# STAT408_HW6

2022-12-02

## Question 1

(15 points) We're going to use the mtcars dataset that can be found in the R package "datasets". Import the dataset by running "library(datasets); data(mtcars)".

```
library(datasets)
data(mtcars)
```

### part (a)

Fit a logistic regression model with the variable am as the response and mpg and hp as predictors. What are the estimated regression coefficients from this model? How do we interpret them here?

```
logistic.model <- glm(am~mpg + hp, data=mtcars, family='binomial')
summary(logistic.model)
```

```
##
## Call:
## glm(formula = am ~ mpg + hp, family = "binomial", data = mtcars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.41460  -0.42809  -0.07021   0.16041   1.66500
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -33.60517   15.07672  -2.229   0.0258 *
## mpg           1.25961    0.56747   2.220   0.0264 *
## hp            0.05504    0.02692   2.045   0.0409 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 31  degrees of freedom
## Residual deviance: 19.233  on 29  degrees of freedom
## AIC: 25.233
##
## Number of Fisher Scoring iterations: 7
```

```
exp(1.25961)
```

```
## [1] 3.524047
```

```
exp(0.05504)
```

```
## [1] 1.056583
```

The coefficient of mpg is 1.25961, meaning that one unit increase in miles per gallon will increase the odds of a car having a manual transmission by a factor of exp(1.25961), or 3.524047. Similarly, a one unit increase in gross horsepower will increase the odds of a car having a manual transmission by a factor of 1.056583.

## part (b)

What is the predicted probability that a car is automatic if it has hp = 180 and mpg = 20?

P(Y = 1) = 1/(1+exp(-(-33.60517 + 1.25961*mpg + 0.05504*hp)))

```
1 - (1/(1+exp(-(-33.60517  + 1.25961*20 + 0.05504*180))))
```

```
## [1] 0.1832877
```

## part (c)

Randomly split the data into a 80% train set and a 20% test set. Fit a logistic model on the training set and predict on the test set. What is the prediction accuracy of transmission type on the test set? (Hint: if the probability of being 1 is greater than 0.5 then set the transmission type equal to 1, otherwise, set it to 0)

```
# random split the data into 80% training and 20% test
set.seed(2022)
index.train <- sample(1:dim(mtcars)[1], 0.8 * dim(mtcars)[1])
data.train <- mtcars[index.train,]
data.test <- mtcars[-index.train,]

# fit a logistic regression on training set
logistic.model2 <- glm(am~mpg+hp, data=data.train, family='binomial')

# predict on the test test, obtain the predicted P(Y=1)
p.pred <- predict(logistic.model2, data.test, type='response')

# transform to binary response
y.pred <- ifelse(p.pred>=0.5, 1, 0)

# calculate classification accuracy
y.truth <- data.test$am
acc.test <- mean(y.pred==y.truth)
acc.test
```

```
## [1] 0.8571429
```

## part (d)

Show the confusion matrix. Calculate the true positive rate, true negative rate, and precision.

```
# confusion matrix
table(y.pred, y.truth)
```

```
##        y.truth
## y.pred 0 1
##      0 3 0
##      1 1 3
```

```
# true positive
TP <- intersect(which(y.truth==1), which(y.pred==1))

# true negative
TN <- intersect(which(y.truth==0), which(y.pred==0))

# all positives
AP <- which(data.test$AHD==1)

# all negatives
AN <- which(data.test$AHD==0)

# predicted positives
PP <- which(y.pred==1)

# true postive rate
TPR <- length(TP) / length(AP)
TPR
```

```
## [1] Inf
```

```
# true negative rate
TNR <- length(TN) / length(AN)
TNR
```

```
## [1] Inf
```

```
# precision
prec <- length(TP) / length(PP)
prec
```

```
## [1] 0.75
```

# Question 2

(15 points) Use seatpos data to conduct the following analysis. Make sure you understand the meaning of each variable in this dataset.

```r
seatpos <- read.csv("seatpos.csv")
```

