P8106_stl2137_HW1

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
## Parsed with column specification:
## cols(
##   .default = col_double()
## )
```

```
## See spec(...) for full column specifications.
```

```
### Creating variables & training control for Linear Model

# matrix of predictors (training)
## [,-1] due to intercept variable
x <- model.matrix(Solubility ~ ., training_dat)[,-1]

# vector of response (training)
y <- training_dat$Solubility

# creating training controls
control1 <- trainControl(method = "repeatedcv", number = 10, repeats = 5)

# matrix of predictors (test)
x.test <- model.matrix(Solubility ~ ., test_dat)[,-1]

# vector of response (test)
y.test <- test_dat$Solubility
```

Part A

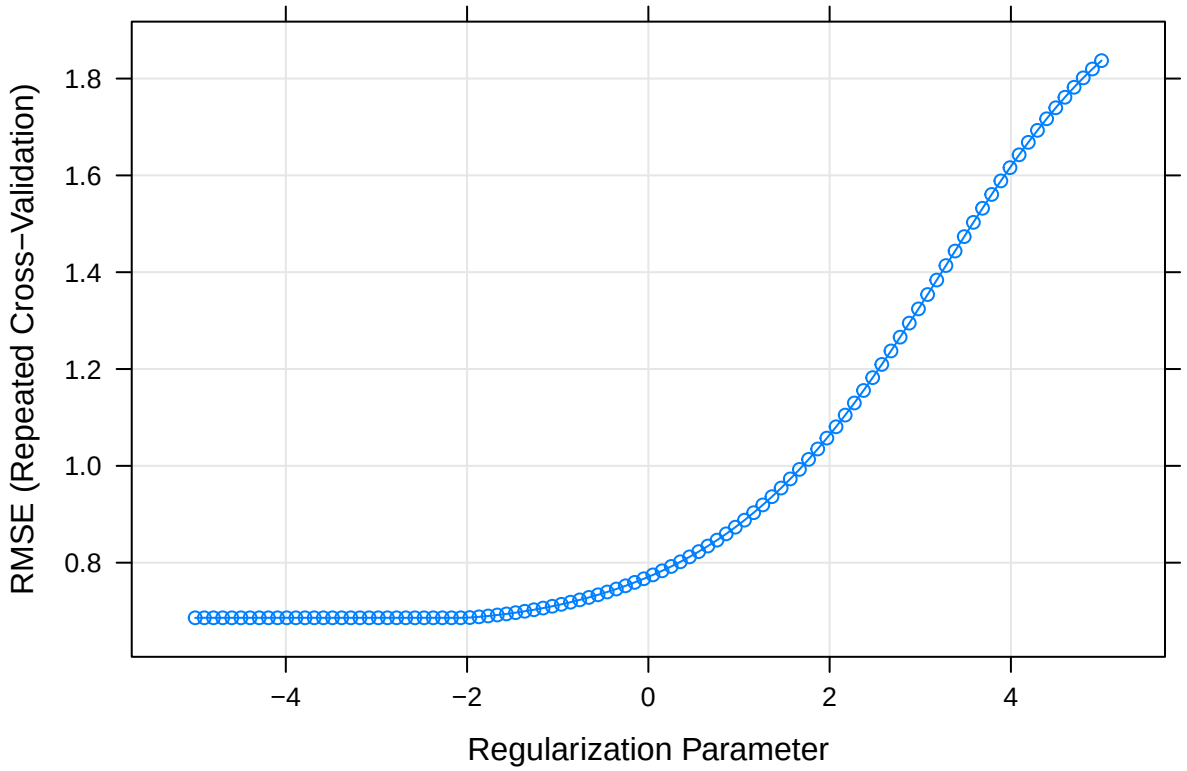
```
lm.fit <- train(x, y,
                method = "lm",
                trControl = control1)

predict.lm.fit <- predict(lm.fit, newdata = test_dat)
linear_mse <- mse(y.test, predict.lm.fit)
```

The MSE of the linear model on the test data is 0.5558898.

Part B

