LASSO and Ridge Regression amsterdam

Laurens van der Maas

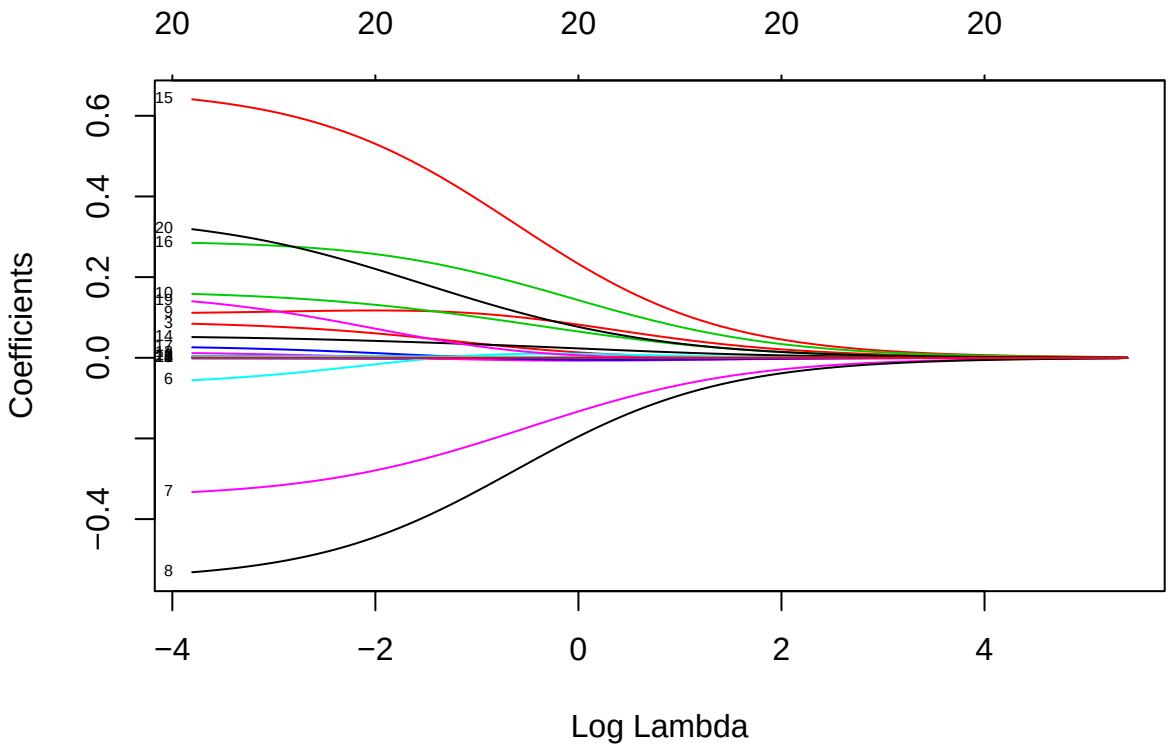
12/1/2019

Ridge Regression

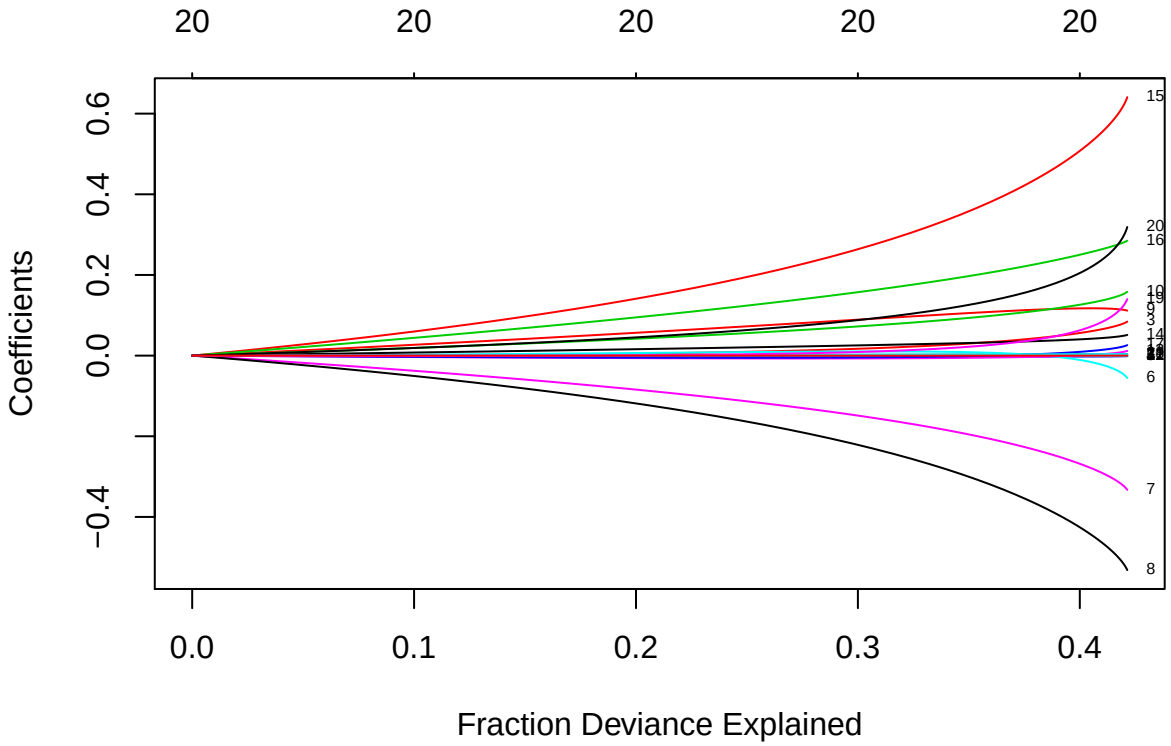
```
suppressMessages(library(readr))
suppressMessages(library(glmnet))

amsterdam <- read_csv('st445_final_data')
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   review_scores_rating = col_double(),
##   host_is_superhost = col_double(),
##   host_listings_count = col_double(),
##   host_identity_verified = col_double(),
##   room_type = col_character(),
##   bathrooms = col_double(),
##   bedrooms = col_double(),
##   minimum_nights = col_double(),
##   number_of_reviews = col_double(),
##   cancellation_policy = col_character(),
##   instant_bookable = col_logical(),
##   cleaning_fee = col_double(),
##   location_3ways = col_character(),
##   realprice = col_double(),
##   host_since_duration = col_double(),
##   logprice = col_double()
## )
```