## part (a)

Use hipcenter as response and all other variables as predictors to fit a linear model. How you interpret this model? What is the issue of this model?

```r
lmodel <- lm(hipcenter~., data = seatpos)
summary(lmodel)
```

```
## 
## Call:
## lm(formula = hipcenter ~ ., data = seatpos)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -73.827 -22.833  -3.678  25.017  62.337 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 436.43213  166.57162   2.620   0.0138 *
## Age           0.77572    0.57033   1.360   0.1843  
## Weight        0.02631    0.33097   0.080   0.9372  
## HtShoes      -2.69241    9.75304  -0.276   0.7845  
## Ht            0.60134   10.12987   0.059   0.9531  
## Seated        0.53375    3.76189   0.142   0.8882  
## Arm          -1.32807    3.90020  -0.341   0.7359  
## Thigh        -1.14312    2.66002  -0.430   0.6706  
## Leg          -6.43905    4.71386  -1.366   0.1824  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 37.72 on 29 degrees of freedom
## Multiple R-squared:  0.6866, Adjusted R-squared:  0.6001 
## F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```

This model has an adjust R^2 of 0.6001 and no significant predictors at the level alpha = 0.05, suggesting that it may not be a good estimate of hipcenter.

## part (b)

Use cor function to check the correlation of all predictors. What predictors are highly correlated? Is there any relation between correlations and model fitting in (a)?

```r
cor(seatpos)
```

```
##                 Age      Weight     HtShoes          Ht     Seated       Arm
## Age      1.00000000  0.08068523 -0.07929694 -0.09012812 -0.1702040 0.3595111
## Weight   0.08068523  1.00000000  0.82817733  0.82852568  0.7756271 0.6975524
## HtShoes -0.07929694  0.82817733  1.00000000  0.99814750  0.9296751 0.7519530
## Ht        -0.09012812  0.82852568  0.99814750  1.00000000  0.9282281  0.7521416
## Seated    -0.17020403  0.77562705  0.92967507  0.92822805  1.0000000  0.6251964
## Arm        0.35951115  0.69755240  0.75195305  0.75214156  0.6251964  1.0000000
## Thigh      0.09128584  0.57261442  0.72486225  0.73496041  0.6070907  0.6710985
## Leg       -0.04233121  0.78425706  0.90843341  0.90975238  0.8119143  0.7538140
## hipcenter  0.20517217 -0.64033298 -0.79659640 -0.79892742 -0.7312537 -0.5850950
##                 Thigh         Leg  hipcenter
## Age        0.09128584 -0.04233121  0.2051722
## Weight     0.57261442  0.78425706 -0.6403330
## HtShoes    0.72486225  0.90843341 -0.7965964
## Ht         0.73496041  0.90975238 -0.7989274
## Seated     0.60709067  0.81191429 -0.7312537
## Arm        0.67109849  0.75381405 -0.5850950
## Thigh      1.00000000  0.64954120 -0.5912015
## Leg        0.64954120  1.00000000 -0.7871685
## hipcenter -0.59120155 -0.78716850  1.0000000
```

Weight is highly correlated with HtShoes, Ht, Seated, and Leg, which are also highly correlated with one another in addition to Arm and Thigh. These are also the predictors that are most strongly correalted with hipcenter. Due to these correaltions among predictors, the model in part a may have issues with collinearity.

## part (c)

Conduct a PCA transformation on all predictors. How much variance the first two PCs have?

