1. The data frame HSWRESTLER, (from package PASWR2) contains information on nine variables for a group of 78 high school wrestlers that was collected by the human performance lab at Appalachian State University. It is of interest to predict wrestler's hydrostatic fat (hwfat) using predictors age,ht,wt,abs,triceps and subscap. Split the data set into a training set and a test set (50%). Use set.seed(1) each time you need to use sample() or cv.glmnet() functions.

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

   (b) Fit a ridge regression model on the training set, with λ chosen by cross-validation (use 15-fold cross validation). Report the test mspe.

   (c) Fit a lasso model on the training set, with λ chosen by cross-validation (use 15-fold CV). Report the test error obtained.

```
# hw3q1.r

library(PASWR2)
library(glmnet)

# a) training and test sets

d0 = HSWRESTLER[,1:7]
n = nrow(d0)
p = ncol(d0)-1

set.seed(1)
train=sample(1:n, n/2)
dtrain = d0[train,]
dtest = d0[-train,]

# (b) linear model
#============================================================

m1 = lm( hwfat ~ ., dtrain )
Y_hat = predict(m1,dtest )

mspe = mean((dtest$hwfat - Y_hat)^2)
mspe
# 14.0963
# (c) ridge regression
#===============================================================

Y = dtrain$hwfat
MM = model.matrix(hwfat ~ ., data=dtrain)[,-1]

newx=model.matrix(hwfat~.,dtest)[,-1]

set.seed(1)
cv.out = cv.glmnet(MM,Y, alpha=0,nfolds=15)
bestlam = cv.out$lambda.min
bestlam
# 0.994199

ridge.mod = glmnet(MM,Y,alpha=0)
Y_hat = predict(ridge.mod,s=bestlam,newx)

mspe2 =mean((dtest$hwfat-Y_hat)^2)
mspe2
# 14.5059

# refit with full data set

Y = d0$hwfat
MM = model.matrix(hwfat~.,data=d0)[,-1]

out=glmnet(MM,Y,alpha=0)
options(scipen=999)     # disable scientific notation
options(digits=4)
ridge.coef=predict(out,type="coefficients",s=bestlam) [1:p,]
ridge.coef
# (Intercept)         age          ht          wt         abs     triceps     subscap    ta
#   15.408559   -0.430032   -0.128271    0.006445    0.136425    0.232287    0.128201    0.19
# (d) lasso regression
#================================================================

Y = dtrain$hwfat
MM = model.matrix(hwfat ~ ., data=dtrain)[,-1]

set.seed(1)
cv.out = cv.glmnet(MM,Y,alpha=1,nfolds=15)
bestlam = cv.out$lambda.min
bestlam
# 0.04948

lasso.mod = glmnet(MM,Y,alpha=1)
Y_hat = predict(lasso.mod,s=bestlam,newx)

mspe3 =mean((dtest$hwfat-Y_hat)^2)
mspe3
# 13.77

# refit with full data set

Y = d0$hwfat
MM = model.matrix(hwfat~.,data=d0)[,-1]

out=glmnet(MM,Y,alpha=1)
options(digits=6)
lasso.coef=predict(out,type="coefficients",s=bestlam) [1:p,]
lasso.coef
#(Intercept)         age          ht          wt         abs     triceps
#   15.214113   -0.382793   -0.104636    0.000000    0.341984    0.397083

lasso.coef[lasso.coef!=0]
#(Intercept)         age          ht         abs     triceps
#   15.214113   -0.382793   -0.104636    0.341984    0.397083
```

2. Generate simulated data, and will then use this data to perform best subset selection. Generate values of a predictor X of length $n = 100$, using `X = rnorm(n)`. Generate values of a noise vector ϵ of length $n = 100$ using ϵ `= 0.1*rnorm(n)` Generate a response vector Y of length n = 100 using

$$Y = 1 - 0.1X + 0.05X^2 + 0.75X^3 + \epsilon$$

   (a) Use `regsubsets()` to choose the best model containing the predictors $X, X^2, ..., X^{10}$. What is the best model obtained according to adjusted R^2?

