R Notebook

Code

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

Lasso Regression

1. Use Split Dataset

Red Dataset:

Hide

```
library(glmnet)

# Red Datasets
red <- read.csv('winequality-red.csv', header = TRUE, sep=";")
red <- na.omit(red)
red.quality <- red$quality
red[,-12] <- scale(red[,-12]) # scale/standardization

train_red_idx <- sample(nrow(red) * 0.8) # 80-20 train-test split
train.red <- red[train_red_idx,]
train.red.quality <- train.red$quality
train.x.red <- model.matrix(quality~., train.red)

test.red <- red[-train_red_idx,]
test.red.quality <- test.red$quality
test.x.red <- model.matrix(quality~., test.red)

# Ridge Regression
start = proc.time()
cv.ridge = cv.glmnet(train.x.red, train.red.quality, alpha=0)
proc.time() - start
```

```
   user  system elapsed 
  0.180   0.020   0.203 
```

Hide

```
par(mfrow=c(1,1))
cv.ridge
```

```


Call:  cv.glmnet(x = train.x.red, y = train.red.quality, alpha = 0) 

Measure: Mean-Squared Error 

    Lambda Index Measure      SE Nonzero
min 0.0406   100  0.4225 0.01700      11
1se 0.4556    74  0.4391 0.01472      11
```

Hide

```
plot(cv.ridge, main='Test MSE vs Lambda (Ridge, Red)')

bestlam.red = cv.ridge$lambda.min
bestlam.red
```

```
[1] 0.04055818
```

Hide

```
start = proc.time()
ridge.best.red = glmnet(train.x.red, train.red.quality, alpha=0, lambda=bestlam.red)
proc.time() - start
```

```
   user  system elapsed 
  0.021   0.002   0.024 
```

Hide

```
ridge.pred.red = predict(ridge.best.red, test.x.red)
test_acc.red.ridge <- mean(abs(ridge.pred.red - test.red.quality) <= 0.5) # test accuracy (prediction within 0.5
of ground truth)
test_acc.red.ridge
```

```
[1] 0.640625
```

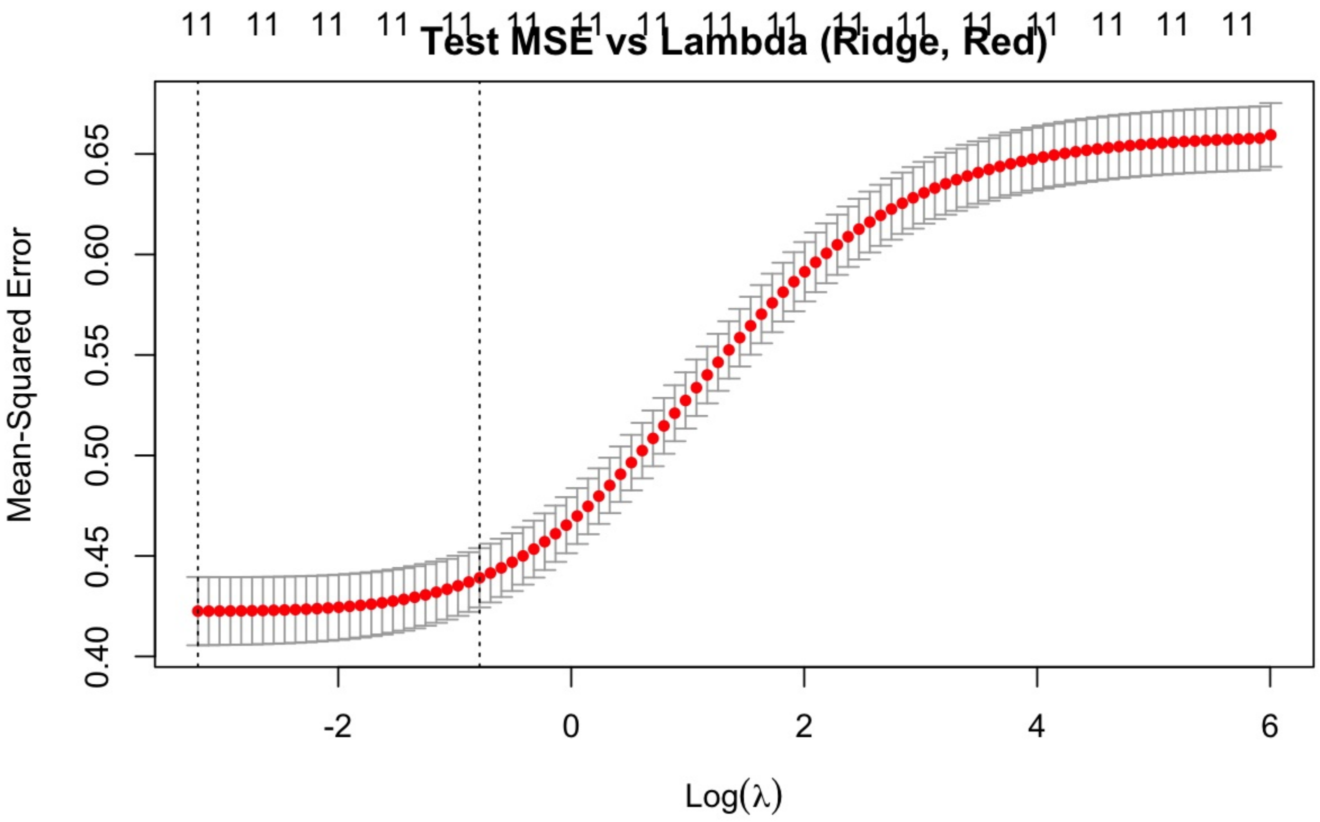
Hide

```
# Lasso Regression
start = proc.time()
cv.lasso = cv.glmnet(train.x.red, train.red.quality, alpha=1)
proc.time() - start
```

```
   user  system elapsed 
  0.115   0.010   0.131 
```