```r
prseatpos <- prcomp(seatpos, scale = TRUE)
prseatpos
```

```
## Standard deviations (1, .., p=9):
## [1] 2.51654318 1.13474718 0.68332350 0.56974718 0.46210478 0.43428041 0.36584961
## [8] 0.22415058 0.03983337
##
## Rotation (n x k) = (9 x 9):
##                    PC1         PC2         PC3         PC4        PC5
## Age       -0.007316739  0.85645574  0.13422484 -0.09684181 -0.1193027
## Weight     0.343388366  0.08478337  0.45543320  0.28831560 -0.6593121
## HtShoes    0.389707803 -0.06571526  0.05594288  0.12354222  0.1565492
## Ht         0.390356164 -0.07189876  0.03405422  0.12933629  0.1463482
## Seated     0.361050816 -0.17087676  0.20919354  0.30421213  0.2283072
## Arm        0.324279944  0.39146542 -0.02831240 -0.22180066  0.4516656
## Thigh      0.307395327  0.14881061 -0.83484311  0.32399483 -0.2331844
## Leg        0.370792856 -0.02732163  0.11278393 -0.22465798  0.2121057
## hipcenter -0.332093941  0.21088460  0.12660269  0.76047407  0.3868581
##                  PC6         PC7         PC8          PC9
## Age        0.44052876 -0.17358350 -0.01872887  0.015958550
## Weight    -0.36950858  0.10492211  0.04526183 -0.008037394
## HtShoes    0.16461329 -0.06163708 -0.53916368 -0.692325964
## Ht         0.13301797 -0.05849761 -0.51199170  0.721038164
## Seated     0.52177375  0.24002295  0.56757798  0.002946485
## Arm       -0.46793753  0.51315681  0.07241350 -0.007697974
## Thigh     -0.04125946 -0.06668735  0.14333133 -0.018960116
## Leg       -0.25660091 -0.76894409  0.31053151 -0.006089260
## hipcenter -0.25660625 -0.18701837 -0.02013588 -0.002223540
```

```
2.51654318**2
```

```
## [1] 6.33299
```

```
1.13474718**2
```

```
## [1] 1.287651
```

The first two principal components have variances 6.33299 and 1.287651, respectively

## part (d)

Show the linear combination coefficients in the first two PCs. Based on those coefficients, what interpretation can you make for the first two PCs?

```
round(prseatpos$rot[,1],2)
```

```
##       Age    Weight   HtShoes        Ht    Seated       Arm     Thigh       Leg 
##     -0.01      0.34      0.39      0.39      0.36      0.32      0.31      0.37 
## hipcenter 
##     -0.33
```

```
round(prseatpos$rot[,2],2)
```

```
##       Age    Weight   HtShoes        Ht    Seated       Arm     Thigh       Leg 
##      0.86      0.08     -0.07     -0.07     -0.17      0.39      0.15     -0.03 
## hipcenter 
##      0.21
```

The first two PCs have extremely different coefficients from one another. For instance, the first PC has a coefficient for Age that is -0.01 while the second coefficient is 0.86. The first PC contrasts age and hipcenter to the other variables about a person's body size and dimensions. Therefore, it compares a person's body size to the seat position in the car. The second PC contrasts age, weight, arm, thigh, and hipcenter to shoes, height, seated, and leg. Therefore, it contrasts the upper body measurement to the lower body measurements and measures relatively where the body carries its weight.

## part (e)

Conduct a PCA regression of hipcenter vs. first two PCs. How do you interpret this model result? Compare this model with the regular linear regression in (a) and give your insight.

```
lmodpcr <- lm(seatpos$hipcenter ~ prseatpos$x[,1:2])
summary(lmodpcr)
```

```
## 
## Call:
## lm(formula = seatpos$hipcenter ~ prseatpos$x[, 1:2])
## 
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -68.602 -19.784   1.804  22.001  44.734 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)           -164.885      4.917 -33.532  < 2e-16 ***
## prseatpos$x[, 1:2]PC1  -19.809      1.980 -10.003  8.4e-12 ***
## prseatpos$x[, 1:2]PC2   12.579      4.392   2.864  0.00702 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 30.31 on 35 degrees of freedom
## Multiple R-squared:  0.7557, Adjusted R-squared:  0.7417 
## F-statistic: 54.14 on 2 and 35 DF,  p-value: 1.943e-11
```

There is no collinearity because the first two PCs are orthogonal. The first PC measures a person's overall body size so hipcenter is going to decrease as size increases and a person's body is closer to the edges of the car. The second PC shows a positive association, so those who carry more weight in their legs are going to likely be thinner and thus have a greater distance between themselves and the edges of the car. This model appears to perform better than the one in part a as it has an adjusted R^2 of 0.7417 that is greater than the other model's adjusted R^2 of 0.6001.

## Question 3

(20 points) Take the fat data, and use the percentage of body fat, siri, as the response and the other variables, except brozek and density, as potential predictors. Remove every tenth observation from the data for use as the test set (1, 11, 21, . . . ). Use the remaining data as the training data building the following models, predict on the test set, and calculate the prediction RMSE on the test set.