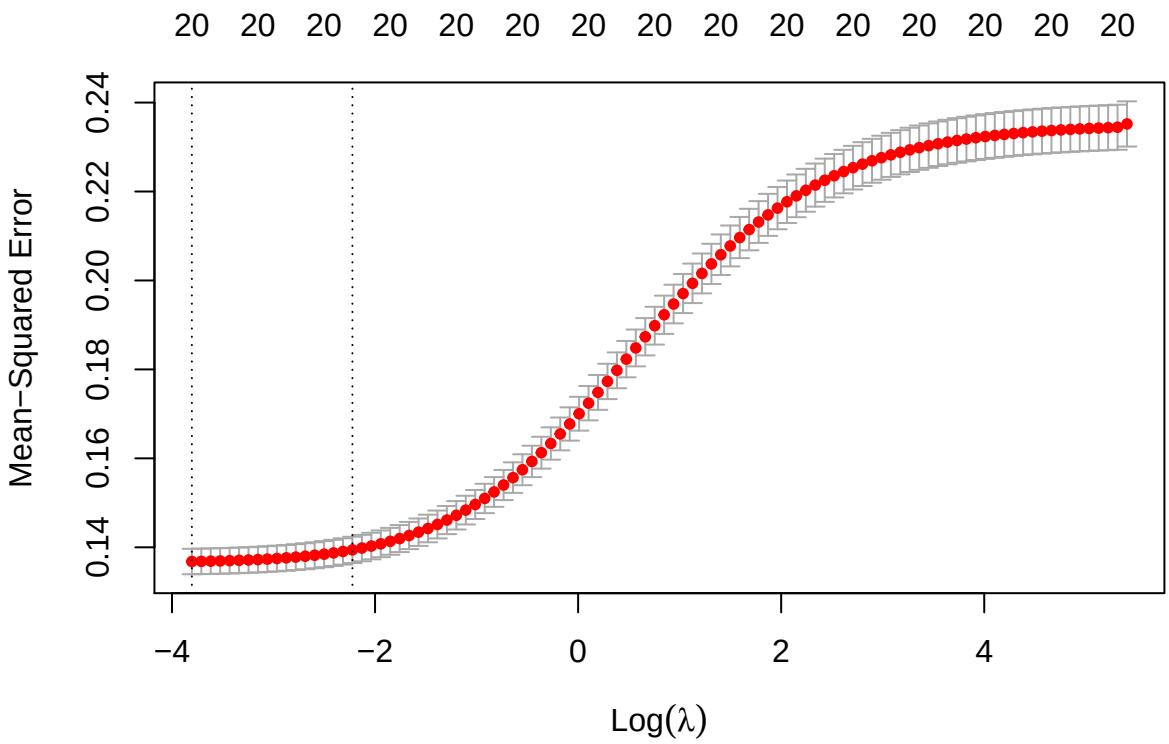
```
amsterdam <- amsterdam[,-c(1,15)]

# glmnet does not use formula language
x <- model.matrix(logprice ~ ., data = amsterdam)
y <- amsterdam$logprice

fit.ridge <-glmnet(x, y, alpha=0)

# 8, 7, 15, 20, 16 most important var
plot(fit.ridge, xvar="lambda", label= TRUE)
```




```
plot(fit.ridge, xvar="dev", label= TRUE)
```




```
cv.ridge <-cv.glmnet(x, y, alpha=0)
## Plot of CV mse vs log (lambda), small lambda is best apparently.
plot(cv.ridge)
```




```
## Coefficent vector corresponding to the mse which is within one standard error of the lowest mse usin
coef(cv.ridge)
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                                       1
## (Intercept)                                4.0718649364
## (Intercept)                                .
## review_scores_rating                       0.0032651038
## host_is_superhost                          0.0652320883
## host_listings_count                       -0.0003663436
## host_identity_verified                    -0.0021231936
## room_typeHotel room                       -0.0222974072
## room_typePrivate room                     -0.2901882819
## room_typeShared room                      -0.4627803907
## bathrooms                                  0.1170837789
## bedrooms                                   0.1364474682
## minimum_nights                            -0.0001937531
## number_of_reviews                         -0.0003570440
## cancellation_policymoderate                0.0035154889
## cancellation_policystrict_14_with_grace_period  0.0434824030
## cancellation_policysuper_strict_30         0.5529793443
## cancellation_policysuper_strict_60         0.2635415827
## instant_bookableTRUE                       0.0141006790
## cleaning_fee                               0.0035043919
## location_3waysModerate                     0.0828561550
## location_3waysnear_centre                  0.2367923240
## host_since_duration                       -0.0000104894
```