Hide

```
par(mfrow=c(1,1))
```



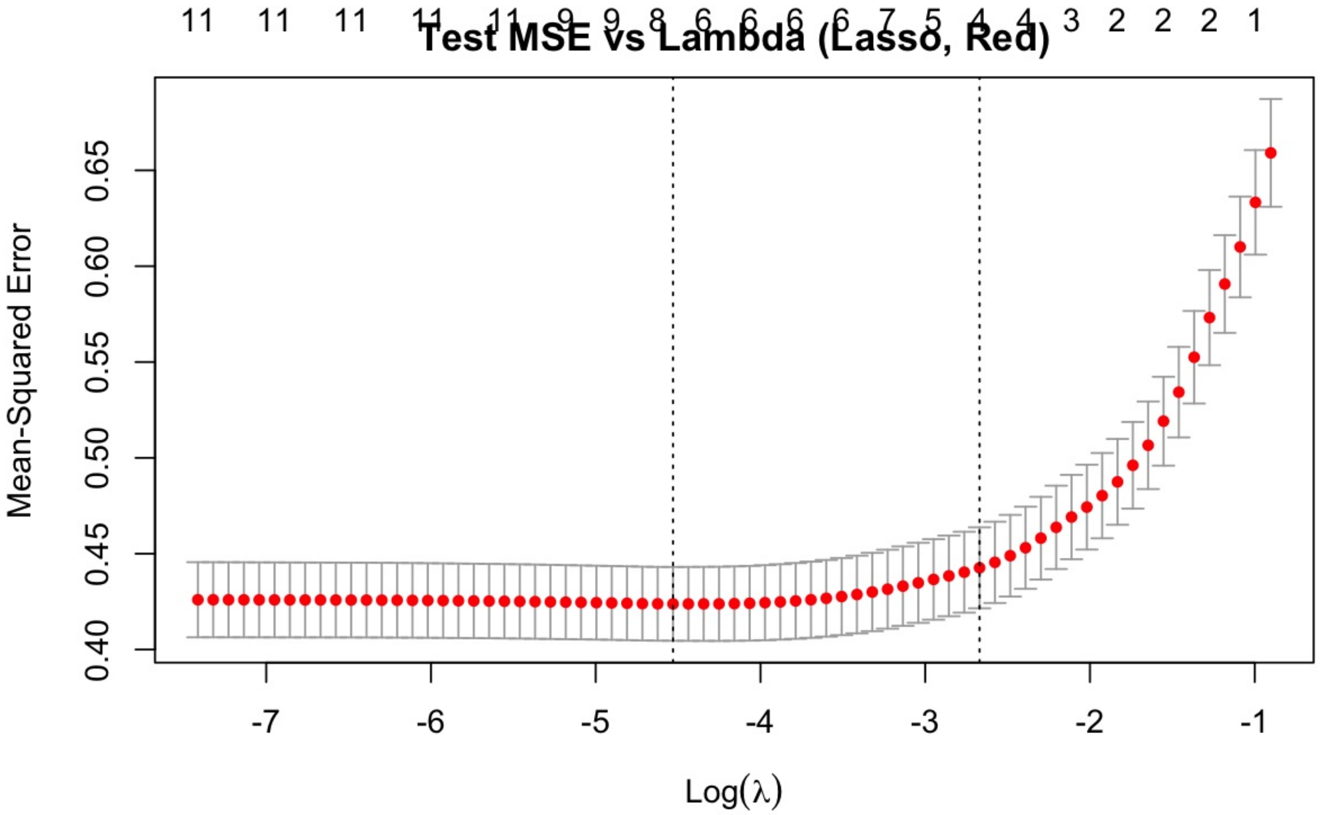
Hide

```
cv.lasso
```

```

Call:  cv.glmnet(x = train.x.red, y = train.red.quality, alpha = 1) 

Measure: Mean-Squared Error 

     Lambda Index Measure      SE Nonzero
min 0.01077    40  0.4238 0.01925       8
1se 0.06925    20  0.4426 0.02113       4
```