```
fat <- read.csv("fat.csv")
```

```
# divide training and testing sets
test_i <- seq(1, 252, by = 10)
train <- fat[-test_i,]
test <- fat[test_i,]
```

### part (a)

Linear regression with all predictors.

```
# linear model
lm_fat <- lm(siri ~ age + weight + height + adipos + free + neck + chest + abdom + hip + thigh + knee +

# make predictions
y.pred <- predict(lm_fat, newdata = test)

# calculate test MSE
y.test <- test$siri
MSE.test <- mean((y.test - y.pred)^2)
```

```
# root MSE
RMSE.test <- sqrt(MSE.test)
RMSE.test
```

```
## [1] 1.946023
```

## part (b)

Linear regression with variables selected using backward AIC (hint: consider step function).

```
library(MASS)
stepAIC(lm_fat)
```

```
## Start:  AIC=186.31
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + hip + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - hip      1       0.0  447.4 184.32
## - neck     1       0.2  447.5 184.39
## - knee     1       0.2  447.5 184.39
## - age      1       0.3  447.6 184.45
## - wrist    1       1.4  448.7 185.02
## - height   1       1.6  449.0 185.13
## - ankle    1       2.9  450.2 185.76
## <none>                  447.3 186.31
## - biceps   1      10.7  458.1 189.66
## - abdom    1      16.1  463.5 192.31
## - forearm  1      18.5  465.8 193.47
## - chest    1      23.3  470.6 195.76
## - thigh    1      25.4  472.7 196.78
## - adipos   1      42.1  489.4 204.62
## - weight   1     576.0 1023.4 371.33
## - free     1    3385.3 3832.6 669.75
##
## Step:  AIC=184.32
## siri ~ age + weight + height + adipos + free + neck + chest +
##     abdom + thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - neck     1       0.2  447.5 182.39
## - knee     1       0.2  447.5 182.39
## - age      1       0.3  447.7 182.47
## - wrist    1       1.4  448.8 183.03
## - height   1       1.7  449.1 183.19
## - ankle    1       3.0  450.4 183.83
## <none>                  447.4 184.32
## - biceps   1      10.8  458.2 187.72
## - abdom    1      16.4  463.7 190.44
## - forearm  1      18.8  466.2 191.63
## - chest    1      24.8  472.1 194.50
## - thigh    1      27.1  474.4 195.59
## - adipos   1      43.6  491.0 203.34
## - weight   1     683.5 1130.8 391.90
## - free     1    3415.7 3863.0 669.54
##
## Step:  AIC=182.39
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     thigh + knee + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - knee     1       0.2  447.7 180.50
## - age      1       0.2  447.8 180.52
## - wrist    1       1.3  448.8 181.03
## - height   1       1.7  449.2 181.23
## - ankle    1       3.3  450.8 182.07
## <none>                  447.5 182.39
## - biceps   1      10.7  458.2 185.74
## - abdom    1      16.4  463.9 188.54
## - forearm  1      18.7  466.2 189.66
## - chest    1      24.7  472.2 192.55
## - thigh    1      26.9  474.4 193.60
## - adipos   1      45.7  493.2 202.38
## - weight   1     688.4 1135.9 390.90
## - free     1    3464.1 3911.6 670.37
##
## Step:  AIC=180.5
## siri ~ age + weight + height + adipos + free + chest + abdom +
##     thigh + ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - age      1       0.4  448.1 