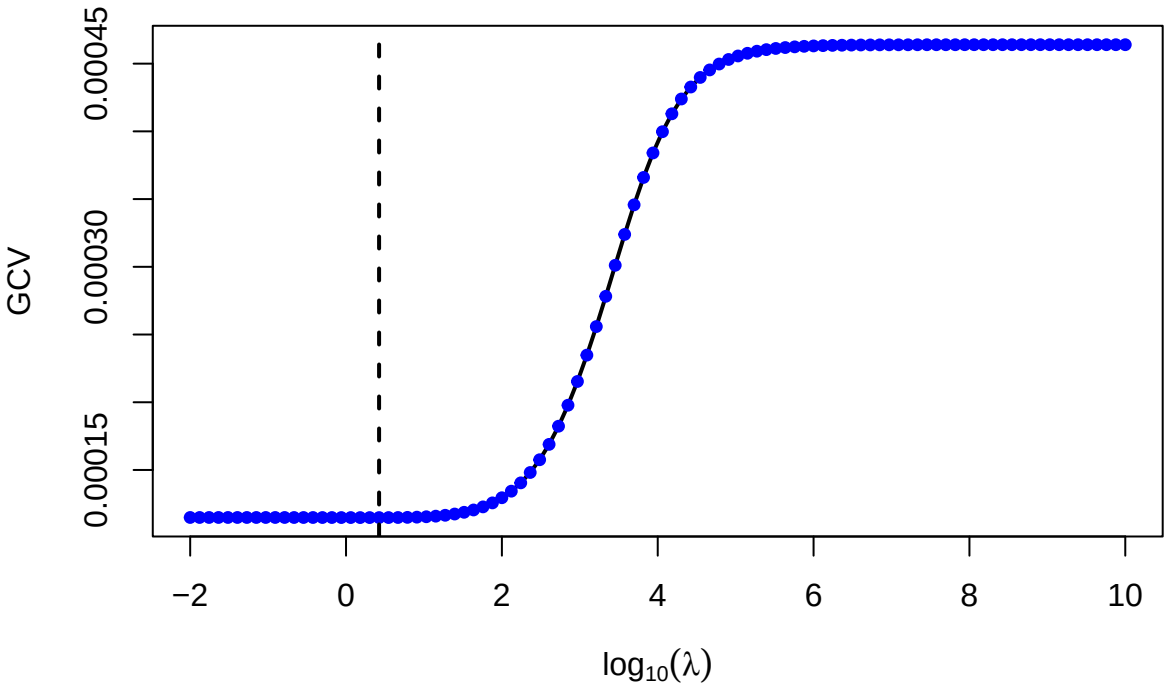
STA 5207: Homework 10

Include your R code as an appendix at the end of your homework. Do not include your code in your answers unless the question explicitly tells you to include your code. Your answers to each exercise should be self-contained without code so that the grader can determine your solution without reading your code or deciphering its output.

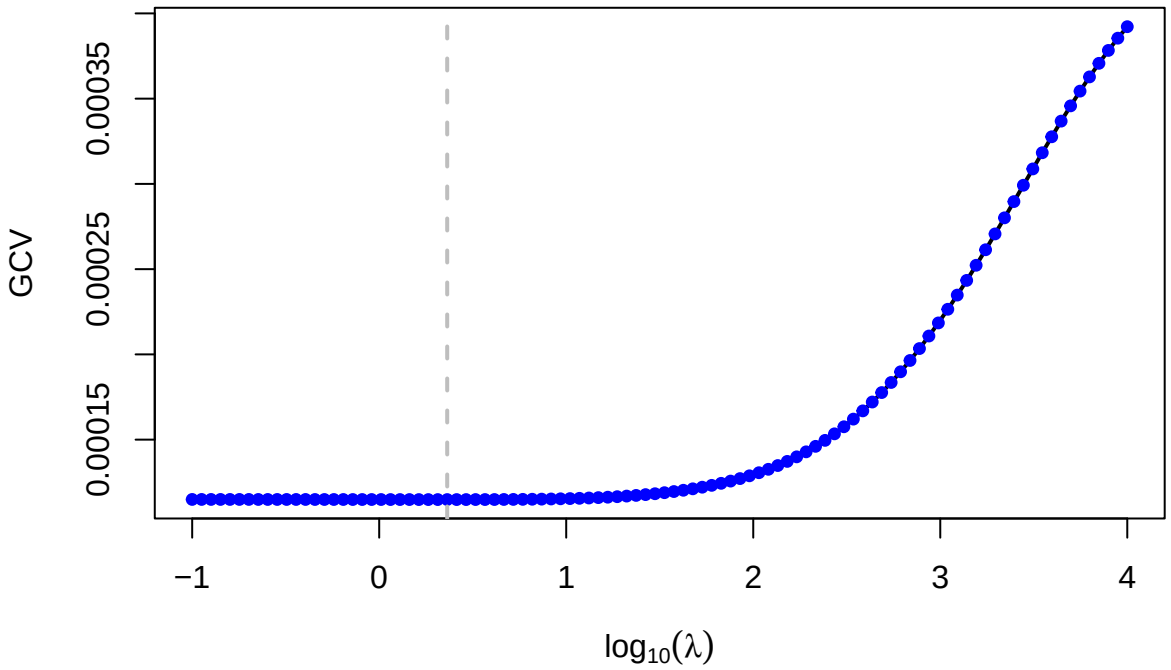
Exercise 1 (Boston Housing) [100 points]

