Exam 3

Tom Wilson

Dember 12th, 2018

Required R packages

```
library(tidyverse)
library(data.table)
library(glmnet)
library(glmnetUtils)
library(MASS)
library(caret)
library(leaps)
library(boot)
library(knitr)
```

1.

Predict the number of applications received using the other variables in the college data set.

```
college <- fread('../data/College.csv') %>% subset(,-V1)
```

a.

Split the data set into a training set and a test set.

```
n=nrow(college)
train_sample <- runif(n,0,1) > 1 - 0.75 #random uniform sample
college_train <- college[ train_sample,] %>% select_if(is.numeric)
college_test  <- college[!train_sample,] %>% select_if(is.numeric)
```

```
x_train <- college_train %>% subset(,-Apps) # all columns except Apps
y_train <- college_train %>% subset(, Apps)
x_test  <- college_test  %>% subset(,-Apps)
y_test  <- college_test  %>% subset(, Apps)
```

b.

Fit a linear model using least squares on the training set, and report the test error obtained.

```
linear_model <- lm(formula = Apps~., data = college_train )
```

```
residual <- predict(linear_model,newdata = college_test) - college_test$Apps
RMSE <- sqrt(sum(residual^2))
```

Using linear regression, Root Mean Squared Error on test dataset is

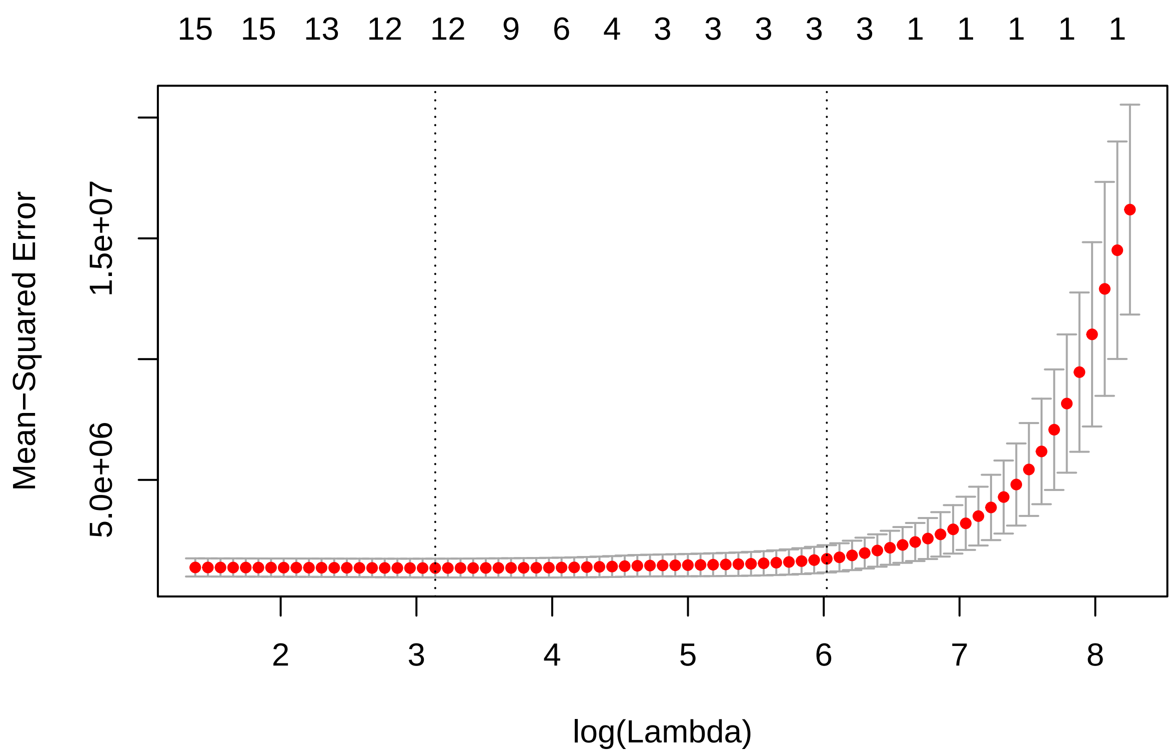
```
## [1] 16948.86
```

c.

Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

using glmnetUtils

```
fit <- cv.glmnet(Apps ~ ., data=college_train)
plot(fit)
```




```
fit
```

```
## Call:
## cv.glmnet.formula(formula = Apps ~ ., data = college_train)
##
## Model fitting options:
##     Sparse model matrix: FALSE
##     Use model.frame: FALSE
##     Number of crossvalidation folds: 10
##     Alpha: 1
##     Deviance-minimizing lambda: 23.07171  (+1 SE): 412.672
```

MSE is minimized at λ of

```
## [1] 23.07171
```

```
best.fit <- glmnet(Apps ~ ., data=college_train, lambda=fit$lambda.min)
pred <- predict(best.fit,newdata=college_test)
residual <- pred - college_test$Apps
```

```
RMSE <- sqrt(sum(residual^2))
```

Using ridge regression, Root Mean Squared Error on test dataset is

```
## [1] 16764.47
```

Which is slightly worse than ordinary least squares shown previously.