178.68
## - wrist    1       1.3  449.1 179.17
## - height   1       1.6  449.3 179.30
## - ankle    1       4.0  451.7 180.49
## <none>                  447.7 180.50
## - biceps   1      10.6  458.3 183.76
## - abdom    1      16.6  464.3 186.72
## - forearm  1      19.1  466.8 187.94
## - chest    1      24.7  472.4 190.62
## - thigh    1      32.1  479.8 194.15
## - adipos   1      48.9  496.6 201.94
## - weight   1     731.7 1179.4 397.41
## - free     1    3464.0 3911.7 668.37
##
## Step:  AIC=178.68
## siri ~ weight + height + adipos + free + chest + abdom + thigh +
##     ankle + biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - height   1       1.4  449.5 177.41
## - wrist    1       2.4  450.5 177.89
## - ankle    1       3.9  452.0 178.63
## <none>                  448.1 178.68
## - biceps   1      10.8  458.9 182.08
## - forearm  1      18.7  466.8 185.94
## - abdom    1      20.1  468.2 186.59
## - chest    1      25.1  473.2 188.99
## - thigh    1      33.4  481.5 192.95
## - adipos   1      49.4  497.5 200.31
## - weight   1     738.0 1186.1 396.68
## - free     1    3491.5 3939.6 667.97
##
## Step:  AIC=177.41
## siri ~ weight + adipos + free + chest + abdom + thigh + ankle +
##     biceps + forearm + wrist
##
##           Df Sum of Sq    RSS    AIC
## - wrist    1       2.6  452.1 176.72
## - ankle    1       3.9  453.5 177.38
## <none>                  449.5 177.41
## - biceps   1      11.2  460.7 180.98
## - forearm  1      19.0  468.6 184.79
## - abdom    1      20.4  469.9 185.44
## - chest    1      25.3  474.9 187.81
## - thigh    1      32.1  481.6 190.99
## - adipos   1      79.2  528.7 212.09
## - weight   1     847.9 1297.4 414.96
## - free     1    3492.9 3942.4 666.14
##
## Step:  AIC=176.72
## siri ~ weight + adipos + free + chest + abdom + thigh + ankle +
##     biceps + forearm
##
##           Df Sum of Sq    RSS    AIC
## <none>                  452.1 176.72
## - ankle    1       6.1  458.2 177.74
## - biceps   1      12.9  465.1 181.09
## - forearm  1      22.1  474.2 185.50
## - abdom    1      23.4  475.5 186.12
## - chest    1      25.3  477.4 187.01
## - thigh    1      29.5  481.7 189.02
## - adipos   1      79.2  531.3 211.20
## - weight   1     847.4 1299.6 413.33
## - free     1    3709.0 4161.1 676.34
```

```
##
## Call:
## lm(formula = siri ~ weight + adipos + free + chest + abdom +
##     thigh + ankle + biceps + forearm, data = train)
##
## Coefficients:
## (Intercept)       weight       adipos         free        chest        abdom
##     -2.9190       0.3925      -0.5277      -0.5698       0.1246       0.1179
##       thigh        ankle       biceps      forearm
##      0.1561       0.1475       0.1490       0.2146
```

```
# new linear model
lm_new <- lm(siri ~ weight + adipos + free + chest + abdom + thigh + ankle + biceps + forearm, data = t
# make predictions
y.pred <- predict(lm_new, newdata = test)

# calculate test MSE
y.test <- test$siri
MSE.test <- mean((y.test - y.pred)^2)

# root MSE
RMSE.test <- sqrt(MSE.test)
RMSE.test
```

```
## [1] 1.98911
```

## part (c)

Principal component regression. Use the first 7 PCs.