For this exercise, we will analyze a data set containing housing values in 506 suburbs of Boston. The data set was split into a training and testing data set. Note that this data set is a version of the `Boston` data set from the `ISLR2` package, so you can type `?ISLR2::Boston` in `R` to read about the data set and the meaning of the variables. The training data set contains 354 suburbs and 13 variables. In the following exercises, use `log(medv)` (the logarithm of the median value of owner-occupied homes in $1000s) as the response and the other variables as predictors. You should use the `boston_train.csv` data set unless otherwise specified.

1. (10 points) Perform ridge regression with `log(medv)` as the response and the other variables as predictors using the data in `boston_train.csv`. Choose an appropriate value of λ using GCV. Justify the range of λ values you searched over and report your final choice of λ. Include any necessary plots in your response.



–The plot above displays a large grid of 100 lambda (λ) values evenly spaced on a log scale. Our selection criterion is based on minimizing the Generalized Cross-Validation (GCV) error, where λ is determined as 2.6560878. To ensure the robustness of our parameter search, we will investigate the impact of altering the range of λ values.

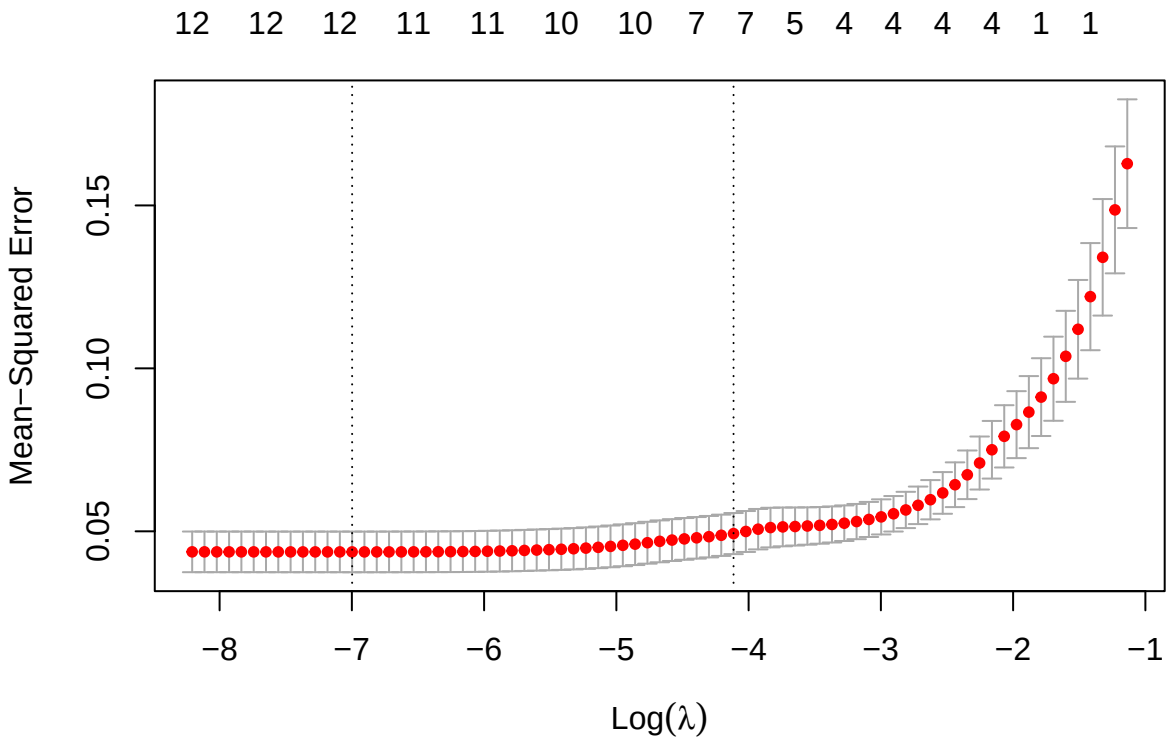


–The plot effectively illustrates the optimal point where the Generalized Cross-Validation (GCV) error is minimized. Therefore, the ultimate selection for the value of lambda is given by 2.31.

2. (6 points) Report the estimated regression equation and R^2 value for the ridge regression model you chose in Question 1.

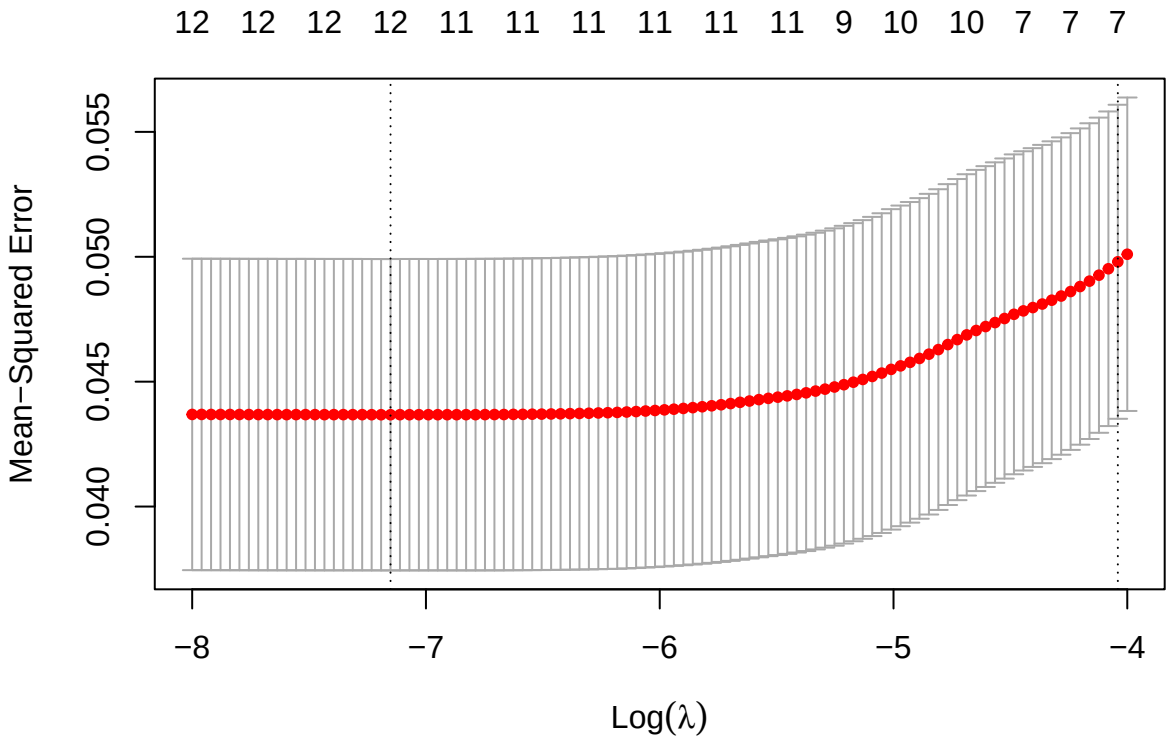
- log(medv) = 4.46313 - 0.01133$crim_i$ + 0.00095zn_i + 0.00061$indus_i$ + 0.08128$chas_i$ - 0.85419nox_i + 0.08171rm_i - 0.00036age_i - 0.05763dis_i + 0.01544rad_i - 0.00061tax_i - 0.04008$ptratio_i$ - 0.02854$lstat_i$
- The R^2 value is 0.75940

3. (10 points) Perform lasso with `log(medv)` as the response and the other variables as predictors using the data in `boston_train.csv`. You should set a random seed of 42. Justify the range of λ values you searched over and report your chosen values of `lambda.min` and `lambda.1se`. Include any necessary plots in your response.



