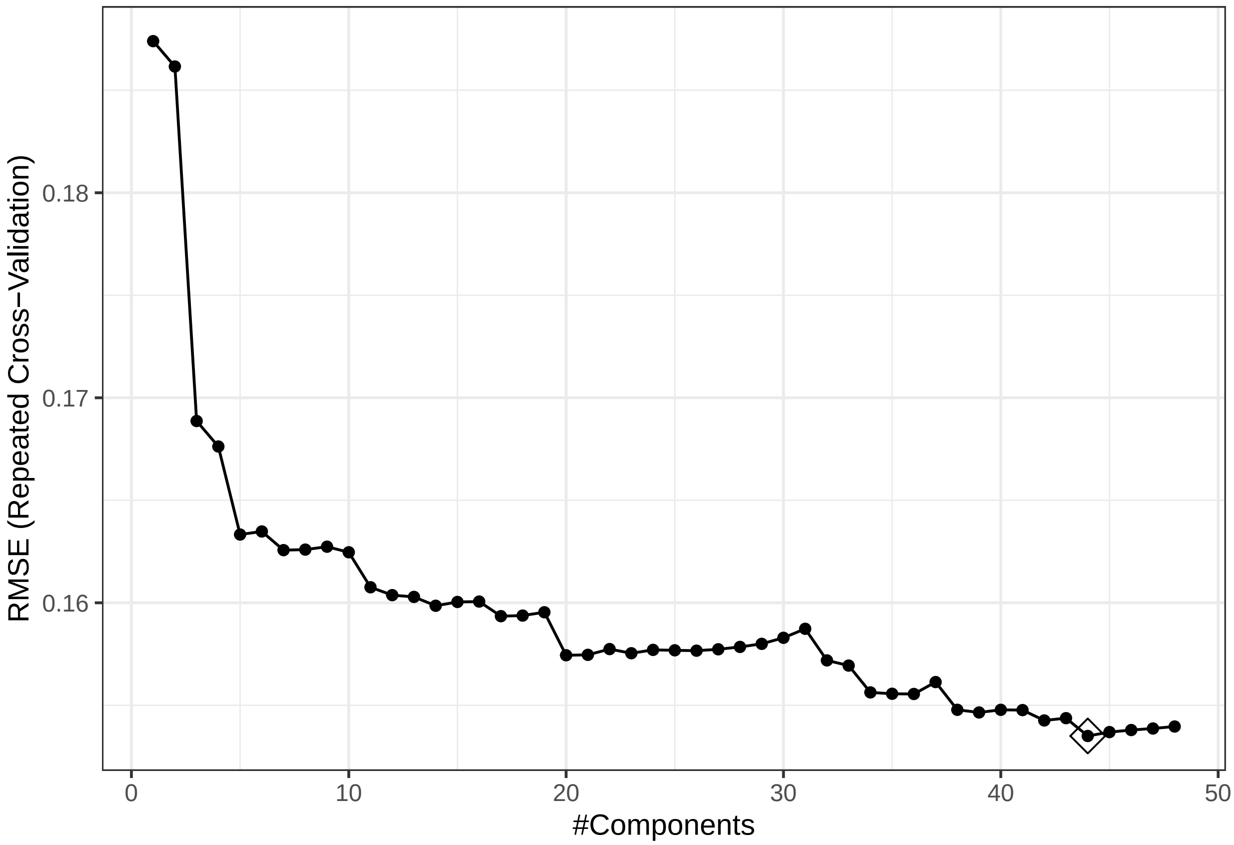
2.5 Principal Component Repression (PCR)

```r
set.seed(2)

pcr.fit <- train(x, y,
                 method = "pcr",
                 tuneGrid  = data.frame(ncomp = 1:ncol(x)),
```
```

```
                 trControl = ctrl1,
                 preProc = c("center", "scale"))