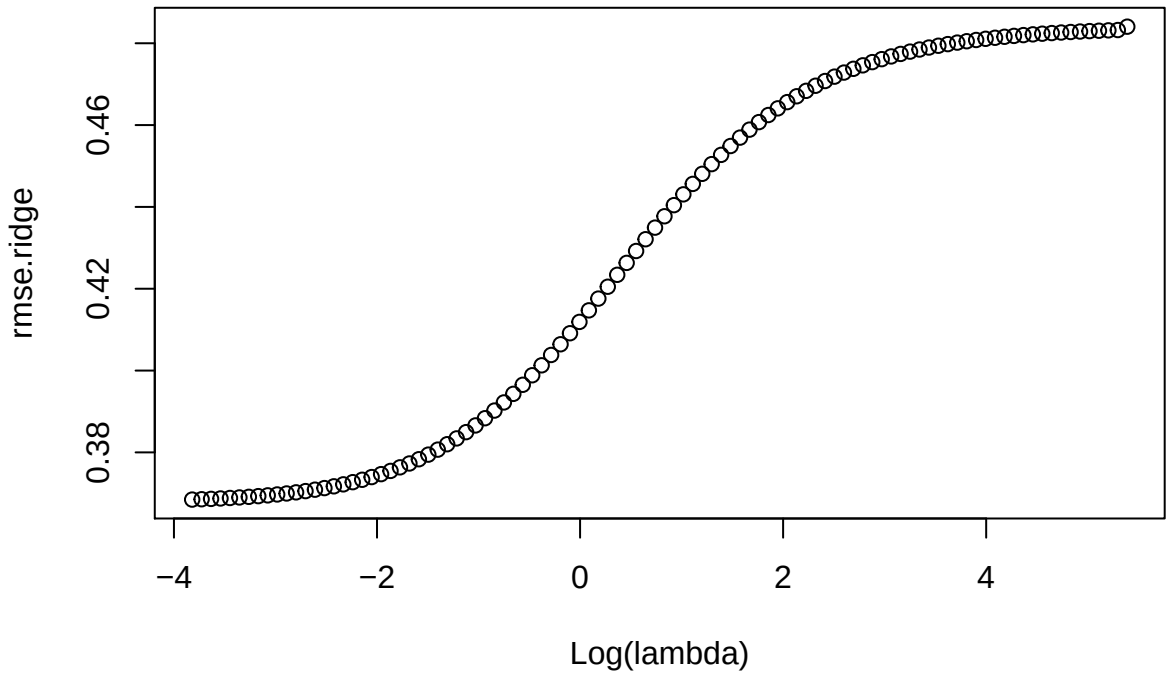
```
## Coefficient vector corresponding to the lowest mse using the best lambda
coef(glmnet(x,y,alpha=0, lambda=cv.ridge$lambda.min))
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                                       s0
## (Intercept)                                  3.967646e+00
## (Intercept)                                  .
## review_scores_rating                         3.592352e-03
## host_is_superhost                            8.430644e-02
## host_listings_count                         -5.073438e-04
## host_identity_verified                      -1.472142e-03
## room_typeHotel room                         -5.567573e-02
## room_typePrivate room                       -3.328267e-01
## room_typeShared room                        -5.317271e-01
## bathrooms                                    1.115002e-01
## bedrooms                                     1.582633e-01
## minimum_nights                              -2.600664e-04
## number_of_reviews                           -3.480520e-04
## cancellation_policymoderate                  