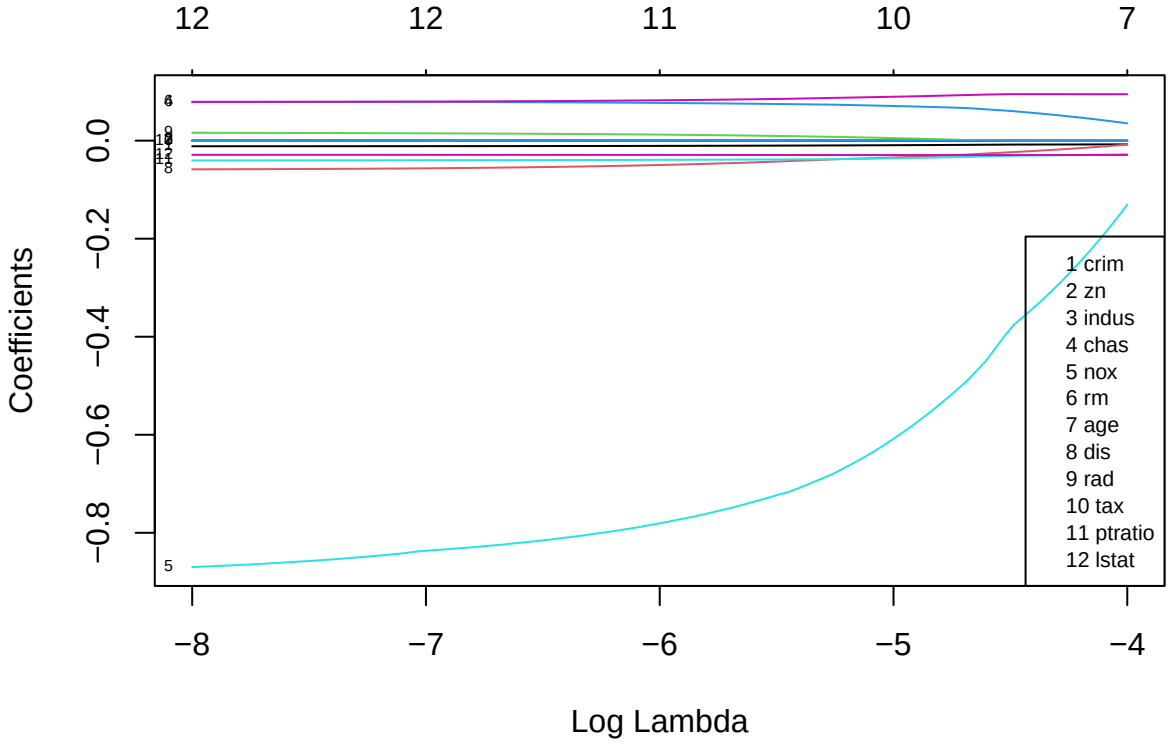
— To confirm the appropriateness of the range, let's ensure that the values for lambda.min (0.000913734) and lambda.1se (0.0163435) fall within the interior of the plot. After verification, we can affirm that the range is indeed suitable.

– For a more detailed examination, let's define a custom range for λ spanning from -8 to -4



– This plot looks good,So the value for lambda.min is 0.0007836758 and the value for lambda.1se is 0.01759036.

4. (4 points) Report the plot of the solution path for the lasso coefficient estimates.



5. (6 points) Report the number of variables with non-zero coefficients and the estimated regression equation for the lasso model estimated using `lambda.min`.

- log(medv) = 4.46712 - 0.01126$crim_i$ + 0.000874zn_i + 0.000122$indus_i$ + 0.079112$chas_i$ - 0.84438nox_i + 0.07960rm_i - 0.0002744age_i - 0.056944dis_i + 0.015191rad_i - 0.000586tax_i - 0.04005$ptratio_i$ - 0.02889$lstat_i$
- 12 predictors with non-zero coefficients.

6. (6 points) Report the number of variables with non-zero coefficients and the estimated regression equation for the lasso model estimated using `lambda.1se`.

- log(medv) = 3.4809 - 0.00758$crim_i$ + 0.03787$chas_i$ -0.15573nox_i + 0.09471rm_i -0.00958dis_i - 0.02809$ptratio_i$ - 0.02917$lstat_i$
- 7 predictors with non-zero coefficients.

7. (8 points) The file `boston_test.csv` on Canvas contains a new test data set of 152 houses not found in `boston_train.csv`. Calculate the RMSE values on this test data (`boston_test.csv`) for the following four models:

   - **Model 1**: The ridge regression model you chose in Question 1,
   - **Model 2**: The lasso model using `lambda.min` you reported in Question 5.
   - **Model 3**: The lasso model using `lambda.1se` you reported in Question 6.
   - **Model 4**: An OLS regression model estimated with `log(medv)` as the response and the other variables as predictors.

