P8106 Data ScienceII Homework1

Yueran Zhang(yz4188)

2023-02-17

In this exercise, we predict the sale price of a house using its other characteristics. The training data are in "housing train.csv", and the test data are in "housing test.csv". The response is in the column "Sale price". Among the 25 feature variables, some are numeric features, such as living area square feet or first floor square feet, and some are categorical features, such as the overall material and finish of the house or kitchen quality. A detailed description of the variables is in "dictionary.txt".

Dataset Preparing

```
library(ISLR)
library(pls)
library(dplyr)
library(glmnet)
library(caret)
library(corrplot)
library(plotmo)
```

Import Datafile

```
set.seed(123)
training = read.csv("/Users/yueranzhang/Desktop/DSII/DSII/Dataset/housing_training.csv")
test = read.csv("//Users/yueranzhang/Desktop/DSII/DSII/Dataset/housing_test.csv")

# delete rows containing the missing data
training <- na.omit(training)
test <- na.omit(test)
```

```
training_x <- model.matrix(Sale_Price ~ ., training) [, -1]
training_y <- training$Sale_Price
test_x <- model.matrix(Sale_Price ~ ., test) [, -1]
test_y <- test$Sale_Price
```

Validiation Control

```r
ctrl1 <- trainControl(method = "repeatedcv", number = 10, repeats = 5)
```

Question1

Fit a linear model using least squares on the training data.

```r
set.seed(123)
Linear_Model <- lm(Sale_Price ~ . ,
                   data = training)
summary(Linear_Model)
```

```
##
## Call:
## lm(formula = Sale_Price ~ ., data = training)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -89864 -12424    416  12143 140205
##
## Coefficients: (1 not defined because of singularities)
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -4.985e+06  3.035e+06  -1.642  0.10076
## Gr_Liv_Area                 2.458e+01  1.393e+01   1.765  0.07778 .
## First_Flr_SF                4.252e+01  1.409e+01   3.017  0.00260 **
## Second_Flr_SF               4.177e+01  1.379e+01   3.029  0.00250 **
## Total_Bsmt_SF               3.519e+01  2.744e+00  12.827  < 2e-16 ***
## Low_Qual_Fin_SF                    NA         NA      NA       NA
## Wood_Deck_SF                1.202e+01  4.861e+00   2.474  0.01350 *
## Open_Porch_SF               1.618e+01  1.004e+01   1.611  0.10736
## Bsmt_Unf_SF                -2.087e+01  1.723e+00 -12.116  < 2e-16 ***
## Mas_Vnr_Area                1.046e+01  4.229e+00   2.473  0.01353 *
## Garage_Cars                 4.229e+03  1.893e+03   2.234  0.02563 *
## Garage_Area                 7.769e+00  6.497e+00   1.196  0.23195
## Year_Built                  3.251e+02  3.130e+01  10.388  < 2e-16 ***
## TotRms_AbvGrd              -3.838e+03  6.922e+02  -5.545 3.51e-08 ***
## Full_Bath                  -4.341e+03  1.655e+03  -2.622  0.00883 **
## Overall_QualAverage        -5.013e+03  1.735e+03  -2.890  0.00391 **
## Overall_QualBelow_Average  -1.280e+04  2.677e+03  -4.782 1.92e-06 ***
## Overall_QualExcellent       7.261e+04  5.381e+03  13.494  < 2e-16 ***
## Overall_QualFair           -1.115e+04  5.240e+03  -2.127  0.03356 *
## Overall_QualGood            1.226e+04  1.950e+03   6.287 4.30e-10 ***
## Overall_QualVery_Excellent  1.304e+05  8.803e+03  14.810  < 2e-16 ***
## Overall_QualVery_Good       3.798e+04  2.741e+03  13.852  < 2e-16 ***
## Kitchen_QualFair           -2.663e+04  6.325e+03  -4.210 2.71e-05 ***
## Kitchen_QualGood           -1.879e+04  4.100e+03  -4.582 5.01e-06 ***
## Kitchen_QualTypical        -2.677e+04  4.281e+03  -6.252 5.37e-10 ***
## Fireplaces                  1.138e+04  2.257e+03   5.043 5.18e-07 ***
## Fireplace_QuFair           -7.207e+03  6.823e+03  -1.056  0.29106
## Fireplace_QuGood            6.070e+02  5.833e+03   0.104  0.91713
```

```
## Fireplace_QuNo_Fireplace    3.394e+03  6.298e+03   0.539  0.59002
## Fireplace_QuPoor           -5.185e+03  7.399e+03  -0.701  0.48362
## Fireplace_QuTypical        -6.398e+03  5.897e+03  -1.085  0.27814
## Exter_QualFair             -3.854e+04  8.383e+03  -4.598 4.66e-06 ***
## Exter_QualGood             -1.994e+04  5.585e+03  -3.569  0.00037 ***
## Exter_QualTypical          -2.436e+04  5.874e+03  -4.147 3.57e-05 ***
## Lot_Frontage                1.024e+02  1.905e+01   5.376 8.90e-08 ***
## Lot_Area                    6.042e-01  7.864e-02   7.683 2.91e-14 ***
## Longitude                  -3.481e+04  2.537e+04  -1.372  0.17016
## Latitude                    5.874e+04  3.483e+04   1.686  0.09193 .
## Misc_Val                    9.171e-01  1.003e+00   0.914  0.36071
## Year_Sold                  -6.455e+02  4.606e+02  -1.401  0.16132
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22190 on 1401 degrees of freedom
## Multiple R-squared:  0.9116, Adjusted R-squared:  0.9092
## F-statistic: 380.3 on 38 and 1401 DF,  p-value: < 2.2e-16
```

