Application

Refer to the Air Quality data described previously, and the analyses we have done with Ozone as the response variable, and the five explanatory variables (including the two engineered features).

1. Use ridge regression on the data:
(a) Using GCV, identify the optimal _ from the sequence 0 to 100 by 0.05. **Report the optimal value for *lambda*.**

```
library(dplyr)
library(MASS)   # For ridge regression
library(glmnet) # For LASSO

data = na.omit(airquality[, 1:4])
data$TWcp = data$Temp*data$Wind
data$TWrat = data$Temp/data$Wind

set.seed(93425633)


### Create a container for MSPEs. Let's include ordinary least-squares
### regression for reference
all.models = c("Ridge", "LASSO-Min", "LASSO-1se")
all.MSPEs = array(0, dim = c(K, length(all.models)))
colnames(all.MSPEs) = all.models

# Ridge Regression
lambda.vals = seq(from = 0, to = 100, by = 0.05)

fit.ridge = lm.ridge(Ozone ~ ., lambda = lambda.vals,
                     data = data)


### Get best lambda value and its index
### Note: Best is chosen according to smallest GCV value. We can
###       get GCV from a ridge regression object using $GCV
ind.min.GCV = which.min(fit.ridge$GCV)
lambda.min = lambda.vals[ind.min.GCV]

# Optimal value for lambda: 0.2
```

(b) **Report the parameter estimates for the optimal model, and compare them to those from least squares. Are the ridge estimates all smaller.**

```
### Get coefficients corresponding to best lambda value
### We can get the coefficients for every value of lambda using
### the coef() function on a ridge regression object
all.coefs.ridge = coef(fit.ridge)
coef.min = all.coefs.ridge[ind.min.GCV,]
#                  Solar.R          Wind          Temp          TWcp         TWrat
# -161.63123617    0.06284069    6.82529896    2.45651021   -0.10883212    1.64685249
```

```
# Lm parameters
fit.ls = lm(Ozone ~ ., data = data)
# (Intercept)      Solar.R         Wind         Temp         TWcp        TWrat
# -191.19856      0.06384      9.56187      2.89466     -0.14751      1.36619
```

No, not all ridge estimates smaller.

2. Use LASSO on the data

(a) Using default CV in cv.glmnet, identify the optimal *lambda*, $lambda_{min}$, and the "1SE" *lambda*, $lambda_{1SE}$. **Report both numbers.**

```
matrix.train.raw = model.matrix(Ozone ~ ., data = data)
matrix.train = matrix.train.raw[,-1]

all.LASSOs = cv.glmnet(x = matrix.train, y = data$Ozone)

### Get both 'best' lambda values using $lambda.min and $lambda.1se
lambda.min = all.LASSOs$lambda.min
lambda.1se = all.LASSOs$lambda.1se
#lambda: 0.366 lambda.min 0.3658798
#lambda: 7.182 lambda.1se 7.18237
```

(b) **For both selections of _, report the parameter estimates for the optimal model. Comment on how the two sets of estimates differ from each other.**

```
### Get the coefficients for our two 'best' LASSO models
coef.LASSO.min = predict(all.LASSOs, s = lambda.min, type = "coef")
# (Intercept) -87.08234278
# Solar.R       0.05712184
# Wind          .
# Temp          1.36664563
# TWcp         -0.01234339
# TWrat         2.29832095

coef.LASSO.1se = predict(all.LASSOs, s = lambda.1se, type = "coef")
# (Intercept) -51.4433762
# Solar.R       .
# Wind          .
# Temp          0.9551653
# TWcp          .
# TWrat         2.0423891
```

LASSO.min drops Wind variable and LASSO.1se drops Solar.R, Wind, and TWcp

(c) Compare the variables selected by both versions of LASSO to those from the hybrid stepwise from Lecture 5. **What differences are there?**

```
### Step wise
fit.start = lm(Ozone ~ 1, data = data)
fit.end = lm(Ozone ~ ., data = data.train)

step.BIC = step(fit.start, list(upper = fit.end), k = log(n.train), trace = 0)

# (Intercept)        TWrat         Temp      Solar.R
# -93.3042       2.8633       1.2523       0.0596
```

Lasso.min selects Solar.R, Temp, Twcp, and TWrat as variables.
Lasso.1se selects Temp and TWrat as variables
Hybrid stepwise selects TWrat, Temp, and Solar.R as variables.

Lasso.1se seems to be the simplest one.