   The RMSE should be calculated using the logarithm of the response, that is,

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}[\log(\texttt{medv}_i) - \hat{y}_i]^2}.$$

   Based on these test RMSE values, which model do you prefer?

Model	RMSE
Ridge	0.1833554
lasso with lambda.min	0.1836097
lasso with lambda.1se	0.1912685
OLS	0.1845913

- Ridge Model has low RMSE , So it can be choosen.

Code Appendix

```
## Solution for Ex 1 Q 1


library(lmridge)

grid = 10 ^ seq(10, -2 , length = 100)
r_mod = lmridge(log(medv) ~ ., data = boston_train, scaling = 'scaled', K = grid)

# Fetch the GCV errors and lambda

k_est = kest(r_mod)

# Plot GCV vs lambda

plot(log10(r_mod$K), k_est$GCV, type = 'l', lwd = 2,
     xlab = expression(log[10](lambda)), ylab = 'GCV')

points(log10(r_mod$K), k_est$GCV,
       pch = 19, col = 'blue', cex = 0.75)

abline(v=log10(k_est$kGCV), lty = 'dashed', col = 'black',
       lwd = 2)


## Solution for Ex 1 Q 1

library(lmridge)

grid = 10 ^ seq(-1, 4 , length = 100)
r_mod=
  lmridge(log(medv) ~ .,data = boston_train,scaling = 'scaled',K = grid)

# Fetch the GCV errors and lambda

k_est = kest(r_mod)

# Plot GCV vs lambda

plot(log10(r_mod$K), k_est$GCV, type = 'l', lwd = 2,
     xlab = expression(log[10](lambda)), ylab = 'GCV')

points(log10(r_mod$K), k_est$GCV,
       pch = 19, col = 'blue', cex = 0.75)
```

```r
abline(v=log10(k_est$kGCV), lty = 'dashed', col = 'gray',
       lwd = 2)

#k_est$kGCV


## Solution for Ex 1 Q 2

# best lambda value
k_best = kest(r_mod)$kGCV

# re-fit the model

ridge_mod_best =
  lmridge(log(medv) ~ ., data = boston_train, scaling = 'scaled', K = k_best)


coef(ridge_mod_best)



## Solution for Ex 1 Q 3
library(glmnet)

x_train = model.matrix(log(medv) ~ ., data = boston_train)[, -1]
y_train = log(boston_train$medv)



set.seed(42)

l_model = cv.glmnet(x_train, y_train)

plot(l_model)



## Solution for Ex 1 Q 3

set.seed(42)

grid = exp(seq(-8, -4, length=100))


l_model = cv.glmnet(x_train, y_train, lambda = grid)

plot(l_model)

#l_model$lambda.min

#l_model$lambda.1se

## Solution for Ex 1 Q 4
```

```r
# solution path plot
plot(l_model$glmnet.fit, xvar = 'lambda', label = TRUE)

# include a legend of the variable names
pred_names = colnames(x_train)
legend('bottomright',
       legend = paste(1:length(pred_names), pred_names),
       cex= 0.7)



## Solution for Ex 1 Q 5

coef(l_model, s = 'lambda.min')


## Solution for Ex 1 Q 6

coef(l_model, s = 'lambda.1se')


## Solution for Ex 1 Q 7

boston_test=
  read.csv("~/Documents/boston_test.csv")

# Function to calculate RMSE
rmse = function(y_true, y_pred) {
    sqrt(mean((y_true - y_pred)^2))
}

x_test = model.matrix(log(medv) ~ ., data = boston_test)[, -1]
y_test = log(boston_test$medv)

# 1 Model

y_pred = predict(ridge_mod_best, newdata = boston_test)
rmse(y_test, y_pred)

# Model 2

y_pred = predict(l_model, newx = x_test, s = 'lambda.min')

# calculate the RMSE
rmse(y_test, y_pred)


# Model 3

y_pred = predict(l_model, newx = x_test, s = 'lambda.1se')

# calculate the RMSE
rmse(y_test, y_pred)
```

```r
# Model 4


ols_mod = lm(log(medv) ~ ., data = boston_train)


y_pred = predict(ols_mod, boston_test)

# calculate the RMSE
rmse(y_test, y_pred)
```