Question2

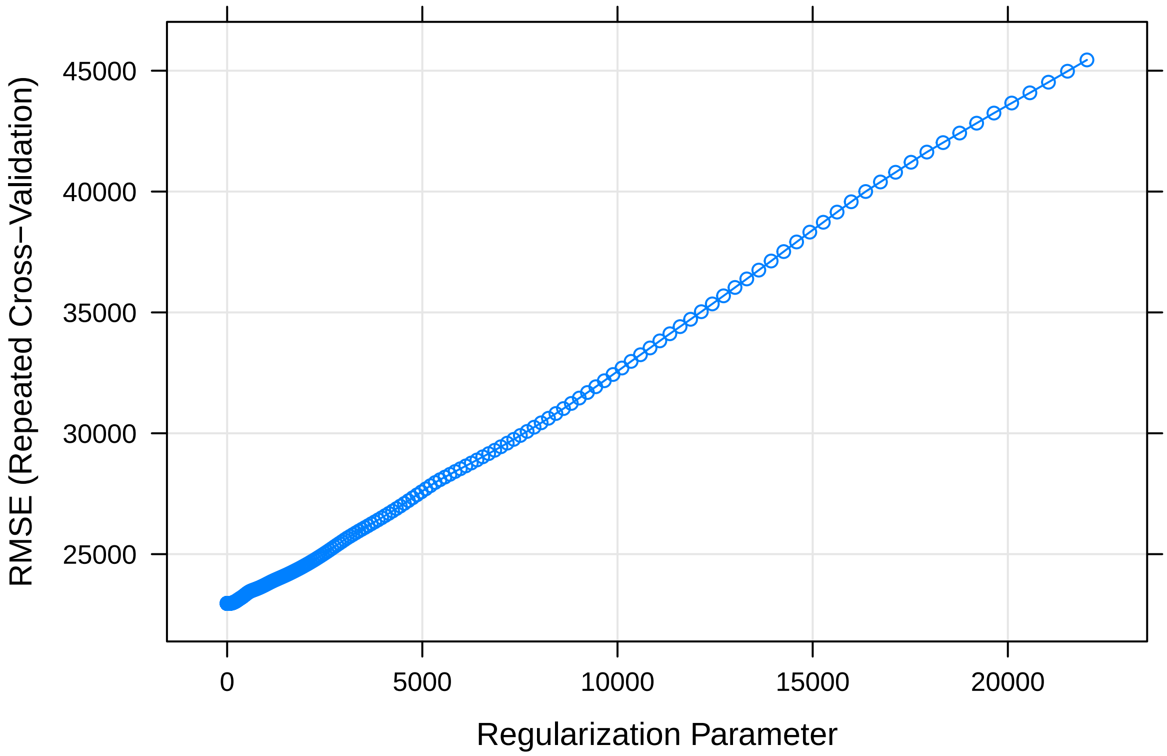
Fit a lasso model on the training data. Report the selected tuning parameter and the test error. When the 1SE rule is applied, how many predictors are included in the model?