```
ridge.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 0,
                                          lambda = exp(seq(-5, 5, length = 100))),
                   trControl = control1)
```

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




```r
ridge.fit$bestTune
```

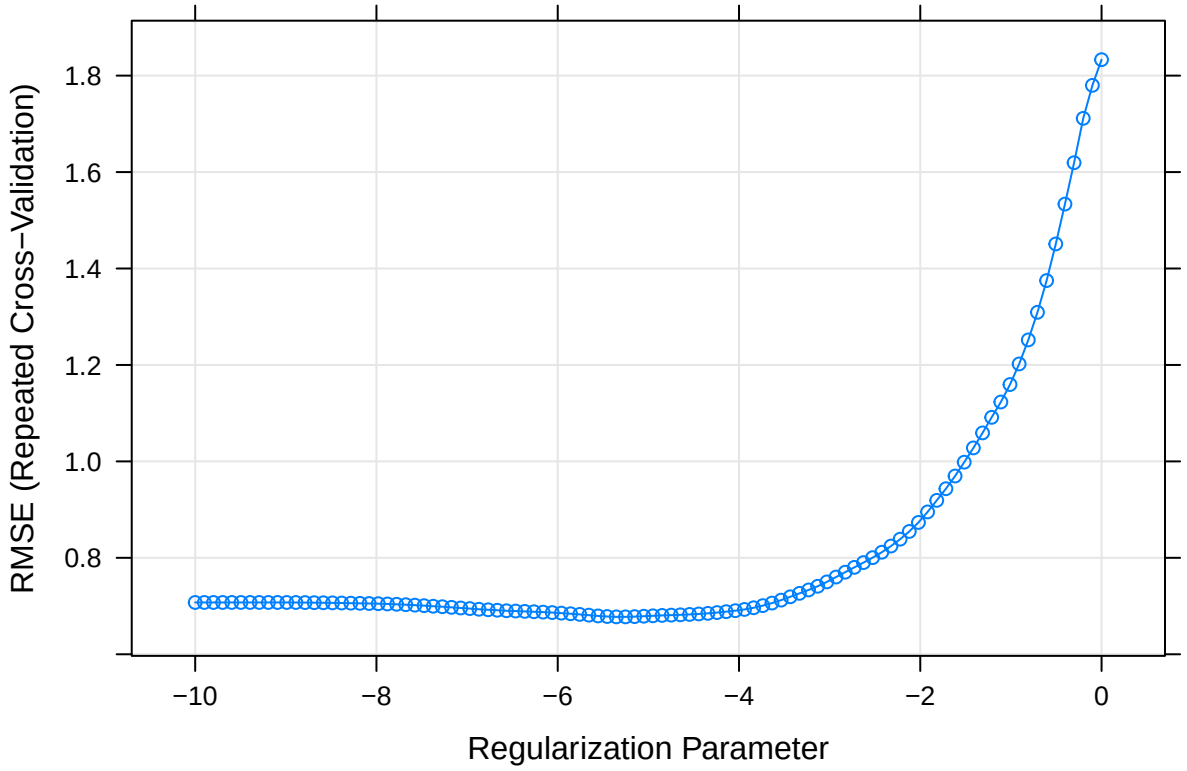
```
##    alpha    lambda
## 30     0 0.1260966
```

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

predict.ridge.fit <- predict(ridge.fit, newdata = test_dat)
ridge_mse <- mse(y.test, predict.ridge.fit)
```

The MSE of the ridge regression model on the test data is 0.5134603, with the chosen λ of 0.1260966.