   (b) Fit a lasso model to the simulated data, again using $X, X^2, ..., X^{10}$ as predictors. Use 10-fold cross-validation to select the optimal value of λ. Report the test error.

```
set.seed(1)
n = 100
x = rnorm(n)
epsilon = 0.1 * rnorm(n)

beta_0 = 1.0
beta_1 = -0.1
beta_2 = +0.05
beta_3 = 0.75
y = beta_0+beta_1*x+beta_2*x^2+beta_3*x^3+epsilon
d0 = data.frame(y,x,x2=x^2,x3=x^3,x4=x^4,x5=x^5,x6=x^6,x7=x^7,x8=x^8,x9=x^9,x10=x^10)

# split training/test sets

set.seed(1)
train=sample(1:n, n/2)
test = (-train)
dtrain = d0[train,]
dtest = d0[test,]

# a) regsubsets
library(leaps)

models = regsubsets(y~.,d0,nvmax=11)
summary(models)
# Selection Algorithm: exhaustive
#          x   x2  x3  x4  x5  x6  x7  x8  x9  x10
# 1  ( 1 ) " " " " "*" " " " " " " " " " " " " " "
# 2  ( 1 ) " " " " "*" " " "*" " " " " " " " " " "
# 3  ( 1 ) "*" " " "*" "*" " " " " " " " " " " " "
# 4  ( 1 ) "*" "*" "*" " " "*" " " " " " " " " " "
# 5  ( 1 ) "*" "*" "*" " " "*" "*" " " " " " " " "
# 6  ( 1 ) "*" "*" "*" " " " " " " "*" "*" "*" " "
# 7  ( 1 ) "*" " " "*" "*" " " "*" "*" "*" " " "*"
# 8  ( 1 ) "*" "*" "*" "*" " " "*" " " "*" "*" "*"
# 9  ( 1 ) "*" "*" "*" "*" "*" "*" " " "*" "*" "*"
# 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"

summary(models)$adjr2
```

```
# 0.996226 0.997172 0.997571 0.997600 0.997586 0.997565 0.997556 0.997534 0.997507 0.997484
a=summary(models)$adjr2
which.max(a)   # 4
# best model with predictors x,x2,x3,x5

#b) lasso
#======================================================================
library(glmnet)

MM = model.matrix(y˜.,dtrain)
ytrain = dtrain$y

# best lambda from train set
set.seed(1)
cv.out = cv.glmnet(MM,ytrain,alpha=1)
bestlam = cv.out$lambda.min
bestlam
# 0.0255913

lasso.mod = glmnet(MM,ytrain,alpha=1)

# predict test set
newx = model.matrix(y˜.,dtest) #[,-1]
yhat = predict(lasso.mod,s=bestlam,newx)   # error

ytest = dtest$y
mspe  = mean((ytest-yhat)^2)
mspe
# 0.0103474

# refit
ytrain = d0$y
MMfull = model.matrix(y˜.,d0)
out=glmnet(MMfull,d0$y,alpha=1)

lasso.coef=predict(out,type="coefficients",s=bestlam)[1:12,]
lasso.coef
lasso.coef[lasso.coef!=0]
# (Intercept)          x3          x5
#  1.03708472  0.64348867  0.01368301
```

3. A real estate appraiser is interested in predicting residential home prices in a mid-western city as a function of various features. For that purpose a regression model is to be constructed from a sample of 522 houses. Use the `homes.xls` data set from blackboard. Consider the predictors
   x_1: lot size (square feet), x_2: area (square feet), x_3: number of bedrooms,
   x_4: number of bathrooms, x_5: year of construction, x_6: garage size (number of cars).
   Split the data set into a training set and a test set (50%).

   (a) Fit a ridge regression model on the training set, with λ chosen by cross-validation (use 15-fold cross validation). Report the test mspe.

   (b) Fit a lasso model on the training set, with λ chosen by cross-validation (use 15-fold CV). Report the test error obtained.