Report the tuning parameter

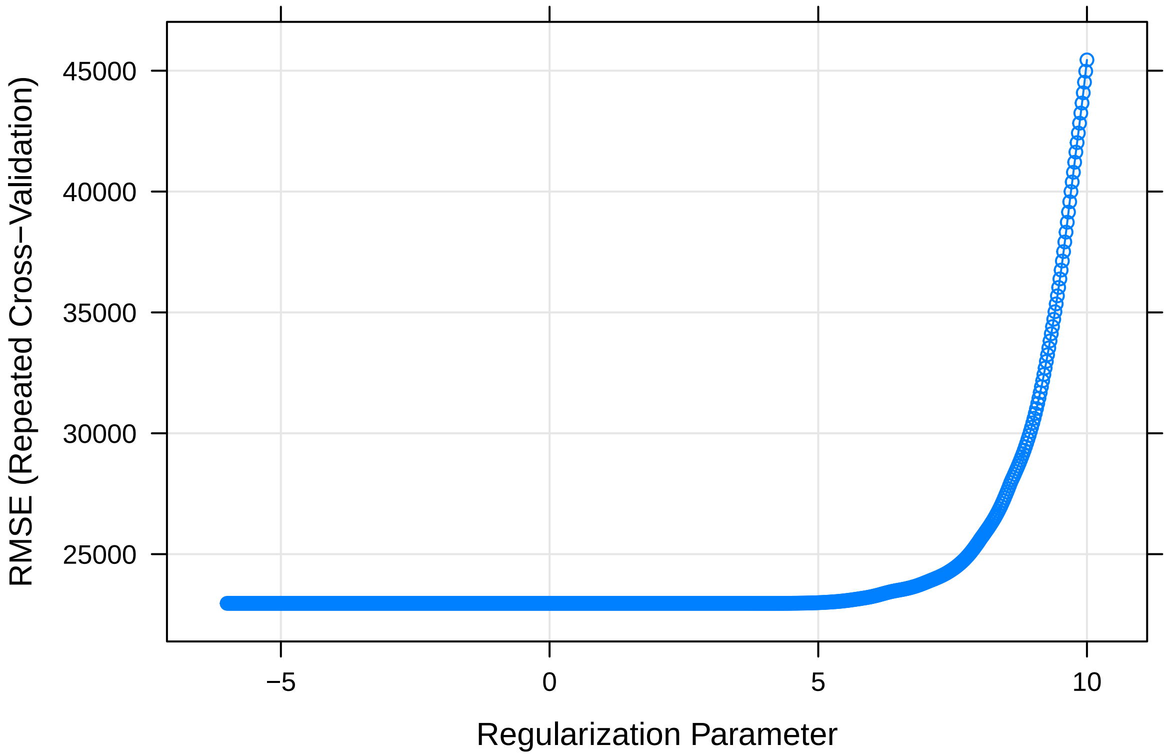
The tuning parameter lambda is chosen by cross-validation

```r
set.seed(123)
Lasso_fit <- train(training_x, training_y,
                       method = "glmnet",
                      tuneGrid = expand.grid(alpha = 1,
                                             lambda = exp(seq(10, -6,length = 700))),
                      trControl = ctrl1)
```

```r
plot(Lasso_fit)
```




```r
plot(Lasso_fit, xTrans = function(x) log(x))
```




```
bestlam_lasso = Lasso_fit$bestTune$lambda
bestlam_lasso
```

```
## [1] 58.64667
```

- The tuning parameter is `r bestlam_lasso`.