Part C

```r
lasso.fit <- train(x, y,
                   method = "glmnet",
                   tuneGrid = expand.grid(alpha = 1,
                                          lambda = exp(seq(-10, 0, length = 100))),
                   trControl = control1
                   )

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




```
lasso.fit$bestTune$lambda
```

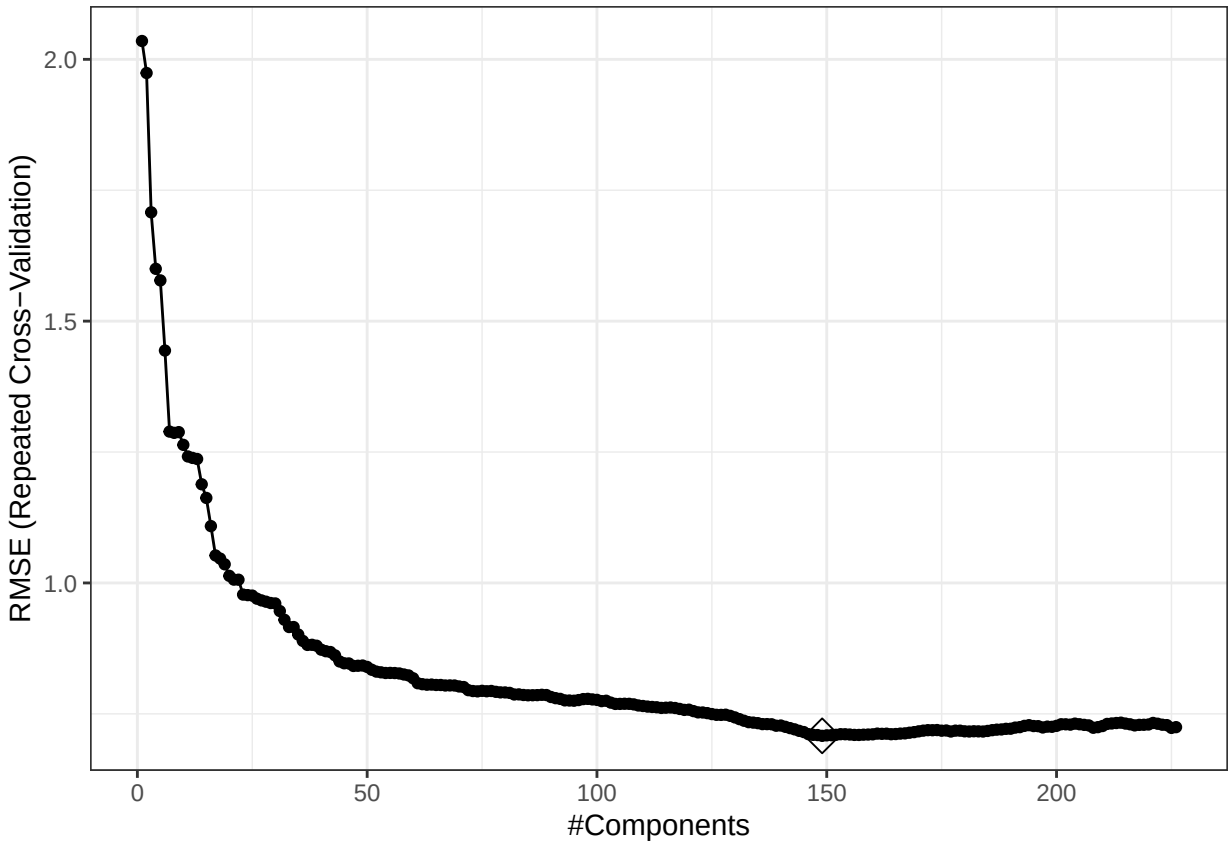
```
## [1] 0.005234284
```

```
coef_estimates <- coef(lasso.fit$finalModel,lasso.fit$bestTune$lambda)
num_coef <- sum(as.vector(coef_estimates) != 0)

predict.lasso.fit <- predict(lasso.fit, newdata = test_dat)
lasso_mse <- mse(y.test, predict.lasso.fit)
```

Using a λ of 0.0052343, the MSE of the lasso regression on the test data is 0.0052343. There are 144non-zero coefficient estimates.

Part D

```
set.seed(1)
pcr.fit <- train(x, y,
                 method = "pcr",
                 tuneGrid  = data.frame(ncomp = 1:226),
                 trControl = control1,
                 preProc = c("center", "scale"))

trans <- preProcess(x, method = c("center", "scale"))
predy2.pcr2 <- predict(pcr.fit$finalModel, newdata = predict(trans, x.test),
                       ncomp = pcr.fit$bestTune$ncomp)
pcr_mse <- mse(y.test, predy2.pcr2)

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



The MSE of the model using PCR on the test data is 0.540555, with M equal to 149.