   (c) Predict the price when all predictors are equal to their median values using both the ridge regression and the lasso models.

```
# hw3q3.r

library(glmnet)
setwd("C:/Users/USC Guest/Downloads2")
d0=read.csv("homes.csv",header=T)
d1 = subset(d0,select=c(price,lotsize,area,beds,baths,year,garage))


# a) training and test sets

set.seed(1)
n = nrow(d1)
p = ncol(d1)-1

train=sample(1:n, n/2)
test = (-train)
dtrain = d1[train,]
dtest = d1[test,]

# linear model (not required)
#===============================================================

m1 = lm( price ~ ., dtrain )
Y_hat = predict(m1,dtest )

mspe = mean((dtest$price - Y_hat)^2)
mspe
# 5048263624
# (a) ridge regression
#================================================================

Y = dtrain$price
newx=model.matrix(price~.,dtest)
newx=model.matrix(price~.,dtest)[,-1]
MM = model.matrix(price ~ ., data=dtrain)[,-1]

head(dtrain)
head(MM)

set.seed(1)
cv.out = cv.glmnet(MM,Y, alpha=0,nfolds=15)
bestlam = cv.out$lambda.min
bestlam
#  12660.1

ridge.mod = glmnet(MM,Y,alpha=0)

# model from training set
ridge.coef=predict(ridge.mod,type="coefficients",s=bestlam) [1:p,]
ridge.coef
#    (Intercept)        lotsize           area           beds          baths           year
# -4157738.62649        1.25728       96.29050    -3482.30449     9980.27572     2096.76099

# Prediction at the median with training data set model
#-----------------------------------------------------------
apply(d1,2,median)
#  price lotsize    area    beds   baths    year  garage
# 229900   22200    2061       3       3    1966       2

# order matters
newval2= data.frame(lotsize=22200,area=2061,beds=3,baths=3,year=1966,garage=2)
newval2 = as.matrix(newval2)
newval2
#      lotsize area beds baths year garage
# [1,]   22200 2061    3     3 1966      2
Y_hat = predict(ridge.mod,s=bestlam,newval2)
Y_hat
#           1
# [1,] 256792

# mspe
Y_hat = predict(ridge.mod,s=bestlam,newx)
mspe2 =mean((dtest$price-Y_hat)^2)
mspe2

# 5280252885

# refit
```

```
Y = d1$price
MM = model.matrix(price˜.,data=d1)[,-1]

out=glmnet(MM,Y,alpha=0)
options(scipen=999)     # disable scientific notation
options(digits=4)
ridge.coef=predict(out,type="coefficients",s=bestlam) [1:p,]
ridge.coef
# (Intercept)      lotsize         area         beds        baths         year
#-3290230.723        1.428      104.289    -8129.583    14828.204     1642.448

# Prediction at the median with refitted model
#----------------------------------------------------------
Y_hat = predict(out,s=bestlam,newval2)
Y_hat
#          1
#[1,] 258975
```

```
# (b) lasso regression
#===============================================================

Y = dtrain$price
MM = model.matrix(price~.,data=dtrain)[,-1]

set.seed(1)
cv.out = cv.glmnet(MM,Y,alpha=1,nfolds=15)
bestlam = cv.out$lambda.min
bestlam   # 574.1

lasso.mod = glmnet(MM,Y,alpha=1)

# Prediction at the median with training data set model
Y_hat = predict(lasso.mod,s=bestlam,newval2)
Y_hat
#          1
# [1,] 251434

# mspe
Y_hat = predict(lasso.mod,s=bestlam,newx)
mspe3 =mean((dtest$price-Y_hat)^2)
mspe3
# 5084835702

# refit with full data set

Y = d1$price
MM = model.matrix(price~.,data=d1)[,-1]

out=glmnet(MM,Y,alpha=1)
options(digits=6)
lasso.coef=predict(out,type="coefficients",s=bestlam) [1:p,]
lasso.coef

#    (Intercept)        lotsize           area           beds          baths           year
# -3542559.21938        1.52412      125.29191   -12118.07170     7565.80255     1766.82300

# No coefficients equal to zero

# Prediction at the median with refitted model
#-----------------------------------------------------------
Y_hat = predict(out,s=bestlam,newval2)
Y_hat
#          1
# [1,] 253919
```