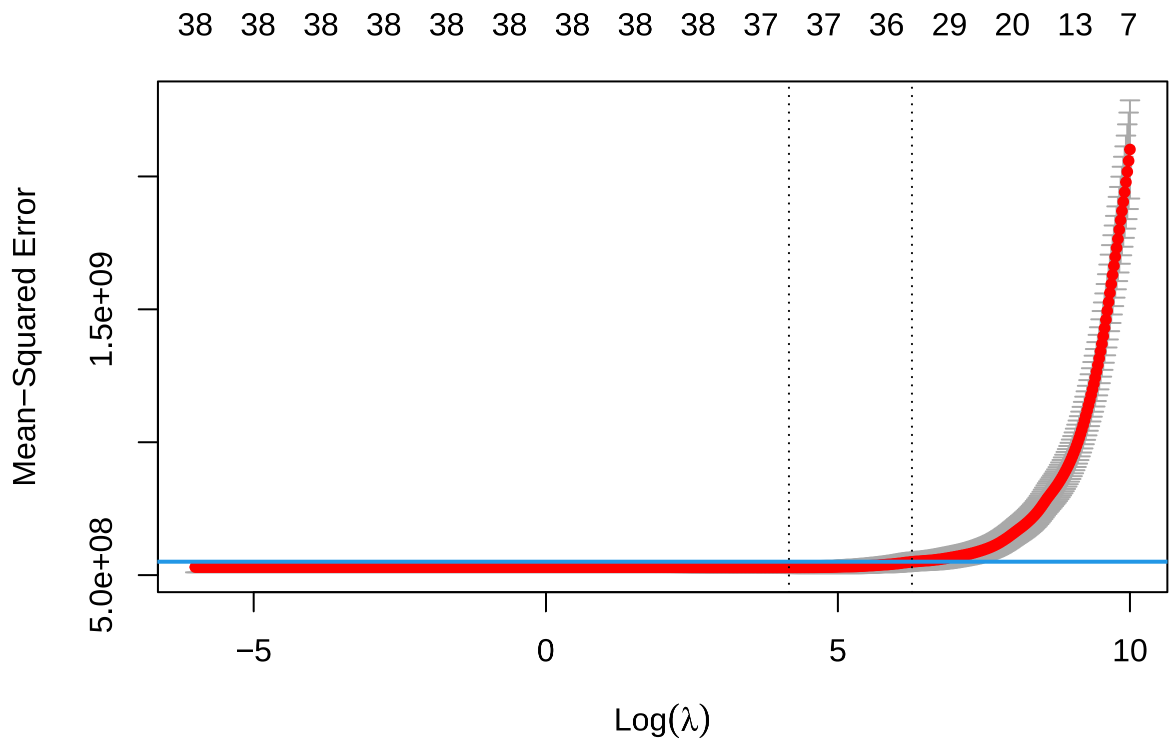
```
lasso_pred = predict(Lasso_fit$finalModel, s = bestlam_lasso, newx = test_x)
mean((lasso_pred - test_y)^2)
```

```
## [1] 440636589
```

- The mean test error is 4.4063659×10^8.

Numbers of the predictors

```
set.seed(123)
cv.ridge <- cv.glmnet(training_x, training_y,
                      alpha = 1,
                      lambda = exp(seq(10, -6, length = 700)))
plot(cv.ridge)
abline(h = (cv.ridge$cvm + cv.ridge$cvsd)[which.min(cv.ridge$cvm)], col = 4, lwd = 2)
```




```r
# the 1SE rule
set.seed(123)
lasso_1SE = cv.ridge$lambda.1se
lasso_1SE
```

```
## [1] 527.9258
```

```r
# number of predictors
## The number of coefficients - 1 (for the intercept) should work to give us the number of predictors.
lasso_coef = predict(cv.ridge, s = "lambda.1se", type = "coefficients")
lasso_coef
```

```
## 40 x 1 sparse Matrix of class "dgCMatrix"
##                                 lambda.1se
## (Intercept)                  -3.525624e+06
## Gr_Liv_Area                   5.960167e+01
## First_Flr_SF                  9.731630e-01
## Second_Flr_SF                 .
## Total_Bsmt_SF                 3.655861e+01
## Low_Qual_Fin_SF              -3.200503e+01
## Wood_Deck_SF                  9.369999e+00
## Open_Porch_SF                 1.071611e+01
## Bsmt_Unf_SF                  -2.042287e+01
## Mas_Vnr_Area                  1.352633e+01
## Garage_Cars                   3.312772e+03
```

```
## Garage_Area                  1.013961e+01
## Year_Built                   3.124432e+02
## TotRms_AbvGrd               -2.160372e+03
## Full_Bath                   -5.791354e+02
## Overall_QualAverage         -3.667714e+03
## Overall_QualBelow_Average   -1.028096e+04
## Overall_QualExcellent        9.014330e+04
## Overall_QualFair            -8.014265e+03
## Overall_QualGood             1.076594e+04
## Overall_QualVery_Excellent   1.597985e+05
## Overall_QualVery_Good        3.717742e+04
## Kitchen_QualFair            -9.103332e+03
## Kitchen_QualGood            -2.797853e+03
## Kitchen_QualTypical         -1.199281e+04
## Fireplaces                   7.347214e+03
## Fireplace_QuFair            -1.901878e+03
## Fireplace_QuGood             3.394685e+03
## Fireplace_QuNo_Fireplace     .
## Fireplace_QuPoor             .
## Fireplace_QuTypical         -2.533979e+03
## Exter_QualFair              -1.616220e+04
## Exter_QualGood               .
## Exter_QualTypical           -4.746311e+03
## Lot_Frontage                 8.178138e+01
## Lot_Area                     5.843030e-01
## Longitude                   -1.846434e+04
## Latitude                     3.150317e+04
## Misc_Val                     8.701813e-02
## Year_Sold                   -4.623821e+01
```

