HW05_#4

Philip Lee

11/4/2019

#Part A

I believe that ridge would perform better than the lasso in terms of model accuracy as ridge will not make the coefficients as 0 for predictors that are deemed insignificant. In addition, we do not know which predictors are significant. Ridge regression keeps all the predictors, which helps in this case as all of these predictors could potentially affect the reponse variable.

#Part B

```
set.seed(1888)
x<-model.matrix(Fertility~.,data)[,-1]   ##removes the first column as that's
##where the response variable is at
y<-data$Fertility ##stores fertiliy
sample.data<-sample.int(nrow(data), floor(.50*nrow(data)), replace = F) ##splits the data in halves
x.train<-x[sample.data,]
x.test<-x[-sample.data,]
y.train<-y[sample.data]
y.test<-y[-sample.data]
```

#Part C

```
result<-lm(Fertility~.,data)     ##fits OLS in the data
ridge.r<-glmnet(x,y,alpha=0, lambda=0, thresh = 1e-14) ##Fits the ridge regression to the data
coefficients(ridge.r)  ##gets the coefficients for both methods to compare
```

```
## 6 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)      66.9151817
## Agriculture      -0.1721140
## Examination      -0.2580083
## Education        -0.8709400
## Catholic          0.1041153
## Infant.Mortality  1.0770482
```

```
coefficients(result)
```

```
##      (Intercept)      Agriculture      Examination        Education
##       66.9151817       -0.1721140       -0.2580082       -0.8709401
##         Catholic Infant.Mortality
##        0.1041153        1.0770481
```

If the coefficients produced from the lm() funciton is same as the coefficients produced from the glmnet() function, then we know that we chose an appropriate value for thresh argument. In this case, we know the thresh has an appropriate value as the coefficients produced from both functions are essentially the same.

#Part D

```
set.seed(2019)
cv.out<-cv.glmnet(x.train,y.train,alpha=0)  ##performs cross-validation
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations
## per fold
```

```r
bestlam.ridge<-cv.out$lambda.min  ##finds the best lambda for ridge regression
print(bestlam.ridge)
```

```
## [1] 3.392629
```

3.3926 was the value of lambda chosen for cv.

#Part E

```r
grid<-10^seq(10,-2,length=100)    ##creates a range of values
ridge.mod<-glmnet(x.train,y.train,alpha=0,lambda=grid, thresh = 1e-14) ##fit ridge regression for
##a range of lambdas
ridge.pred<-predict(ridge.mod,s=bestlam.ridge,newx=x.test) ##gets the predicted values based on ridge
mean((ridge.pred-y.test)^2) ##calculates the test MSE
```

```
## [1] 50.98311
```

50.9831 for test MSE for ridge

#Part F

```r
set.seed(2019)
cv.out<-cv.glmnet(x.train,y.train,alpha=1)  ##performs cross-validation
```

```
## Warning: Option grouped=FALSE enforced in cv.glmnet, since < 3 observations
## per fold
```

```r
bestlam.lasso<-cv.out$lambda.min  ##finds the best lambda for lasso
print(bestlam.lasso)
```

```
## [1] 0.168852
```

```r
lasso.mod<-glmnet(x.train,y.train,alpha=1,lambda=grid,thresh = 1e-14)  ##fit lasso regression
##for a range of lambdas
lasso.pred<-predict(lasso.mod,s=bestlam.lasso,newx=x.test)  ##gets predicted values for lasso
mean((lasso.pred-y.test)^2)  ##calculates test MSE for lasso
```

```
## [1] 71.9308
```

Lambda chosen for lasso is 0.1689. 71.9308 for test MSE for Lasso

#Part G

```r
ols.pred<-predict(ridge.mod,s=0,newx=x.test)  ##gets the predicted values for OLS
mean((ols.pred-y.test)^2)  ##calculates OLS test MSE
```

```
## [1] 81.8142
```