Hide

```
plot(cv.lasso, main='Test MSE vs Lambda (Lasso, Red)')
```



Hide

```
bestlam.red = cv.lasso$lambda.min
bestlam.red
```

```
[1] 0.01077261
```

Hide

```
start = proc.time()
lasso.best.red = glmnet(train.x.red, train.red.quality, alpha=1, lambda=bestlam.red)
proc.time() - start
```

```
   user  system elapsed 
  0.021   0.003   0.023 
```

Hide

```
lasso.pred.red = predict(lasso.best.red, test.x.red)
test_acc.red.lasso <-  mean(abs(lasso.pred.red - test.red.quality) <= 0.5) # test accuracy (prediction within 0.5
of ground truth)
test_acc.red.lasso
```

```
[1] 0.646875
```

White Dataset:

Hide

```
white <- read.csv('winequality-white.csv', header = TRUE, sep=";")
white <- na.omit(white)
white.quality <- white$quality
white[,-12] <- scale(white[,-12]) # scale/standardization

set.seed(1)
train_white_idx <- sample(nrow(white) * 0.8) # 80-20 train-test split
train.white <- white[train_white_idx,]
# train.white$quality <- as.factor(train.white$quality)
train.white.quality <- train.white$quality
train.x.white <- model.matrix(quality~., train.white)

test.white <- white[-train_white_idx,]
# test.white$quality <- as.factor(test.white$quality)
test.white.quality <- test.white$quality
test.x.white <- model.matrix(quality~., test.white)

# Ridge Regression
start = proc.time()
cv.ridge = cv.glmnet(train.x.white, train.white.quality, alpha=0)
proc.time() - start
```

```
   user  system elapsed 
  0.294   0.032   0.383 
```

Hide

```
par(mfrow=c(1,1))
cv.ridge
```

```


Call:  cv.glmnet(x = train.x.white, y = train.white.quality, alpha = 0) 

Measure: Mean-Squared Error 

     Lambda Index Measure      SE Nonzero
min 0.04141   100  0.5888 0.01888      11
1se 0.26619    80  0.6058 0.01844      11
```

Hide

```
plot(cv.ridge, main='Test MSE vs Lambda (Ridge, White)')

bestlam.white = cv.ridge$lambda.min
bestlam.white
```

```
[1] 0.04141091
```

Hide

```
start = proc.time()
ridge.best.white = glmnet(train.x.white, train.white.quality, alpha=0, lambda=bestlam.white)
proc.time() - start
```

```
   user  system elapsed 
  0.018   0.003   0.022 
```

Hide

```
ridge.pred.white = predict(ridge.best.white, test.x.white)
test_acc.white.ridge <- mean(abs(ridge.pred.white - test.white.quality) <= 0.5) # test accuracy (prediction withi
n 0.5 of ground truth)
test_acc.white.ridge
```

```
[1] 0.5408163
```

