Problem Set 7, Fall 2021

Ben Karabinus

```
# Load any packages, if any, that you use as part of your answers here
# For example:

library(mlbench)
library(glmnet)

## Loading required package: Matrix

## Loaded glmnet 4.1-2
```

CONTEXT - HOUSE VALUES IN BOSTON, CIRCA 1970

This dataset was obtained through the mlbench package, which contains a subset of data sets available through the UCI Machine Learning Repository. From the help file:

Housing data for 506 census tracts of Boston from the 1970 census. The dataframe BostonHousing contains the original data by Harrison and Rubinfeld (1979).

The original data are 506 observations on 14 variables.

Continuous variables:

crim per capita crime rate by town zn proportion of residential land zoned for lots over 25,000 sq.ft
indus proportion of non-retail business acres per town nox nitric oxides concentration (parts per 10 million) rm average number of rooms per dwelling age proportion of owner-occupied units built prior to 1940 dis weighted distances to five Boston employment centres rad index of accessibility to radial highways tax full-value property-tax rate per USD 10,000 ptratio pupil-teacher ratio by town b 1000(B - 0.63)^2 where B is the proportion of blacks by town lstat percentage of lower status of the population medv median value of owner-occupied homes in USD 1000's

Categorical variables:

chas Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)

Question 1 - 10 points

The BostonHousing data is contained inside of an R package, so you'll load the data into memory a little differently than usual. Run the following code chunk, confirm that the data in loaded into memory, and ensure that your variables are of the proper type (they should be)

```
# load the Boston housing data
data(BostonHousing) # loads the BostonHousing dataset into memory from the ml
bench package
# verify variable types
str(BostonHousing)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.5
24 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : num  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : num  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ b      : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

For this question, conduct a cross-validated ridge regression. Use medv as the outcome and all of the other variables in the data set as the predictors. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated ridge regression. Be sure to use the set.seed() provided to make your analysis reproducible (in theory).

```
# create the matrix of explanatory variables
X <- model.matrix(medv ~., data=BostonHousing)
# drop the intercept column
X <- X[,-1]
# create the vector of outcomes
Y<-BostonHousing$medv
# set seed for reproducability
set.seed(1000)
# create cross validated ridge regression model (alpha = 0 for ridge)
cvfit.house.ridge <- cv.glmnet(x=X, y=Y,alpha=0)
```

Next, display the value for lambda.min *and* the coefficients associated with it. Make sure these are visible in your knitted document.

```
# print the smallest lambda for the fitted model
cvfit.house.ridge$lambda.min

## [1] 0.6777654

# print coefficients associated with smallest lambda
coef(cvfit.house.ridge, s = "lambda.min")

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)  28.001475781
## crim         -0.087572712
## zn            0.032681030
## indus        -0.038003639
## chas1         2.899781646
## nox         -11.913360447
## rm            4.011308386
## age          -0.003731470
## dis          -1.118874605
## rad           0.153730052
## tax          -0.005751054
## ptratio      -0.854984614
## b             0.009073740
## lstat        -0.472423800
```

Finally, display the value for lambda.1se *and* the coefficients associated with it. Again, make sure these are visible in your knitted document.

```
# print lambda with the lowest MSE
cvfit.house.ridge$lambda.1se

## [1] 3.002922

# print coefficients computed for lambda.lse
coef(cvfit.house.ridge, s = "lambda.1se")

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) 21.360545954
## crim        -0.070614787
## zn           0.021786752
## indus       -0.065928119
## chas1        2.842394463
## nox         -6.204013805
## rm           3.777295539
## age         -0.007121277
## dis         -0.619584773
## rad          0.048495418
## tax         -0.003063933
```

```
## ptratio     -0.708289189
## b            0.007996469
## lstat       -0.374059082
```

Question 2 - 10 points

For this question, you will use the same outcome (medv) and the same predictors in the as in the last question, but you will instead conduct a cross-validated lasso regression. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated lasso regression. Be sure to use the set.seed() provided to make your analysis reproducible (again, in theory).

```
# create the matrix of explanatory variables
X<-model.matrix(medv~., data=BostonHousing)
# drop the intercept column
X <- X[,-1]
# create the vector of outcomes
Y<-BostonHousing$medv
# set seed for reproducability
set.seed(1000)
# create lasso regression model using k- fold cross validation
cvfit.house.lasso <- cv.glmnet(x=X, y=Y, alpha=1)
```

Next, display the value for lambda.min and the coefficients associated with it. Make sure these are visible in your knitted document.