```
prfat <- prcomp(train, scale = TRUE)
lmodpcr <- lm(train$siri ~ prfat$x[,1:2])

# make predictions
y.pred <- predict(lmodpcr, newdata = test)
```

```
## Warning: 'newdata' had 26 rows but variables found have 226 rows
```

```
# calculate test MSE
y.test <- test$siri
MSE.test <- mean((y.test - y.pred)^2)
```

```
## Warning in y.test - y.pred: longer object length is not a multiple of shorter
## object length
```

```
# root MSE
RMSE.test <- sqrt(MSE.test)
RMSE.test
```

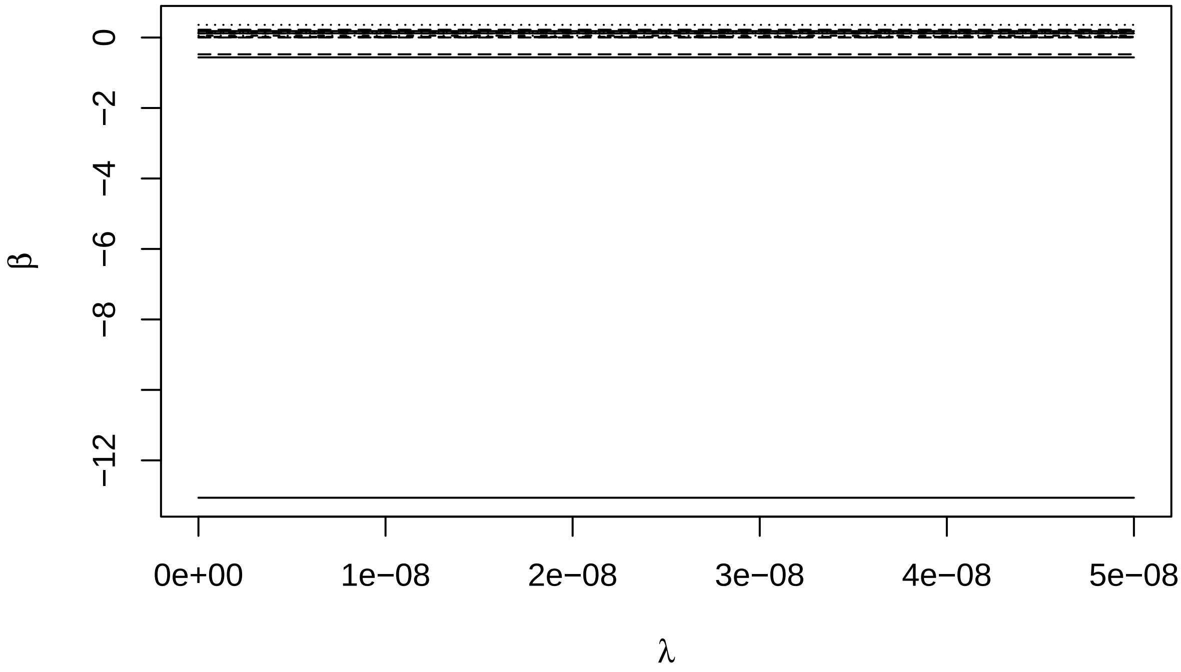
```
## [1] 11.85472
```

## part (d)

Ridge regression. Use cross-validation on the training set to select best penalty.

```
# ridge regression

require(MASS)
rgmod <- lm.ridge(siri ~ age + weight + height + adipos + free + neck + chest + abdom + hip + thigh + kn
matplot(rgmod$lambda, coef(rgmod), type="l", xlab=expression(lambda)
          ,ylab=expression(hat(beta)),col=1)
```



```
# regular linear model
modlm <- lm(siri ~ age + weight + height + adipos + free + neck + chest + abdom + hip + thigh + knee + a
yhat <- predict(modlm, test)
MSE <- mean((yhat - test$siri)^2)
MSE
```

```
## [1] 3.787006
```

```
# select best lamda by cross-validation
rgmod <- lm.ridge(siri ~ age + weight + height + adipos + free + neck + chest + abdom + hip + thigh + kn
rgmod$GCV
```

```
##   0.00e+00   2.50e-09   5.00e-09   7.50e-09   1.00e-08   1.25e-08   1.50e-08
## 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777
##   1.75e-08   2.00e-08   2.25e-08   2.50e-08   2.75e-08   3.00e-08   3.25e-08
## 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777
##   3.50e-08   3.75e-08   4.00e-08   4.25e-08   4.50e-08   4.75e-08   5.00e-08
## 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777 0.01004777
```