When applying the 1SE, the number of predictors is 35.

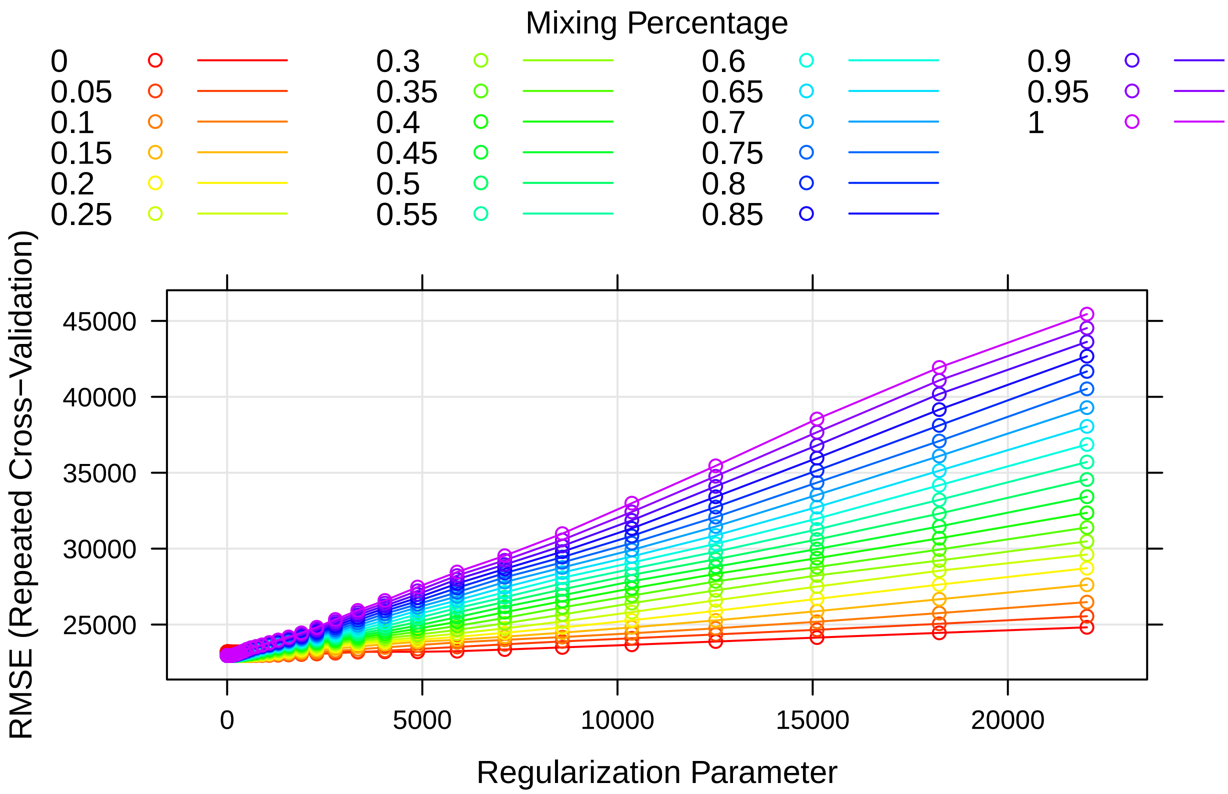
Question3

Fit an elastic net model on the training data. Report the selected tuning parameters and the test error. Is it possible to apply the 1SE rule to select the tuning parameters?