```
# print the smallest lambda for the fitted model
cvfit.house.lasso$lambda.min

## [1] 0.02325053

# print coefficients associated with smallest lambda
coef(cvfit.house.lasso, s = "lambda.min")

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)  34.741082550
## crim         -0.099983351
## zn            0.042163911
## indus         .
## chas1         2.691362170
## nox         -16.484121376
## rm            3.856387746
## age           .
## dis          -1.412063666
## rad           0.260319555
## tax          -0.010144690
## ptratio      -0.932724941
```

```
## b             0.009070332
## lstat        -0.522516834
```

Finally, display the value for lambda.1se and the coefficients associated with it. Again, make sure these are visible in your knitted document.

```
# print the lambda with smallest MSE
cvfit.house.lasso$lambda.1se

## [1] 0.2611788

# print coefficients associated with lambda.1se
coef(cvfit.house.lasso, s = "lambda.1se")

## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) 21.248597748
## crim        -0.032386551
## zn           0.008396934
## indus        .
## chas1        2.244958387
## nox         -7.332177446
## rm           4.251522557
## age          .
## dis         -0.631213118
## rad          .
## tax          .
## ptratio     -0.815597187
## b            0.007076584
## lstat       -0.519864563
```

Question 3 - 5 points

An important difference between ridge regression and lasso regression is that predictors can be dropped from a model in lasso but not in ridge. The number of predictors set to zero (if any) in lasso depends on the extent of the coefficient shrinkage at a given lambda. Answer the two questions below about the results of your cross-validated lasso models.

1) Among the set of coefficients associated with *lambda.min* in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here:

The “indus” and “age” variables were set to zero for lambda.min in the lasso regression model.

2) Among the set of coefficients associated with *lambda.1se* in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here:

The “indus”, “age”, “rad” and “tax” variables were set to zero for lambda.1se in the lasso regression model.

3) Which of these - lambda.min or lambda.1se - had more coefficients set to zero?

Your answer here:

Lambda.lse had more coefficients set to zero.

CONTEXT - NATIONAL INDONESIA CONTRACEPTIVE PRELAVENCE SURVEY (1987)

This dataset was obtained from the UCI Machine Learning Repository. From the description on https://archive.ics.uci.edu/ml/datasets/Contraceptive+Method+Choice:

This dataset is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples are married women who were either not pregnant or do not know if they were at the time of interview. The problem is to predict the current contraceptive method choice (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

Continuous variables

Wife’s age (w.age) Number of children ever born (children)

Categorical variables:

Wife’s education (w.edu) 1=low, 2, 3, 4=high Husband’s education (h.edu) 1=low, 2, 3, 4=high Wife’s religion (w.relig) 0=Non-Islam, 1=Islam Wife’s now working? (w.work) 0=Yes, 1=No Husband’s occupation (h.occ) 1, 2, 3, 4 Standard-of-living index (sol.index) 1=low, 2, 3, 4=high Media exposure (media) 0=Good, 1=Not good Contraceptive method used (contra) 1=No-use, 2=Long-term, 3=Short-term

Question 4

First, load the data set into memory and change variables into their proper type.

```
contraception <- read.csv("contra.csv", header=TRUE, sep=",")
```

Next, re-code the contra variable such that non-use of contraception (contra=1) is equal to zero and use of any contraception (contra=2 or 3) is equal to one.

```
# re-code the contra variable
contraception$contra_use <- ifelse(contraception$contra == 1, 0, 1)
```

Now, run the code chunk below to use the table() function to verify that your recoded variable is correct and answer the question below it.

```
# sanity check re-coding
table(contraception$contra)

##
##   1   2   3
## 629 333 511

table(contraception$contra_use)

##
##   0   1
## 629 844
```

1) Does the number of 2's and 3's in the original contra variable match the number of 1's in your recoded outcome variable? (Hint: if not, you have a mistake in your recoding)

Your answer here (yes/no): Yes

One more thing to do: your variables probably all came in as integer, but some of them are categorical variables (included your recoded outcome). Change your variables to the appropriate type

```
# change categorical variables to type factor
contraception$w.edu <- as.factor(contraception$w.edu)
contraception$h.edu <- as.factor(contraception$h.edu)
contraception$w.relig <- as.factor(contraception$w.relig)
contraception$w.work <- as.factor(contraception$w.work)
contraception$h.occ <- as.factor(contraception$h.occ)
contraception$sol.index <- as.factor(contraception$sol.index)
contraception$media <- as.factor(contraception$media)
contraception$contra_use <- as.factor(contraception$contra_use)
# check data structure
str(contraception)