1.165951e-02
## cancellation_policystrict_14_with_grace_period  5.136956e-02
## cancellation_policysuper_strict_30           6.407773e-01
## cancellation_policysuper_strict_60           2.849320e-01
## instant_bookableTRUE                         2.598437e-02
## cleaning_fee                                 3.518221e-03
## location_3waysModerate                       1.400964e-01
## location_3waysnear_centre                    3.188280e-01
## host_since_duration                         -1.652107e-05
```

```
# finding MSE
set.seed(1)
train <-sample(seq(15018), 7509, replace=FALSE)
ridge.train <-glmnet(x[train,], y[train], alpha = 0)
pred.test.ridge <-predict(ridge.train, x[-train,])
dim(pred.test.ridge)
```

```
## [1] 7509  100
```

```
rmse.ridge <-sqrt(apply((y[-train]-pred.test.ridge)^2,2,mean))
plot(log(ridge.train$lambda), rmse.ridge, type="b", xlab="Log(lambda)")
```




```
lambda.best.ridge <- ridge.train$lambda[order(rmse.ridge)[1]]
lambda.best.ridge
```

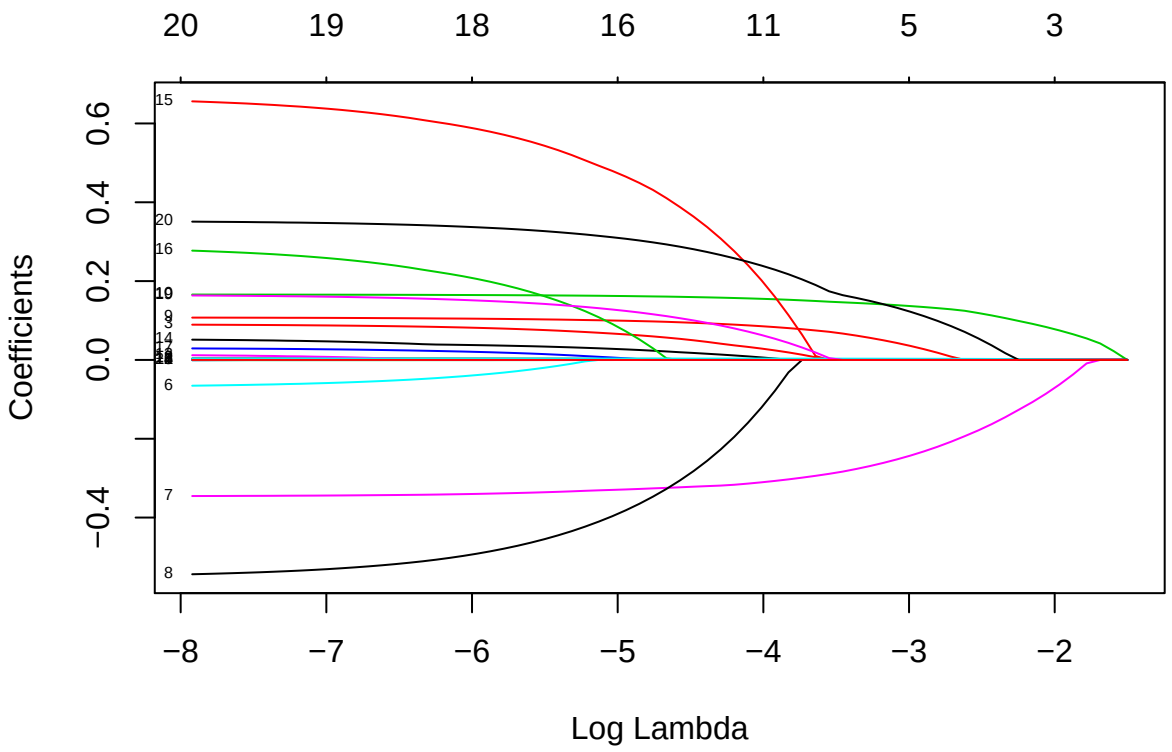
```
## [1] 0.0219165
```

```
mseRidge <- min(rmse.ridge)
mseRidge
```

