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. The variables are

- `age` (in years)
- `ht` (height in inches)
- `wt` (weight in pounds)
- `abs` (abdominal skinfold measure)
- `triceps` (tricep skinfold measure)
- `subscap` (subscapular skinfold measure)
- `hwfat` (hydrostatic determination of fat)
- `tanfat` (Tanita determination of fat)
- `skfat` (skinfold determination of fat)

It is of interest to predict wrestler's hydrostatic fat (`hwfat`) using predictors `age,ht,wt,abs,triceps` and `subscap`. Remove observations 22, 27, 32,35, and 60, which may have been poorly measured.
Use 5-fold cross validation to find the best regression model.

To find the best set of predictors with the highest predictive performance, we will

- Select best set of predictors using `regsubsets()`
- Select the number of predictors using cross validation
- Build the model and find coefficient estimates using the full data set

```
# hwrestler.r

# K-fold cross validation

library(PASWR2)    # dataset
library(leaps)     # regsubsets()

d0=HSWRESTLER
ig = c(22, 27, 32, 35, 60)
d1=d0[-ig,1:7]          # main dataset
n <- nrow(d1)           # [1] 73

k = 5                  # 5 folds
set.seed(5)

# create folds
x = rep(1:5,each=14)
length(x)
# [1] 70
x = sample(x)
# [1] 2 4 5 2 1 4 3 4 5 5 2 3 2 3 5 1 2 5 5 4 4 3 1 4 5 2 5 3 1 3 2 1 4 1 1
#[36] 3 4 4 1 5 2 5 2 4 3 1 5 3 3 2 1 3 5 4 2 5 1 5 3 2 4 2 1 3 2 1 3 4 1 4
x2 = sample(1:5,3)
# [1] 5 1 3
folds = c(x,x2)
table(folds)
# 1  2  3  4  5
#15 14 15 14 15
plot(folds)

mspe <- matrix(0, k, 6)          # 5-by-6 matrix
#     [,1] [,2] [,3] [,4] [,5] [,6]
#[1,]    0    0    0    0    0    0
#[2,]    0    0    0    0    0    0
#[3,]    0    0    0    0    0    0
#[4,]    0    0    0    0    0    0
#[5,]    0    0    0    0    0    0

dim(mspe)    # [1] 5 6
# mspe[j,i] = MSPE of best model with i predictors ignoring jth fold
# fold 1
y = d1[folds == 1,]
#   age    ht    wt abs triceps subscap hwfat
#5   17 69.50 299.2  54    42.0      37 41.89
#16  17 71.50 181.6   9    10.0      10  8.27
#24  15 68.25 133.6  11    10.5       9  9.49
#31  15 68.75 201.4  37    27.0      31 31.71
#36  15 63.25 152.6  21    13.0       9 17.83
#38  14 67.25 124.2  10    10.0       8 13.87
#39  16 69.00 209.8  41    35.0      36 33.53
#43  14 67.00 128.6   9    11.0       9  7.69
#50  15 68.50 224.0  41    30.0      34 27.01
#55  18 69.00 146.4   9    10.0       8 10.40
#62  17 68.00 155.4   8     7.0       8 11.79
#68  18 67.00 161.4   7     6.0       7  9.81
#71  17 69.00 174.2  10     7.0       8  6.33
#74  16 69.00 140.2   7     6.0       6  6.86
#78  15 66.00 258.6  45    37.0      43 33.75

y = d1$hwfat[folds == 1]
# 41.89 13.08  7.97 31.71 17.83 13.87 33.53  9.91  7.17 11.40 11.27 10.26 33.75

# function predict.regsubsets()

predict.regsubsets <- function(object, newdata, id, ...)
{
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi = coef(object, id = id)
  xvars <- names(coefi)
  mat[, xvars]%*%coefi
}


for(j in 1:k)   # loop over all folds
{
  y = d1$hwfat[folds == j]     # y-values in j-th fold
  d2 = d1[folds != j,]         # training set ignores j-th fold
  cvmodels <- regsubsets(hwfat ˜.,d2)
  for(i in 1:6)                # i number of predictors in model
  {
    newdata = d1[folds ==j,]       # test set
    yhat <- predict.regsubsets(cvmodels,newdata,id=i)  # predict jth fold (vector)
    mspe[j, i] <- mean((y - yhat)ˆ2)
  }
}
```

```
mspe
#          [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
#[1,]  9.418801  6.018222  6.544758  7.201831  7.284653  7.468159
#[2,]  8.982257  6.694449  6.561723  7.666092  7.777638  7.706906
#[3,]  6.942410  4.751430  6.340462  6.686701  6.881903  6.871311
#[4,]  7.091732  7.775369  7.631415  7.538602  7.692280  7.673438
#[5,] 12.534920 10.070260 10.042608 10.732882 10.862931 10.785316

# rows are folds, cols are number of predictors in model
# 6.018222 is mse when predicting 1st fold
# using best model with 2 predictors

CVk <- apply(mspe, 2, mean)
#        1        2        3        4        5        6
# 8.994024 7.061946 7.424193 7.965222 8.099881 8.101026

aux = which.min(CVk)   # 2
models <- regsubsets(hwfat~.,d1)
coef(models,aux)
# (Intercept)         abs     triceps
#   1.9119410   0.3929936   0.4211225

# in general select the folds as follows
m = floor(n/k)
x = rep(1:k,each=m)
x = sample(x)
m2 = n-length(x)
x2 = sample(1:k,m2)
folds = c(x,x2)
table(folds)
# folds
#  1  2  3  4  5
# 15 15 15 14 14
# Leave-One-Out cross validation    (n-fold)
#==============================================================

k = n       # folds are the rows
mspe = matrix(0, k, 6)
dim(mspe)               # [1] 73  6
head(mspe)
#     [,1] [,2] [,3] [,4] [,5] [,6]
#[1,]    0    0    0    0    0    0
#[2,]    0    0    0    0    0    0
#[3,]    0    0    0    0    0    0
#[4,]    0    0    0    0    0    0
#[5,]    0    0    0    0    0    0
#[6,]    0    0    0    0    0    0

# mspe[j,i] = MSPE of best model with i predictors ignoring jth row

for(j in 1:k)         # loop over all rows
{
  y = d1$hwfat[j]     # jth y-value
  d2 = d1[-j,]        # training set ignores jth row
  models <- regsubsets(hwfat ~.,d2)
  for(i in 1:6)
  {
    newdata = d1[j,]  # test set is the jth row
    yhat <- predict.regsubsets(models,newdata,id=i) # predict jth row
    mspe[j, i] <- mean((y - yhat)^2)
  }
}

cv1 <- apply(mspe, 2, mean)
#        1        2        3        4        5        6
# 8.686337 7.361655 7.856348 8.314736 8.291063 8.223795

aux = which.min(cv1)     # [1] 2

# LOOCV selects same model as 5-fold CV

regfit.best <- regsubsets(hwfat~.,d1)
coef(regfit.best,aux)
# (Intercept)         abs     triceps
#   1.9119410   0.3929936   0.4211225
```