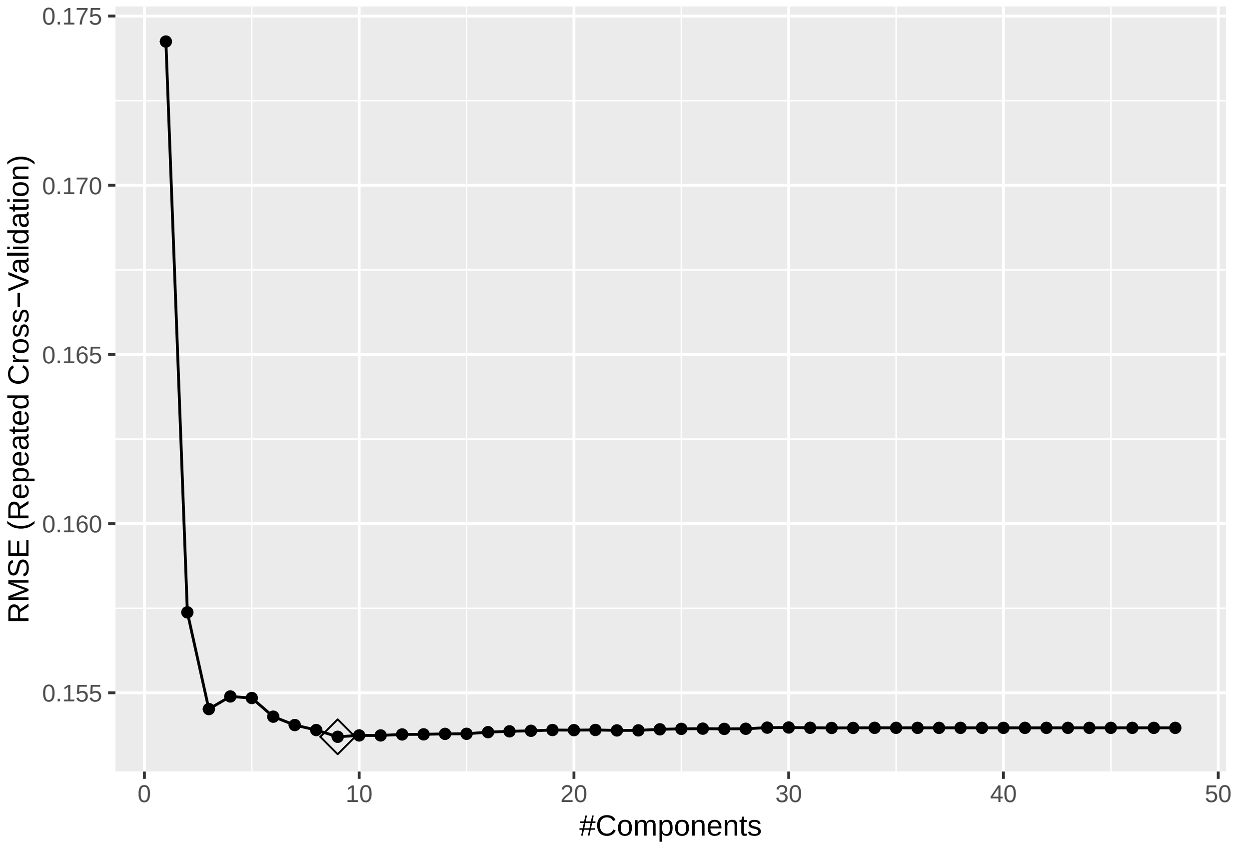
ggplot(pcr.fit, highlight = TRUE) + theme_bw()
```



2.6 Partial Least Squares

```
set.seed(2)
pls.fit <- train(x, y,
                 method = "pls",
                 tuneGrid = data.frame(ncomp = 1:ncol(x)),
                 trControl = ctrl1,
                 preProc = c("center", "scale"))

ggplot(pls.fit, highlight = TRUE)
```



2.7 GAM

```
set.seed(2)
gam.fit <- train(x, y,
                 method = "gam",
                 tuneGrid = data.frame(method = "GCV.Cp", select = c(TRUE,FALSE)),
                 trControl = ctrl1)
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
##
## Attaching package: 'nlme'
```

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