Report the selected tuning parameters and the test error

```
set.seed(123)
enet_fit <- train(training_x,training_y,
                  method = "glmnet",
                  tuneGrid = expand.grid(alpha = seq(0, 1, length = 21),
                                         lambda = exp(seq(10, -3, length = 70))),
                  trControl = ctrl1)
```

```
myCol <- rainbow(25)
myPar <- list(superpose.symbol = list(col = myCol),
                superpose.line = list(col = myCol))
plot(enet_fit, par.settings = myPar)
```




```
bestlam_enet = enet_fit$bestTune
bestlam_enet
```

```
##     alpha   lambda
## 121  0.05 614.1811
```

- The tuning parameter λ is 614.1811118. The tuning parameter α is 0.05.

```
pred_enet = predict(enet_fit$finalModel,s = enet_fit$bestTune$lambda,newx = test_x)
mean((pred_enet - test_y)^2)
```

```
## [1] 438262058
```

- The mean test error is 4.3826206×10^8.

Apply the 1SE rule to select the tuning parameters?

- In order to fit a linear regression model using the Elastic net model method on the training, we will set with lambda 1 and lambda 2 (or lambda and alpha) chosen by cross-validation. When we applying the 1SE, and try to tune alpha and lambda.1se for an elastic net,in the glmnet package, it is possible to tune lambda.1se, but it is not possible to tune alpha and lambda at the same time. Therefore, it is not suitable to apply the 1SE rule to select the tuning parameters.

Question 4

Fit a partial least squares model on the training data and report the test error. How many components are included in your model?