## 'data.frame':    1473 obs. of  11 variables:
##  $ w.age     : int  24 45 43 42 36 19 38 21 27 45 ...
```

```
##  $ w.edu     : Factor w/ 4 levels "1","2","3","4": 2 1 2 3 3 4 2 3 2 1 ...
##  $ h.edu     : Factor w/ 4 levels "1","2","3","4": 3 3 3 2 3 4 3 3 3 1 ...
##  $ children  : int  3 10 7 9 8 0 6 1 3 8 ...
##  $ w.relig   : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
##  $ w.work    : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 1 2 2 ...
##  $ h.occ     : Factor w/ 4 levels "1","2","3","4": 2 3 3 3 3 3 3 3 3 2 ...
##  $ sol.index : Factor w/ 4 levels "1","2","3","4": 3 4 4 3 2 3 2 2 4 2 ...
##  $ media     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 2 ...
##  $ contra    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ contra_use: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Now that you've recoded your outcome variable, you can now move on to the analysis!

Conduct a cross-validated ridge regression. Use your recoded contraception use variable as the outcome and all of the other variables (except contra) in the data set as the predictors. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated ridge regression. Be sure to use the set.seed() provided to make your analysis reproducible.

```
# drop contra predictor
contra.dat <- contraception[,-10]
# create the matrix of explanatory variables
X <- model.matrix(contra_use ~., data=contra.dat)
# drop the intercept column
X <- X[,-1]
# create the vector of outcomes
Y <- contra.dat$contra_use
# set seed for reproducability
set.seed(1000)
# create ridge regression model (alpha = 0, family = binomial)
cvfit.contra.ridge <- cv.glmnet(x=X, y=Y,alpha=0, family = 'binomial')
```

Next, display the value for lambda.min and the coefficients associated with it. Make sure these are visible in your knitted document.

```
# print the smallest lambda for the fitted model
cvfit.contra.ridge$lambda.min
```

```
## [1] 0.009928821
```

```
# print coefficients associated with smallest lambda
coef(cvfit.contra.ridge, s = "lambda.min")
```

```
## 18 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.73693374
## w.age       -0.07268161
## w.edu2       0.09666223
## w.edu3       0.50509918
```

```
## w.edu4       1.18617804
## h.edu2       0.16040255
## h.edu3       0.31845306
## h.edu4       0.20590260
## children     0.29742316
## w.relig1    -0.41096096
## w.work1      0.13237218
## h.occ2      -0.18800188
## h.occ3       0.04858944
## h.occ4       0.35949244
## sol.index2   0.24600895
## sol.index3   0.40062334
## sol.index4   0.63194424
## media1      -0.60732976
```

Finally, display the value for lambda.1se and the coefficients associated with it. Again, make sure these are visible in your knitted document.

```
# print the smallest lambda for the fitted model
cvfit.contra.ridge$lambda.1se

## [1] 0.04399085

# print coefficients associated with smallest lambda
coef(cvfit.contra.ridge, s = "lambda.1se")

## 18 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.814050662
## w.age       -0.050989316
## w.edu2      -0.100077950
## w.edu3       0.229780297
## w.edu4       0.767744800
## h.edu2      -0.007409206
## h.edu3       0.177617372
## h.edu4       0.152341647
## children     0.209873633
## w.relig1    -0.321093486
## w.work1      0.129024363
## h.occ2      -0.160770912
## h.occ3       0.030110021
## h.occ4       0.190076691
## sol.index2   0.058094804
## sol.index3   0.197325368
## sol.index4   0.390180390
## media1      -0.620230457
```

Question 5

For this question, you will use the same outcome (your binary contraception use variable) and the same predictors in the as in the last question, but you will instead conduct a cross-validated lasso regression. Do NOT split your data into training and test sets for this question; conduct the analysis on the whole data set.