```
## This is mgcv 1.8-28. For overview type 'help("mgcv-package")'.
```

```
gam.fit$finalModel
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## .outcome ~ neighborhoodNAmes + house_style1Story + house_style2Story +
```

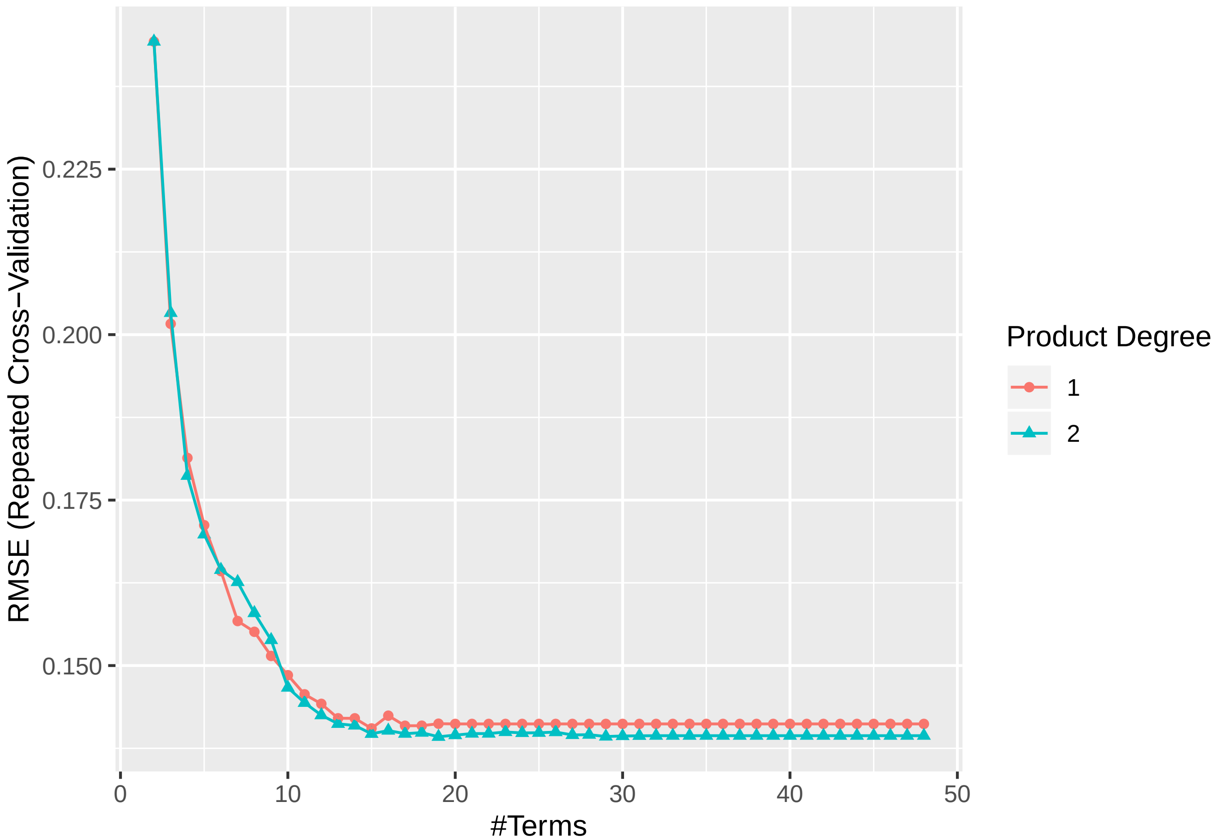
```
##     exterior1stHdBoard + exterior1stMetalSd + exterior1stVinylSd +
##     exterior2ndVinylSd + mas_vnr_typeBrkFace + mas_vnr_typeNone +
##     foundationCBlock + foundationPConc + garage_typeAttchd +
##     garage_typeDetchd + mo_sold6 + mo_sold7 + yr_sold2007 + yr_sold2008 +
##     yr_sold2009 + bsmt_full_bath + half_bath + garage_finish +
##     lot_shape + exter_qual + bsmt_qual + full_bath + kitchen_qual +
##     fireplaces + garage_cars + heating_qc + bsmt_fin_type1 +
##     bedroom_abv_gr + overall_qual + tot_rms_abv_grd + s(ms_sub_class) +
##     s(year_remod_add) + s(garage_yr_blt) + s(lot_frontage) +
##     s(year_built) + s(open_porch_sf) + s(wood_deck_sf) + s(mas_vnr_area) +
##     s(garage_area) + s(bsmt_fin_sf1) + s(total_bsmt_sf) + s(bsmt_unf_sf) +
##     s(gr_liv_area) + s(total_sq) + s(lot_area)
##
## Estimated degrees of freedom:
## 8.49 1.00 6.42 5.41 6.49 1.00 1.00
## 1.00 5.35 3.74 8.40 2.05 8.17 9.00
## 2.81  total = 104.33
##
## GCV score: 0.01549634
```

2.8 Multivariable Adaptive Regression Splines (MARS)