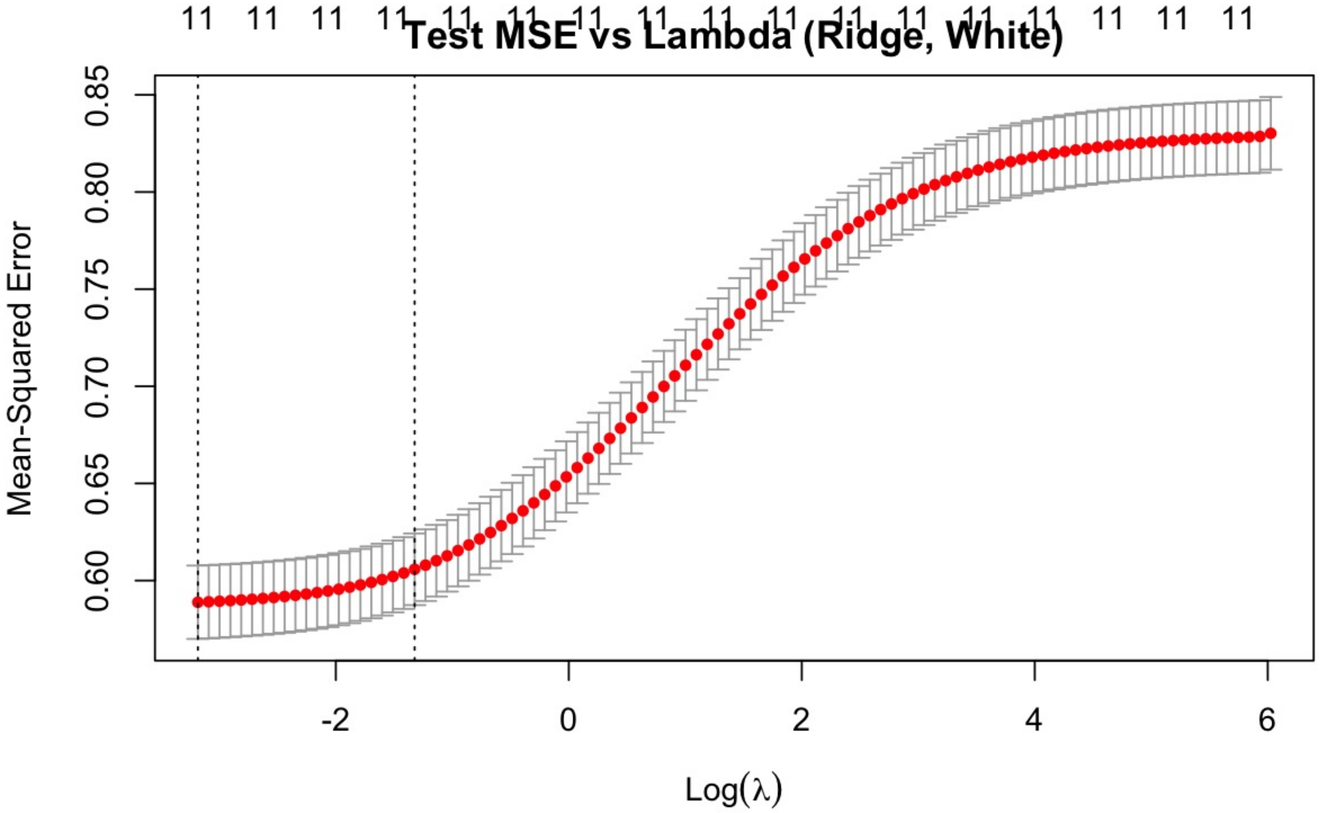
Hide

```
# Lasso Regression
start = proc.time()
cv.lasso = cv.glmnet(train.x.white, train.white.quality, alpha=1)
proc.time() - start
```

```
   user  system elapsed 
  0.406   0.032   0.517 
```