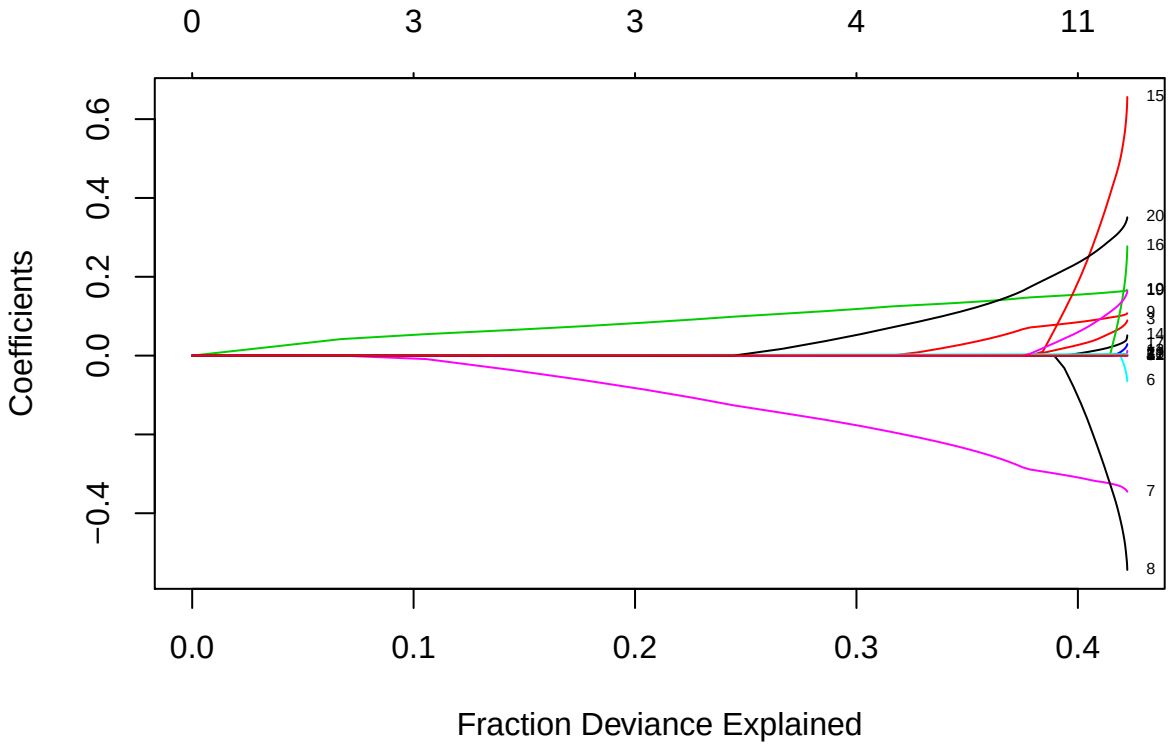
```
## [1] 0.3684685
```

LASSO Regression

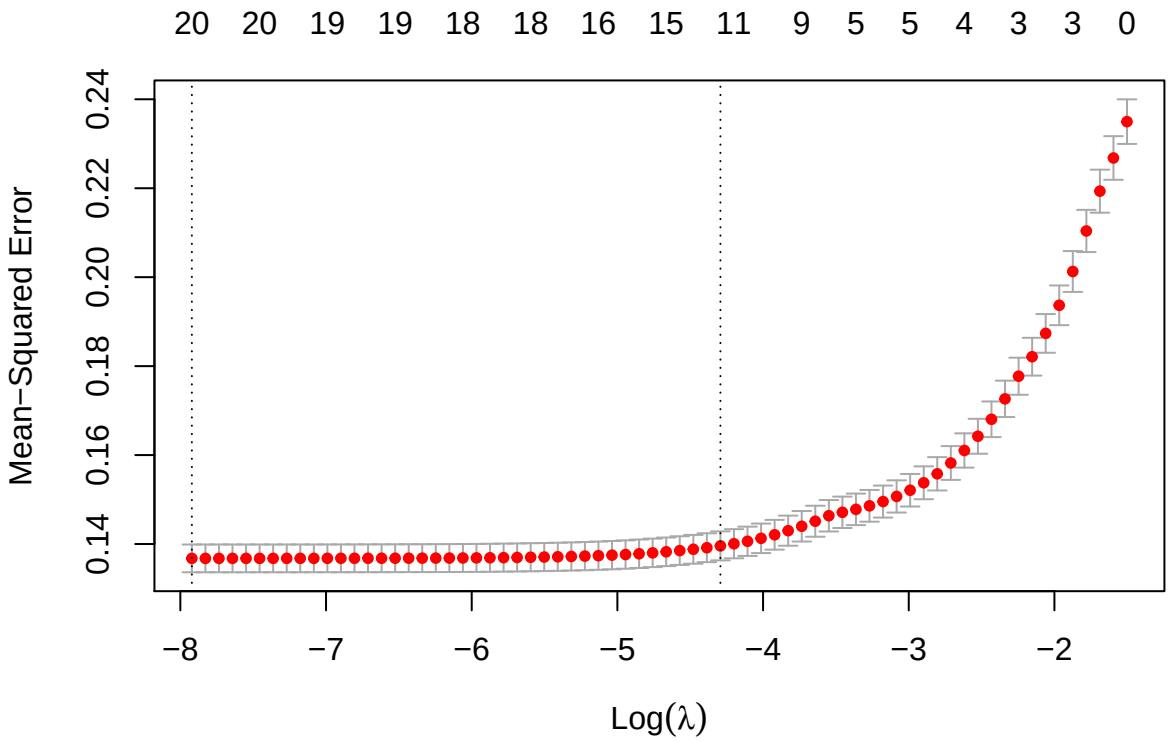
```
fit.lasso <- glmnet(x,y)
plot(fit.lasso, xvar="lambda", label= TRUE)
```




```
plot(fit.lasso, xvar="dev", label= TRUE)
```




```
cv.lasso <-cv.glmnet(x, y)
# Again, 8, 15, 7, 20.

plot(cv.lasso)
```




```
# Use very small lambda, again
## coefficent vector corresponding to the mse which is within one standard error of the lowest mse usin
coef(cv.lasso)
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                                          1
## (Intercept)                                   4.188284e+00
## (Intercept)                                   .
## review_scores_rating                          1.923599e-03
## host_is_superhost                             4.118934e-02
## host_listings_count                           .
## host_identity_verified                        .
## room_typeHotel room                           .
## room_typePrivate room                        -3.188685e-01
## room_typeShared room                         -2.271210e-01
## bathrooms                                     9.125891e-02
## bedrooms                                      1.579574e-01
## minimum_nights                                .
## number_of_reviews                            -1.820841e-05
## cancellation_policymoderate                   .
## cancellation_policystrict_14_with_grace_period 1.355910e-02
## cancellation_policysuper_strict_30            3.078769e-01
## cancellation_policysuper_strict_60            .
## instant_bookableTRUE                          .
## cleaning_fee                                  3.405361e-03
## location_3waysModerate                        8.732526e-02
## location_3waysnear_centre                     2.661972e-01
## host_since_duration                           .
```