First, conduct the cross-validated lasso regression and display the output the function produces. Be sure to use the set.seed() provided to make your analysis reproducible.

```
# drop contra predictor
contra.dat <- contraception[,-10]
# create the matrix of explanatory variables
X <- model.matrix(contra_use ~., data=contra.dat)
# drop the intercept column
X <- X[,-1]
# create the vector of outcomes
Y <- contra.dat$contra_use
# set seed for reproducibility
set.seed(1000)
# create lasso regression model (alpha = 1, family = binomial)
cvfit.contra.lasso <- cv.glmnet(x=X, y=Y,alpha=1, family = 'binomial')
```

Next, display the value for lambda.min and the coefficients associated with it. Make sure these are visible in your knitted document.

```
# print the smallest lambda for the fitted model
cvfit.contra.lasso$lambda.min

## [1] 0.002637182

# print coefficients associated with smallest lambda
coef(cvfit.contra.lasso, s = "lambda.min")

## 18 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  1.07263268
## w.age       -0.07967850
## w.edu2       0.12291182
## w.edu3       0.57018206
## w.edu4       1.30106082
## h.edu2       .
## h.edu3       0.13427083
## h.edu4       .
## children     0.32297017
## w.relig1    -0.41079952
## w.work1      0.10223272
## h.occ2      -0.19717023
## h.occ3       0.01795036
```

```
## h.occ4       0.27443645
## sol.index2   0.18842304
## sol.index3   0.36016519
## sol.index4   0.60771165
## media1      -0.58852890
```

Finally, display the value for lambda.1se and the coefficients associated with it. Again, make sure these are visible in your knitted document.

```
# print the smallest lambda for the fitted model
cvfit.contra.lasso$lambda.1se

## [1] 0.0186048

# print coefficients associated with smallest lambda
coef(cvfit.contra.lasso, s = "lambda.1se")

## 18 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept)  0.93786785
## w.age       -0.04959185
## w.edu2       .
## w.edu3       0.19166091
## w.edu4       0.86230530
## h.edu2       .
## h.edu3       .
## h.edu4       .
## children     0.20810536
## w.relig1    -0.14569925
## w.work1      .
## h.occ2      -0.03005510
## h.occ3       .
## h.occ4       .
## sol.index2   .
## sol.index3   .
## sol.index4   0.18865244
## media1      -0.53339719
```

Question 6 - 5 points

Answer the two questions below about the results of your cross-validated lasso models for the contraception data. 1) Among the set of coefficients associated with lambda.min in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here:

The "h.edu2" and "h.edu4" variables were set to zero for lambda.min in the lasso regression model.

2) Among the set of coefficients associated with lambda.1se in the cross-validated lasso regression, which predictors were set to zero? Please list them.

Your answer here:

The "w.edu2", "h.edu2","h.edu3", "h.edu4", "w.work1", "h.occ3", "h.occ4", "Sol.index2", "sol.index3" variables were set to zero for lambda.lse in the lasso regression model.

3) Which of these - lambda.min or lambda.1se - had more coefficients set to zero?

Your answer here:

lambda.1se had more coefficients set to zero for the cvfit.contra.lasso regression model.

Extra credit opportunity!

Now that you're done, I want to check a few things about your R session. Please be sure that you have the glmnet package loaded before completing this step.

```
# Install and load the rstudioapi package

# install.packages("rstudioapi")
library(rstudioapi)
```

Run the following code chunks and make sure the output displays in your knitted document. If your knitted document contains these three things, I will award you one extra credit point for this assignment (max 50 points for this assignment).

Version of R

```
R.Version() # base function that outputs current R version
```

```
## $platform
## [1] "x86_64-apple-darwin17.0"
##
## $arch
## [1] "x86_64"
##
## $os
## [1] "darwin17.0"
##
## $system
## [1] "x86_64, darwin17.0"
##
## $status
## [1] ""
##
## $major
## [1] "4"
##
## $minor
## [1] "1.1"
##
## $year
## [1] "2021"
##
## $month
## [1] "08"
##
## $day
## [1] "10"
##
## $`svn rev`
## [1] "80725"
##
## $language
## [1] "R"
##
## $version.string
## [1] "R version 4.1.1 (2021-08-10)"
##
## $nickname
## [1] "Kick Things"
```

Version of RStudio

```
versionInfo() # rstudioapi function that outputs current RStudio version
```

```
## $citation
##
```

```
## To cite RStudio in publications use:
##
##   RStudio Team (2021). RStudio: Integrated Development Environment for
##   R. RStudio, PBC, Boston, MA URL http://www.rstudio.com/.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {RStudio: Integrated Development Environment for R},
##     author = {{RStudio Team}},
##     organization = {RStudio, PBC},
##     address = {Boston, MA},
##     year = {2021},
##     url = {http://www.rstudio.com/},
##   }
##
##
## $mode
## [1] "desktop"
##
## $version
## [1] '1.4.1717'
##
## $release_name
## [1] "Juliet Rose"
```

Version of glmnet

```
packageVersion("glmnet") # base function that outputs current package version
```

```
## [1] '4.1.2'
```