Hide

```
par(mfrow=c(1,1))
```



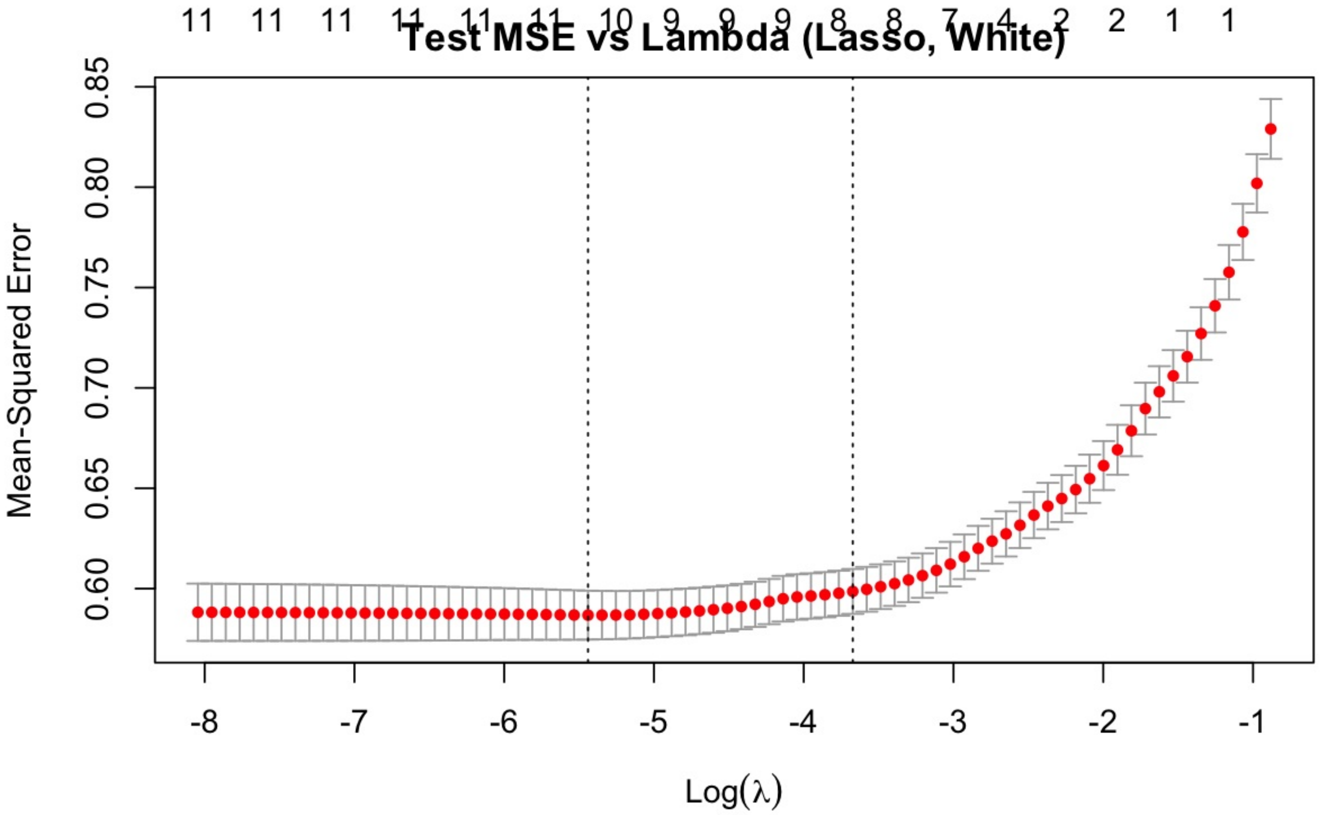
Hide

```
cv.lasso
```

```


Call:  cv.glmnet(x = train.x.white, y = train.white.quality, alpha = 1) 

Measure: Mean-Squared Error 

      Lambda Index Measure      SE Nonzero
min 0.004338    50  0.5868 0.01220      10
1se 0.025409    31  0.5986 0.01119       8
```

Hide

```
plot(cv.lasso, main='Test MSE vs Lambda (Lasso, White)')
```



Hide

```
bestlam.white = cv.lasso$lambda.min
bestlam.white
```

```
[1] 0.004338272
```

Hide

```
start = proc.time()
lasso.best.white = glmnet(train.x.white, train.white.quality, alpha=1, lambda=bestlam.white)
proc.time() - start
```

```
   user  system elapsed 
  0.021   0.004   0.034 
```

Hide

```
lasso.pred.white = predict(lasso.best.white, test.x.white)
test_acc.white.lasso <- mean(abs(lasso.pred.white - test.white.quality) <= 0.5) # test accuracy (prediction withi
n 0.5 of ground truth)
test_acc.white.lasso
```

```
[1] 0.5316327
```

Weighted Accuracy:

Hide

```
(test_acc.white.ridge * nrow(test.white) + test_acc.red.ridge * nrow(test.red)) / ( nrow(test.white) +  nrow(test
.red))
```

```
[1] 0.5653846
```

Hide

```
(test_acc.white.lasso * nrow(test.white) + test_acc.red.lasso * nrow(test.red)) / ( nrow(test.white) +  nrow(test
.red))
```

```
[1] 0.56
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.