```r
set.seed(2)
mars_grid <- expand.grid(degree = 1:2,
                         nprune = 2:ncol(x))

set.seed(2)
mars.fit <- train(x, y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl1)

ggplot(mars.fit)
```




```r
mars.fit$bestTune
```

```
##    nprune degree
## 65     19      2
```

```r
coef(mars.fit$finalModel)
```

```
##                                 (Intercept)
##                                1.273323e+01
##                           h(overall_qual-6)
##                                8.763895e-02
##                           h(6-overall_qual)
##                               -7.845237e-02
##                            h(total_sq-3140)
##                                3.752975e-04
##                            h(3140-total_sq)
##                               -3.010992e-04
##                        h(1619-bsmt_fin_sf1)
##                               -1.073354e-04
##       h(lot_area-4426) * h(3140-total_sq)
##                                3.959339e-09
##       h(4426-lot_area) * h(3140-total_sq)
##                               -3.808132e-08
##                      h(year_remod_add-2004)
##                                1.803435e-02
##                      h(2004-year_remod_add)
##                               -2.366615e-03
##          open_porch_sf * h(total_sq-3140)
```

```
##                             -3.539570e-06
##      h(1959-year_built) * h(2-fireplaces)
##                             -2.939510e-03
##  h(total_bsmt_sf-796) * h(4-kitchen_qual)
##                             -1.039618e-04
##       h(1959-year_built) * h(1-full_bath)
##                             -7.429495e-02
##    h(3-exter_qual) * h(796-total_bsmt_sf)
##                             -2.755509e-03
##                       h(1025-garage_area)
##                             -1.349736e-04
##      h(1959-year_built) * h(heating_qc-3)
##                              1.246163e-03
## h(total_bsmt_sf-796) * h(garage_finish-1)
##                              1.169567e-04
## h(total_bsmt_sf-796) * h(1-garage_finish)
##                              1.411310e-04
```

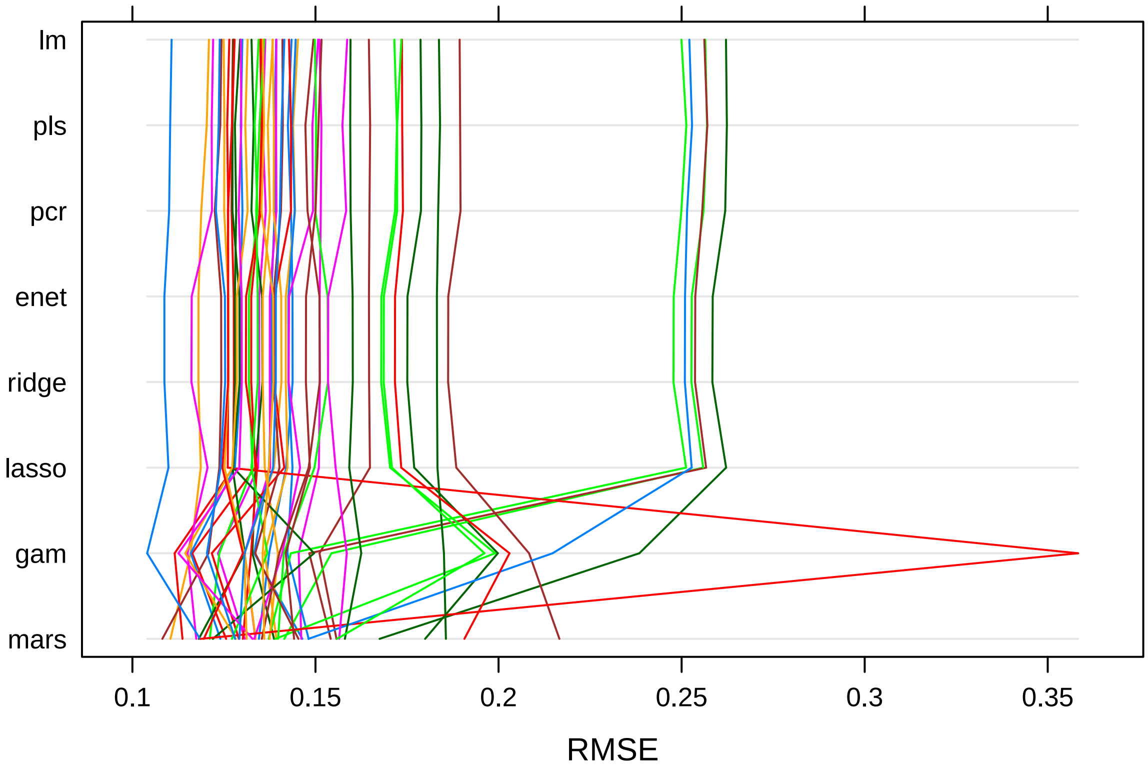
3 Between Model Comparison