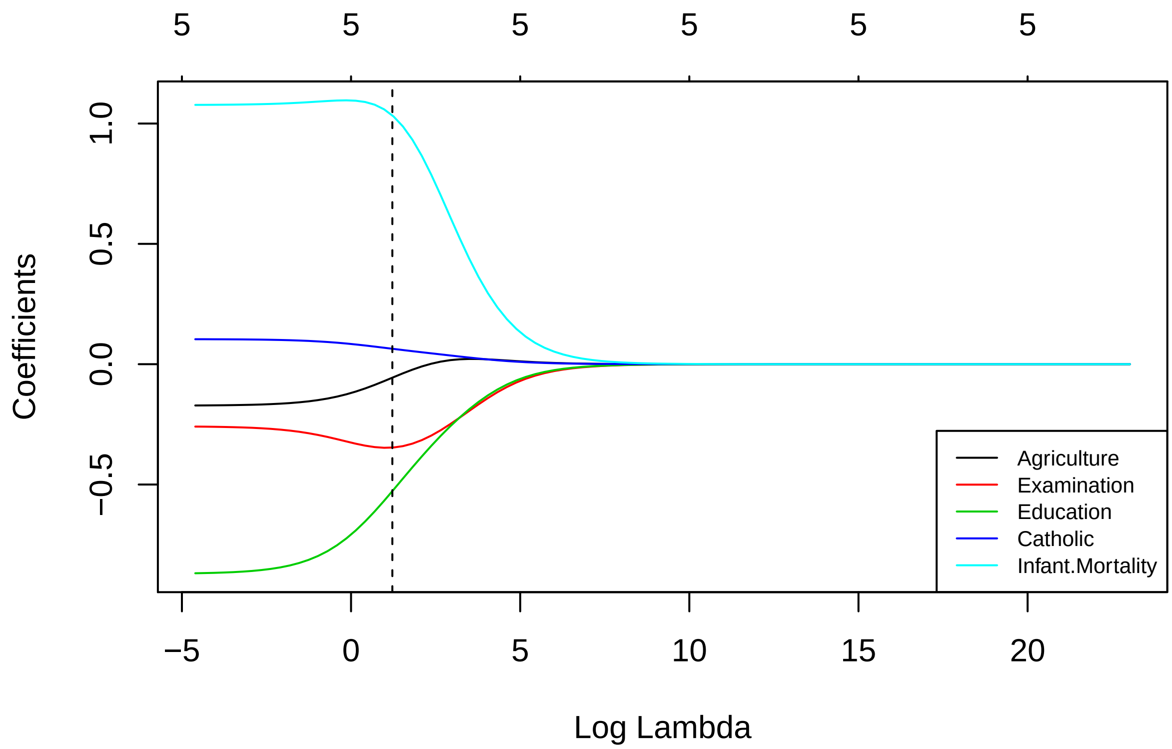
81.8142 for OLS test MSE

#Part H

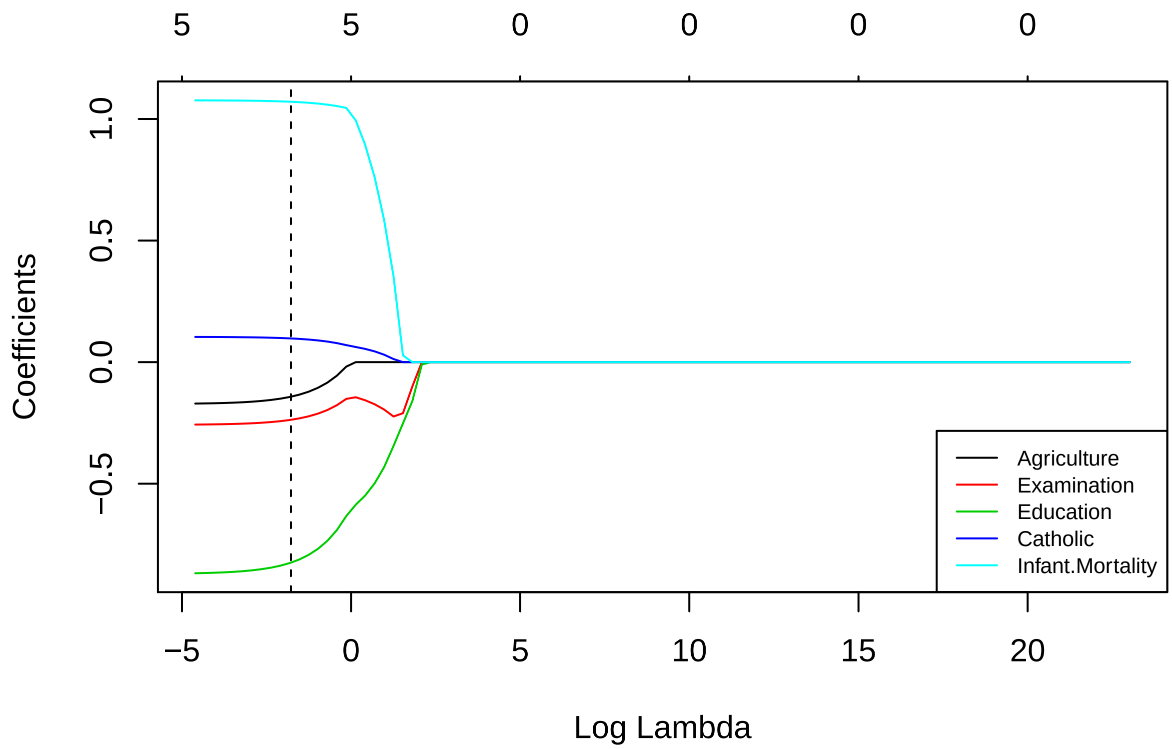
Ridge has the lowest MSE of 50.9831, indicating that it has the best accuracy when we are comparing only the test MSEs.

#Part I

```r
out.all<-glmnet(x,y,alpha=0,lambda=grid,thresh = 1e-14)  ##plots the coefficients against
##lambdas for ridge and lasso
plot(out.all, xvar = "lambda")
abline(v=log(bestlam.ridge), lty=2)
legend("bottomright", lwd = 1, col = 1:6, legend = colnames(x), cex = .7)
```




```
out.all2<-glmnet(x,y,alpha=1,lambda=grid,thresh = 1e-14)
plot(out.all2, xvar = "lambda")
abline(v=log(bestlam.lasso), lty=2)
legend("bottomright", lwd = 1, col = 1:6, legend = colnames(x), cex = .7)
```



First graph is for ridge regression while the second graph is for lasso. Coefficients shrinks to 0 as lamda increases for both ridge and lasso. Lasso shrinks the coefficients to 0 at lower values of lambda.