2.

Consider the Boston housing data set, from the MASS library.

```
Boston %>% head() %>% kable()
```

crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
0.02985	0	2.18	0	0.458	6.430	58.7	6.0622	3	222	18.7	394.12	5.21	28.7

a.

Based on this data set, provide an estimate for the population mean of medv. Call this estimate.

```
mu_hat <- mean(Boston$medv)
```

An estimate for the population mean of medv, $\hat{\mu}$ is

```
round(mu_hat,4)
```

```
## [1] 22.5328
```

b.

Provide an estimate of the standard error of Interpret this result. Hint: We can compute the standard error of the sample mean by dividing the sample standard deviation by the square root of the number of observations.

```
n <- nrow(Boston)
SE <- sd(Boston$medv)/sqrt(n)
```

an estimate of the standard error of the population mean of medv is

```
## [1] 0.4089
```

c.

Now estimate the standard error of using the bootstrap. How does this compare to your answer from (b)?

```
SE.fn=function(data,index){
    n=length(data[index])
 return(sd(data[index])/sqrt(n))
 }
```

```
SE_bootstrap <- boot(data = Boston$medv,statistic = SE.fn,R=1000)
```

the average bootstrap estimate of SE is

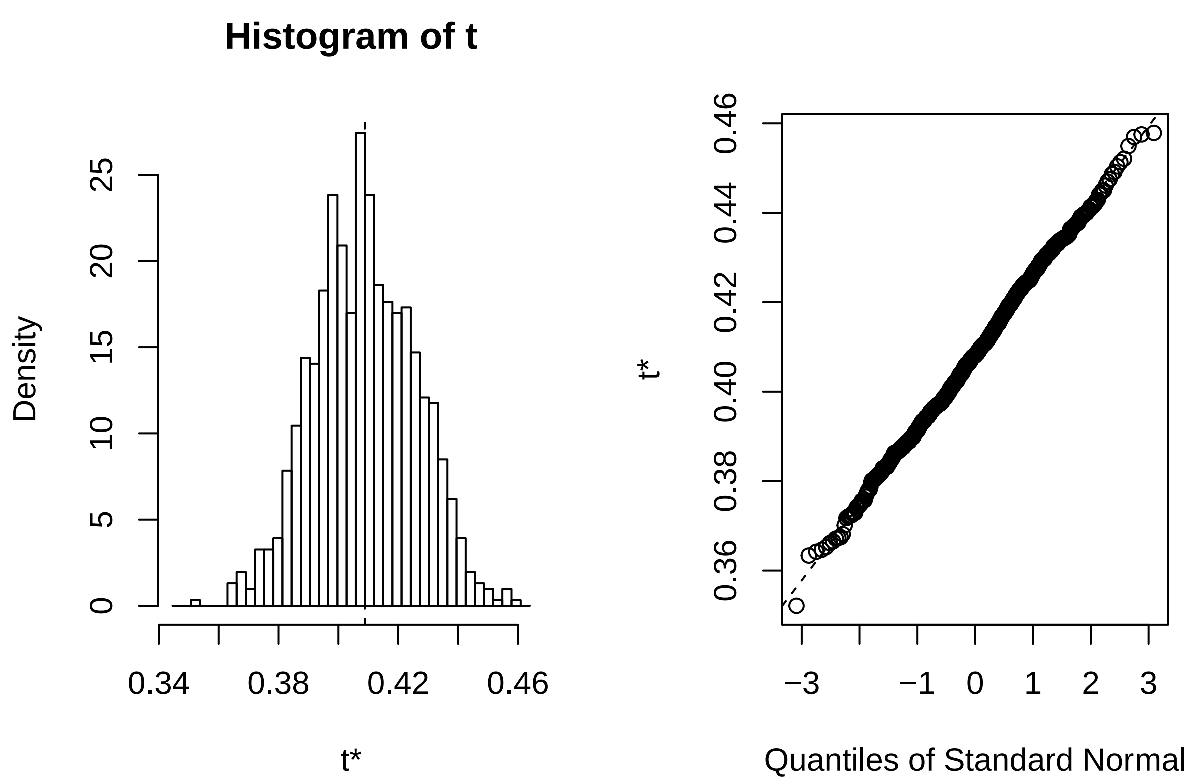
```
## [1] 0.4086
```

which differs from the original estimate by

```
## [1] -0.06
```

percent

```
plot(SE_bootstrap)
```



d.

Based on your bootstrap estimate from (c), provide a 95% confidence interval for the mean of medv. compare it to the results obtained using t.test(Boston$medv).

```
n <- nrow(Boston)
mu <- mean(Boston$medv)
SE <- mean(SE_bootstrap$t)

alpha <- 1 - 0.95
z <- qnorm(1 - alpha/2)
error <- z*SE

ttest <- t.test(Boston$medv)$conf.int
```

```
## [1] "using z stat, the 95% confidence interval is between 21.732 and 23.3336"
## [1] "using 2*SE, the 95% confidence interval is between 21.7156 and 23.35"
## [1] "using t.test, the 95% confidence interval is between 21.7295 and 23.3361"
```