Part E

```
mse_table <- tibble(
  model = c("Linear", "Ridge", "Lasso", "PCR"),
  mse = c(linear_mse, ridge_mse, lasso_mse, pcr_mse)
)
```

Based off the table of MSE's derived from each model, we can see that lasso has the lowest mse.

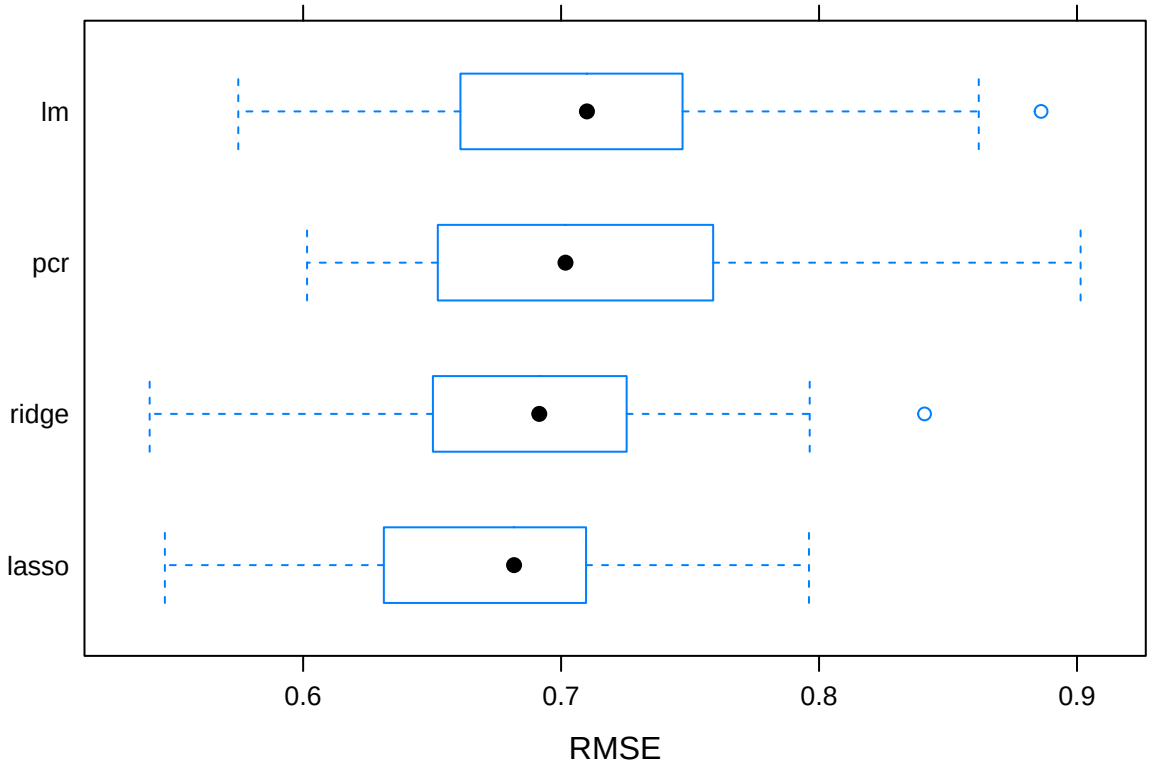
Part F

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

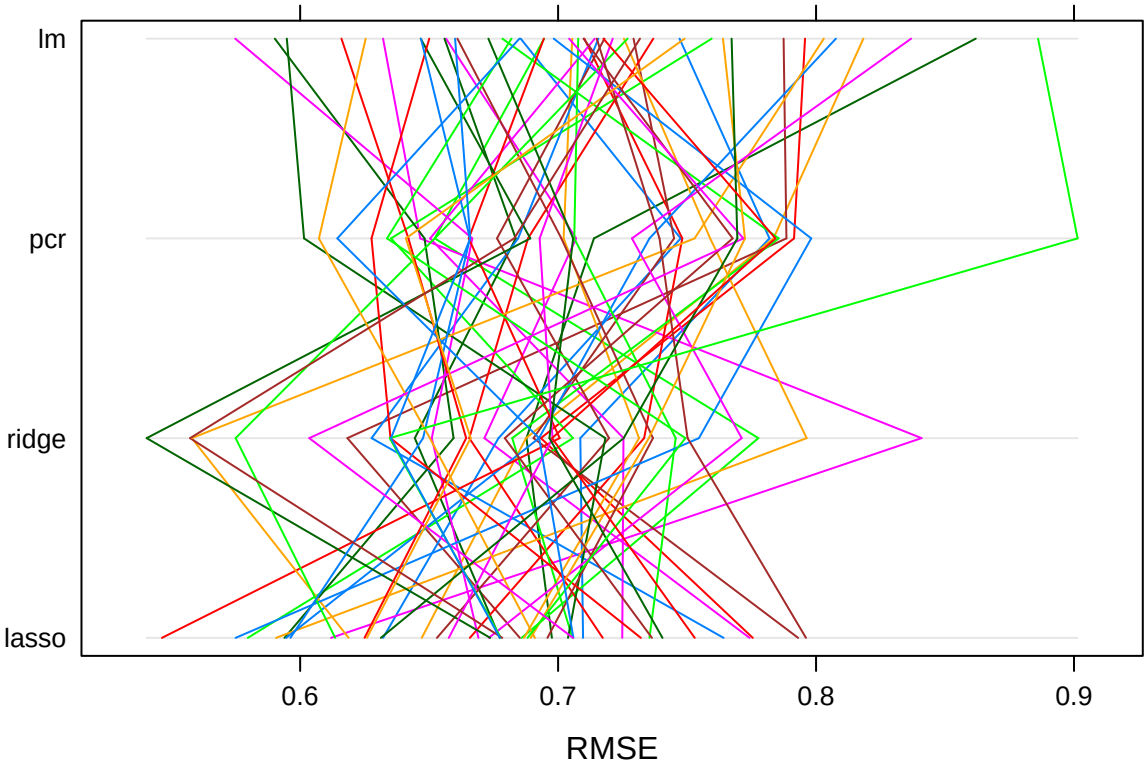
```
##
## Call:
## summary.resamples(object = resamp)
##
## Models: lasso, ridge, pcr, lm
## Number of resamples: 50
```

```
##
## MAE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.4282768 0.4913853 0.5211405 0.5184036 0.5470334 0.6275593    0
## ridge 0.4192301 0.4985993 0.5233721 0.5234005 0.5523380 0.6398975    0
## pcr   0.4493418 0.5021588 0.5417937 0.5427523 0.5736116 0.7111992    0
## lm    0.4277377 0.4915503 0.5249318 0.5310003 0.5530262 0.7069382    0
##
## RMSE
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.5463913 0.6314975 0.6817604 0.6775705 0.7087330 0.7960964    0
## ridge 0.5404832 0.6504855 0.6915303 0.6861373 0.7253943 0.8409048    0
## pcr   0.6015178 0.6554806 0.7016990 0.7080427 0.7574392 0.9014038    0
## lm    0.5748467 0.6639700 0.7100120 0.7115004 0.7445387 0.8860734    0
##
## Rsquared
##            Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## lasso 0.8341586 0.8753556 0.8943625 0.8903736 0.9079979 0.9341288    0
## ridge 0.8194326 0.8713551 0.8876903 0.8865597 0.9057885 0.9380168    0
## pcr   0.8238483 0.8687718 0.8818169 0.8815807 0.8990229 0.9200157    0
## lm    0.8208162 0.8630037 0.8851720 0.8809016 0.9009296 0.9202780    0
```

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




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



Based off the MSE, box plot, and RMSE summary, I would use the lasso model for predicting purposes, as it has the lowest MSE out of all the models.