```
resamp <- resamples(list(enet = enet.fit, lasso = lasso.fit, ridge = ridge.fit, lm = lm.fit, pcr = pcr.f
summary(resamp)
```

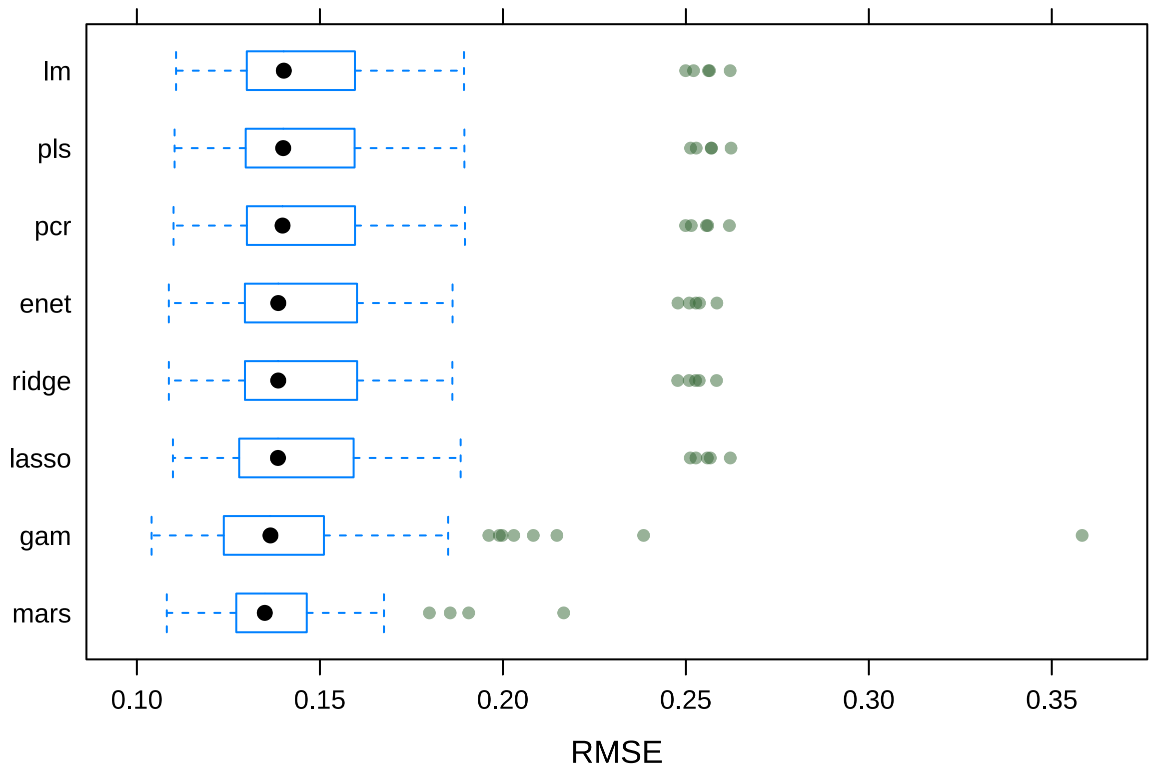
```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: enet, lasso, ridge, lm, pcr, pls, gam, mars
## Number of resamples: 50
##
## MAE
##             Min.    1st Qu.     Median       Mean    3rd Qu.      Max.
## enet  0.08485928 0.09496854 0.09933829 0.10022145 0.10467924 0.1242727
## lasso 0.08640795 0.09553317 0.10053599 0.10106436 0.10611749 0.1254208
## ridge 0.08483697 0.09498603 0.09931401 0.10020243 0.10464688 0.1242280
## lm    0.08767512 0.09715747 0.10221758 0.10279236 0.10817216 0.1270527
## pcr   0.08686837 0.09629902 0.10183647 0.10232126 0.10766447 0.1270487
## pls   0.08738633 0.09676378 0.10202907 0.10255634 0.10747977 0.1279280
## gam   0.07855074 0.08781370 0.09404797 0.09488848 0.09996202 0.1226690
## mars  0.07977574 0.09031136 0.09501718 0.09514837 0.10032268 0.1139879
##       NA's
## enet     0
## lasso    0
## ridge    0
## lm       0
## pcr      0
## pls      0
## gam      0
## mars     0
##
## RMSE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## enet  0.1087351 0.1295883 0.1386462 0.1523779 0.1584816 0.2584927    0
## lasso 0.1098458 0.1282808 0.1385552 0.1528063 0.1582949 0.2621588    0
```