Report the test error

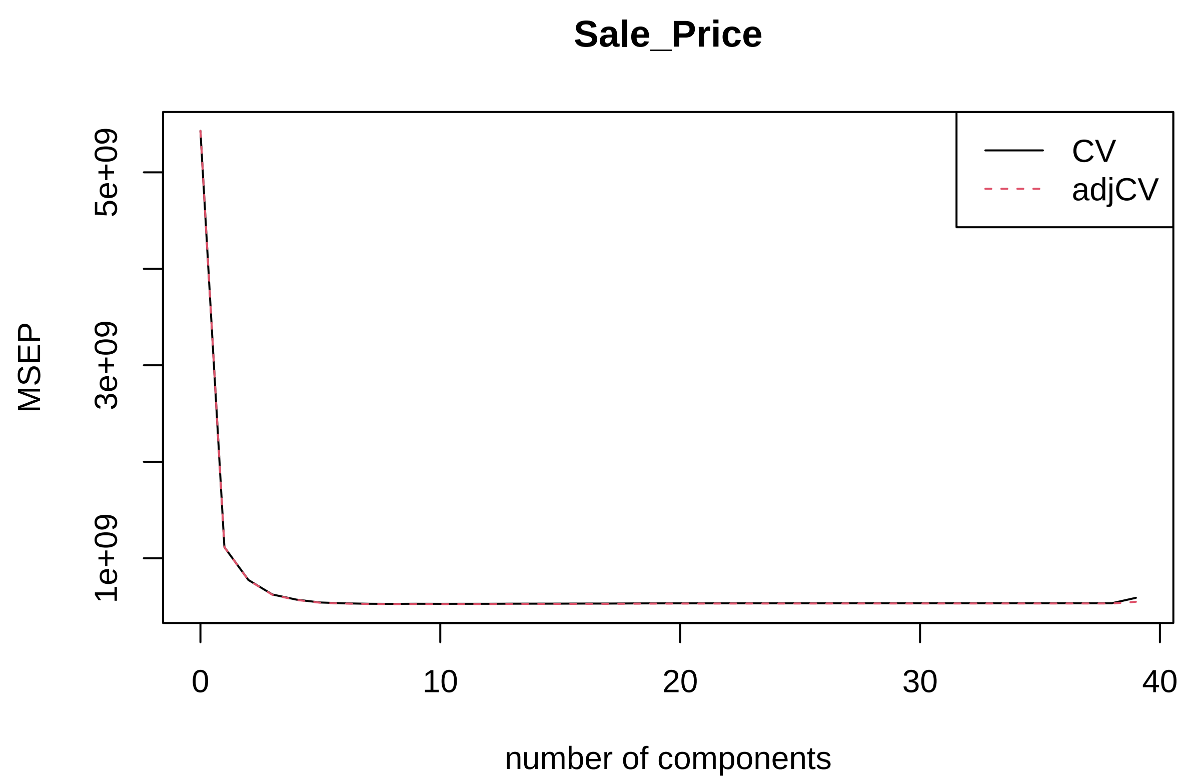
```
set.seed(123)
pls_mod <- plsr(Sale_Price~.,
                data = training,
                scale = TRUE,
                validation = "CV")

summary(pls_mod)
```

```
## Data:    X dimension: 1440 39
##  Y dimension: 1440 1
## Fit method: kernelpls
## Number of components considered: 39
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV           73685    33384    27836    24986    23902    23267    23082
## adjCV        73685    33379    27806    24924    23839    23207    23030
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       22979    22958    22980     22968     22975     22975     22995
## adjCV    22929    22909    22926     22914     22920     22920     22938
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV        23001     23008     23027     23025     23059     23078     23089
## adjCV     22943     22950     22967     22966     22997     23015     23024
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV        23095     23098     23100     23101     23104     23106     23109
## adjCV     23030     23033     23034     23036     23038     23040     23043
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV        23110     23110     23110     23110     23110     23110     23110
## adjCV     23044     23044     23044     23044     23044     23044     23044
##        35 comps  36 comps  37 comps  38 comps  39 comps
## CV        23110     23110     23110     23110     24287
## adjCV     23044     23044     23044     23044     23421
##
## TRAINING: % variance explained
##             1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X             20.02    25.93    29.67    33.59    37.01    40.03    42.49
## Sale_Price    79.73    86.35    89.36    90.37    90.87    90.99    91.06
##             8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X             45.53    47.97     50.15     52.01     53.69     55.35     56.86
## Sale_Price    91.08    91.10     91.13     91.15     91.15     91.16     91.16
##             15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## X              58.64     60.01     62.18     63.87     65.26     67.10
## Sale_Price     91.16     91.16     91.16     91.16     91.16     91.16
##             21 comps  22 comps  23 comps  24 comps  25 comps  26 comps
```

```
## X              68.44     70.12     71.72     73.35     75.20     77.27
## Sale_Price     91.16     91.16     91.16     91.16     91.16     91.16
##             27 comps  28 comps  29 comps  30 comps  31 comps  32 comps
## X              78.97     80.10     81.83     83.55     84.39     86.34
## Sale_Price     91.16     91.16     91.16     91.16     91.16     91.16
##             33 comps  34 comps  35 comps  36 comps  37 comps  38 comps
## X              88.63     90.79     92.79     95.45     97.49    100.00
## Sale_Price     91.16     91.16     91.16     91.16     91.16     91.16
##             39 comps
## X             100.67
## Sale_Price     91.16
```

```r
validationplot(pls_mod, val.type ="MSEP", legendpos = "topright")
```




```r
# training error
cv.mse <- RMSEP(pls_mod)
mean(min(cv.mse$val[1,,])^2)
```

```
## [1] 527076810
```

```r
# number of components
num_cv <- which.min(cv.mse$val[1,,])-1
num_cv
```

```
## 8 comps
##       8
```

- The model is with 8 components.

```r
# MSE
pls_pred <- predict(pls_mod, newdata = test_x,
                    ncomp = num_cv)
# test MSE
mean((test_y - pls_pred)^2)
```

```
## [1] 440217938
```

- The testing error is 4.4021794×10^8.

Question5

Which model will you choose for predicting the response? Why?

summary for models

```r
## Lasso
mean((lasso_pred - test_y)^2)
```

```
## [1] 440636589
```

```r
## Elastic Net Model
mean((pred_enet - test_y)^2)
```

```
## [1] 438262058
```

```r
## Partial Least Squares model
mean((test_y - pls_pred)^2)
```

```
## [1] 440217938
```

- As for response predicting model, we would choose the model with the lowest error on predicting the test set. From the above summary, the test error(MSE) for each model is that lasso regression model with 4.4063659×10^8, Elastic Net Model with 4.3826206×10^8, and Partial Least Squares model with 4.4021794×10^8. By comparing the MSE value for each model, we see that Partial Least Squares model has the lowest test error, Partial Least Squares model regression model would be the best choice for predicting the response.