```
which.min(rgmod$GCV)
```

```
## 5.00e-08
##       21
```

```
# predict using best lamda
yhat.ridge <- cbind(1,as.matrix(test[,-1:-3])) %*% coef(rgmod)[21,]
MSE.ridge <- mean((yhat.ridge - test$siri)^2)
MSE.ridge
```

```
## [1] 3.787006
```

### part (e)

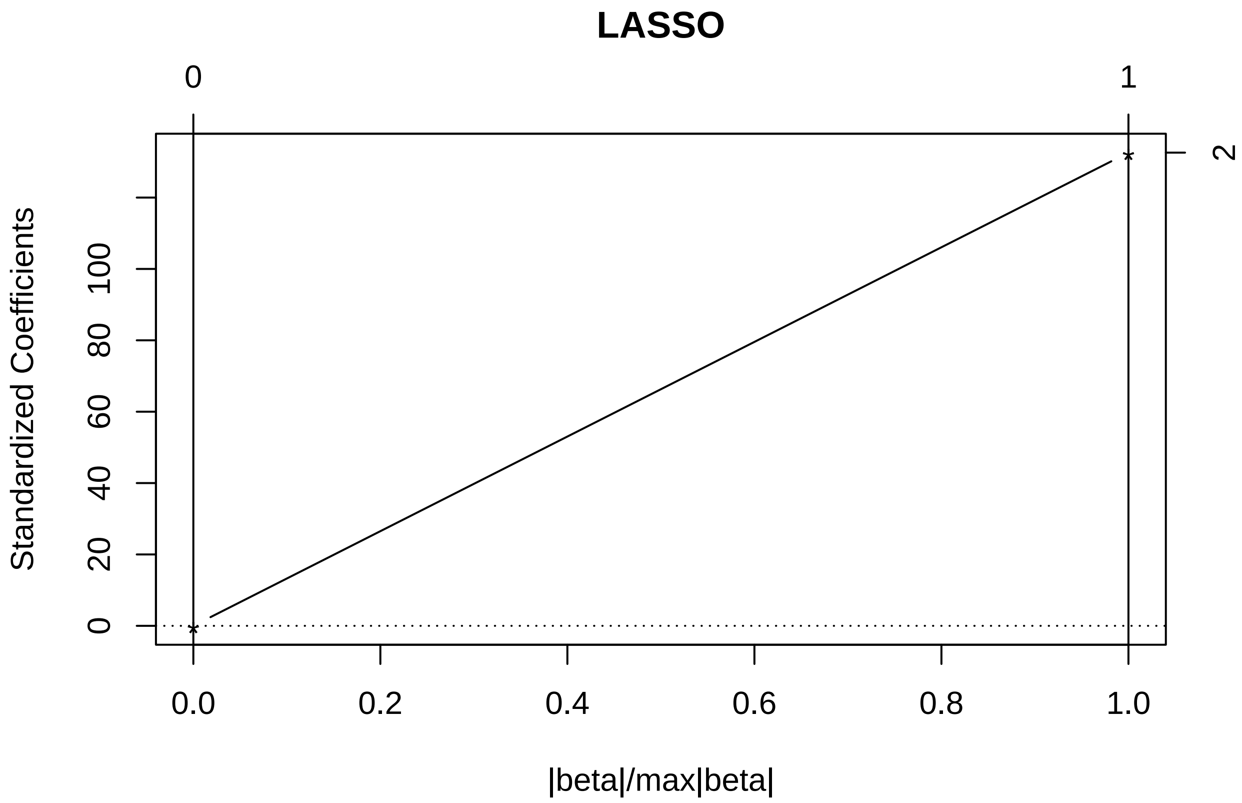
Lasso. Use cross-validation on the training set to select best penalty.