```
## ridge 0.1087385 0.1296083 0.1386274 0.1523776 0.1585024 0.2583984    0
## lm    0.1107092 0.1303882 0.1401511 0.1539649 0.1593398 0.2621257    0
## pcr   0.1100299 0.1304045 0.1398093 0.1535041 0.1592903 0.2619114    0
## pls   0.1103109 0.1300128 0.1399854 0.1537010 0.1589612 0.2623594    0
## gam   0.1040241 0.1253585 0.1365071 0.1487910 0.1507237 0.3582962    0
## mars  0.1081766 0.1273990 0.1349530 0.1392410 0.1463472 0.2166314    0
##
## Rsquared
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## enet  0.6034126 0.8409571 0.8648807 0.8380351 0.8837886 0.9132362    0
## lasso 0.6028958 0.8392119 0.8628049 0.8369765 0.8842992 0.9108037    0
## ridge 0.6033484 0.8410111 0.8648859 0.8380514 0.8838071 0.9132556    0
## lm    0.6063441 0.8314147 0.8593545 0.8345493 0.8795830 0.9092278    0
## pcr   0.6084139 0.8337164 0.8600473 0.8354930 0.8817183 0.9104176    0
## pls   0.6061271 0.8341593 0.8599035 0.8351255 0.8819106 0.9097822    0
## gam   0.4886485 0.8389810 0.8664405 0.8464234 0.8941007 0.9241065    0
## mars  0.7032250 0.8525139 0.8682535 0.8633930 0.8922254 0.9232260    0
```

```r
parallelplot(resamp, metric = "RMSE")
```




```r
bwplot(resamp, metric = "RMSE")
```



4 Final Model Exploration

```r
varImp(mars.fit)
```

```
## earth variable importance
##
##   only 20 most important variables shown (out of 48)
##
##                     Overall
## overall_qual        100.000
## total_sq             59.325
## year_built           45.155
## fireplaces           45.155
## year_remod_add       36.419
## bsmt_fin_sf1         31.983
## open_porch_sf        28.083
## lot_area             24.453
## total_bsmt_sf        14.410
## garage_finish        14.410
## full_bath            12.702
## exter_qual           11.584
## garage_area          10.452
## kitchen_qual          9.222
## heating_qc            7.746
## garage_typeDetchd     0.000
## yr_sold2009           0.000
## bedroom_abv_gr        0.000
## bsmt_fin_type1        0.000
## bsmt_qual             0.000
```