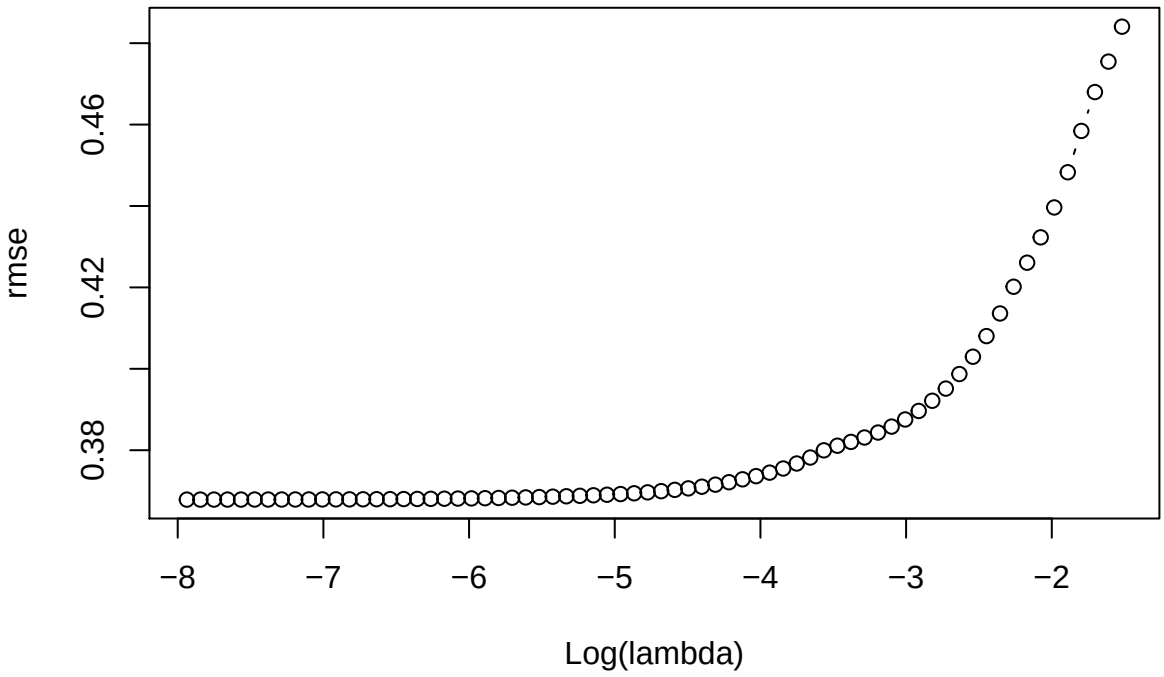
```
## coefficient vector corresponding to the lowest mse using the best lambda
coef(glmnet(x,y, lambda=cv.lasso$lambda.min))
```

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                                                       s0
## (Intercept)                                  3.940950e+00
## (Intercept)                                  .
## review_scores_rating                         3.632109e-03
## host_is_superhost                            8.941803e-02
## host_listings_count                         -5.363794e-04
## host_identity_verified                      -5.333925e-04
## room_typeHotel room                         -6.555163e-02
## room_typePrivate room                       -3.453835e-01
## room_typeShared room                        -5.437931e-01
## bathrooms                                    1.074298e-01
## bedrooms                                     1.658676e-01
## minimum_nights                              -2.592541e-04
## number_of_reviews                           -3.362666e-04
## cancellation_policymoderate                  1.237059e-02
## cancellation_policystrict_14_with_grace_period  5.173575e-02
## cancellation_policysuper_strict_30           6.563447e-01
## cancellation_policysuper_strict_60           2.775788e-01
## instant_bookableTRUE                         2.921495e-02
## cleaning_fee                                 3.470366e-03
## location_3waysModerate                       1.636307e-01
## location_3waysnear_centre                    3.509282e-01
## host_since_duration                         -1.841013e-05
```

```
## Validation set approach to select best lambda in Lasso
set.seed(1)
train <-sample(seq(15018), 7509, replace=FALSE)
lasso.train <-glmnet(x[train,], y[train])
pred.test <-predict(lasso.train, x[-train,])
dim(pred.test)
```

```
## [1] 7509   70
```

```
rmse <-sqrt(apply((y[-train]-pred.test)^2,2,mean))
plot(log(lasso.train$lambda), rmse, type="b", xlab="Log(lambda)")
```




```
lambda.best <-lasso.train$lambda[order(rmse)[1]]
lambda.best
```

```
## [1] 0.0003571843
```

```
mseLasso <- min(rmse)
mseLasso
```

```
## [1] 0.3678683
```