```
# Lasso
require(lars)
```

```
## Loading required package: lars
```

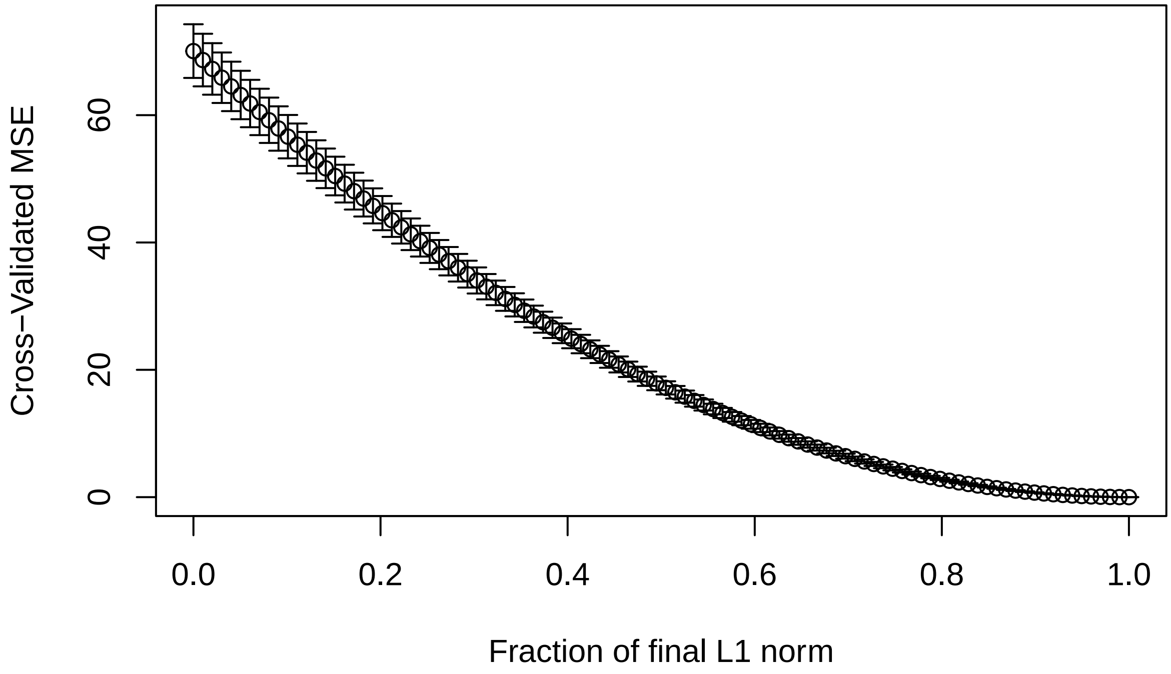
```
## Loaded lars 1.3
```

```
# lars function requires the matrix of predictors as its first argument,
# and the vector of response as its second argument
lmod <- lars(as.matrix(fat[,-4]),fat$siri)
plot(lmod)
```



```
# lasso for prediction
trainy <- train$siri
trainx <- as.matrix(train[,-101])
lassomod <- lars(trainx,trainy)

# select best penalty by cross-validation
set.seed(2022)
cvout <- cv.lars(trainx,trainy)
```



```
#cvout$index[which.min(cvout$cv)]

# predict using best penalty
testx <- as.matrix(test[,-101])
yhat <- predict(lassomod,testx,s=0.0101,mode="fraction")
MSE <- mean((yhat$fit - test$siri)^2)

# root MSE
RMSE.test <- sqrt(MSE)
RMSE.test
```

```
## [1] 8.351119
```

## part (f)

Compare all the RMSEs. Are you surprised on the model performance comparison? Give you speculation about why you see such result.

The smallest RMSE was 1.946023 for the linear model with all predictors, followed by the linear model chosen by stepAIC. Next was ridge regression, lasso, and then principal component regression. I am slightly surprise that the full linear model performed the best as this was the simplest model, but there may be clear simple relationships between percentage of body fat and variables such as height, weight, or different body measurements.