Exercise 2 (Conceptual: Lasso Regression)

Exercise 3 (p. 260): Discuss properties of lasso regression.

(a) As we increase s from 0, the training RSS will :

iv. Steadily decrease. As we increase s from 0, all β's increase from 0 to their least square estimates, and so the model is becoming more and more flexible which provokes a steady decrease in the training RSS.

(b) Repeat (a) for test RSS.

ii. Decrease initially, and then eventually start increasing in a U shape. When $s = 0$, all β's are 0, the model is extremely simple and has a high test RSS. As we increase s from 0, all β's increase from 0 to their least square estimates, and so the model is becoming more and more flexible which provokes at first a decrease in the test RSS. Eventually, as *beta*'s approach their full blown OLS values, they start overfitting to the training data, increasing test RSS.

(c) Repeat (a) for variance.

iii. Steadily increase. When $s = 0$, the model effectively predicts a constant and has almost no variance. As we increase s from 0, the models includes more β's and their values start increasing. At this point, the values of β's become highly dependent on training data, thus increasing the variance.

(d) Repeat (a) for (squared) bias.

iv. Steadily decrease. When $s = 0$, the model effectively predicts a constant and hence the prediction is far from actual value. As we increase s from 0, more β's become non-zero and thus the model continues to fit training data better. And thus, bias decreases.

(e) Repeat (a) for the irreducible error.

v. Remain constant. By definition, the irreducible error is independant of the model, and consequently independant of the value of s.

Exercise 3 (Applied: Model Comparison)

Exercise 9 (p. 263): Compare different methods (least squares, ridge, lasso, PCR, PLS) for College data. Perform additionally hybrid stepwise variable selection as well as elastic net.

(a) Split the data set into a training and a test set.

```
library(ISLR)
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix
## Loading required package: foreach
## Loaded glmnet 2.0-3
```

```
data(College)
set.seed(11)
train = sample(1:dim(College)[1], dim(College)[1] / 2)
test <- -train
College.train <- College[train, ]
College.test <- College[test, ]
```

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

```
fit.lm <- lm(Apps ~ ., data = College.train)
pred.lm <- predict(fit.lm, College.test)
mean((pred.lm - College.test$Apps)^2)
```

```
## [1] 1538442
```

The test error obtained is 1538442.

(c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

```
train.mat <- model.matrix(Apps ~ ., data = College.train)
test.mat <- model.matrix(Apps ~ ., data = College.test)
grid <- 10 ^ seq(4, -2, length = 100)
fit.ridge <- glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)
cv.ridge <- cv.glmnet(train.mat, College.train$Apps, alpha = 0, lambda = grid, thresh = 1e-12)
bestlam.ridge <- cv.ridge$lambda.min
bestlam.ridge
```

```
## [1] 18.73817
```

```
pred.ridge <- predict(fit.ridge, s = bestlam.ridge, newx = test.mat)
mean((pred.ridge - College.test$Apps)^2)
```

```
## [1] 1608859
```

The test error obtained is 1608859.

(d) Fit a lasso model on the training set, with λ chosen by cross-validation. Report the test error obtained, along with the number of non-zero coefficient estimates.

```
fit.lasso <- glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
cv.lasso <- cv.glmnet(train.mat, College.train$Apps, alpha = 1, lambda = grid, thresh = 1e-12)
bestlam.lasso <- cv.lasso$lambda.min
bestlam.lasso
```

```
## [1] 21.54435
```

```
pred.lasso <- predict(fit.lasso, s = bestlam.lasso, newx = test.mat)
mean((pred.lasso - College.test$Apps)^2)
```

```
## [1] 1635280
```

The test error obtained is 1635280.

The coefficients are,

```
predict(fit.lasso, s = bestlam.lasso, type = "coefficients")
```

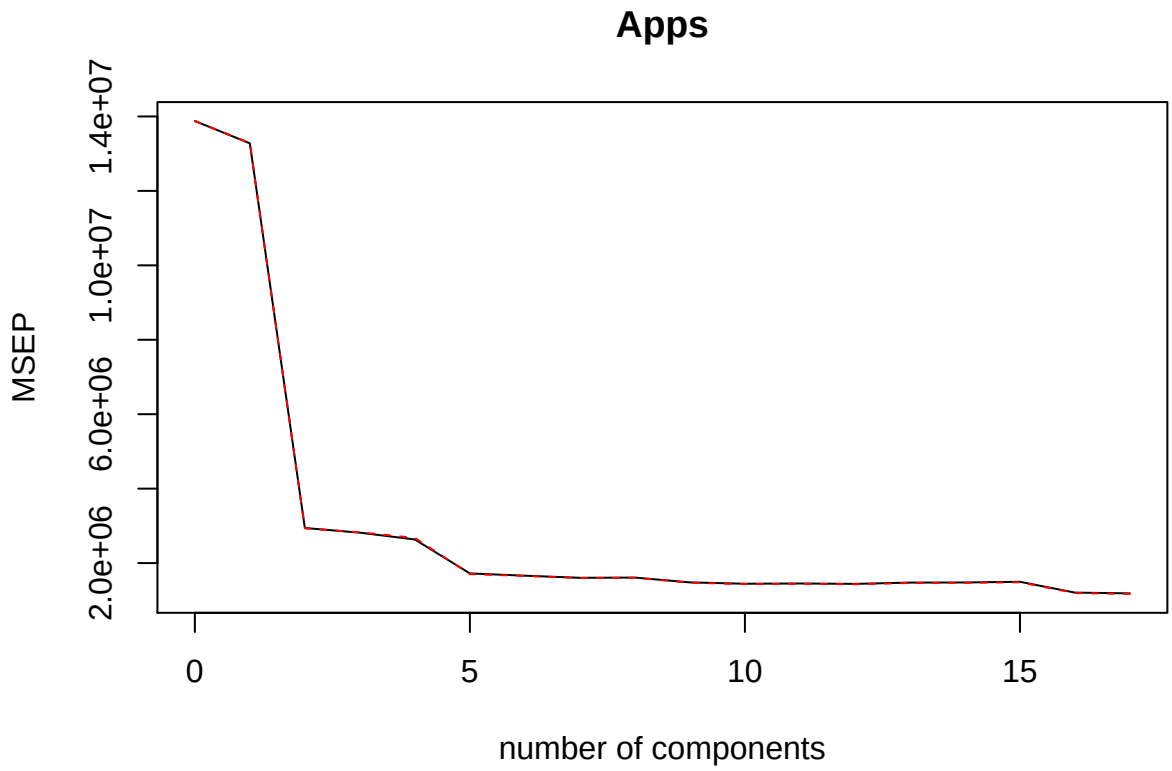
```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          1
## (Intercept) -836.50402310
## (Intercept)    .
## PrivateYes  -385.73749394
## Accept         1.17935134
## Enroll         .
## Top10perc     22.70211938
## Top25perc      .
## F.Undergrad    0.07062149
## P.Undergrad    0.01366763
## Outstate      -0.03424677
## Room.Board     0.01281659
## Books         -0.02167770
## Personal       .
## PhD           -1.46396964
## Terminal      -5.17281004
## S.F.Ratio      5.70969524
## perc.alumni   -9.95007567
## Expend         0.14852541
## Grad.Rate      5.79789861
```

(e) Fit a PCR model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation.

```
library(pls)
```

```
##
## Attaching package: 'pls'
##
## The following object is masked from 'package:stats':
##
##     loadings
```

```
fit.pcr <- pcr(Apps ~ ., data = College.train, scale = TRUE, validation = "CV")
validationplot(fit.pcr, val.type = "MSEP")
```



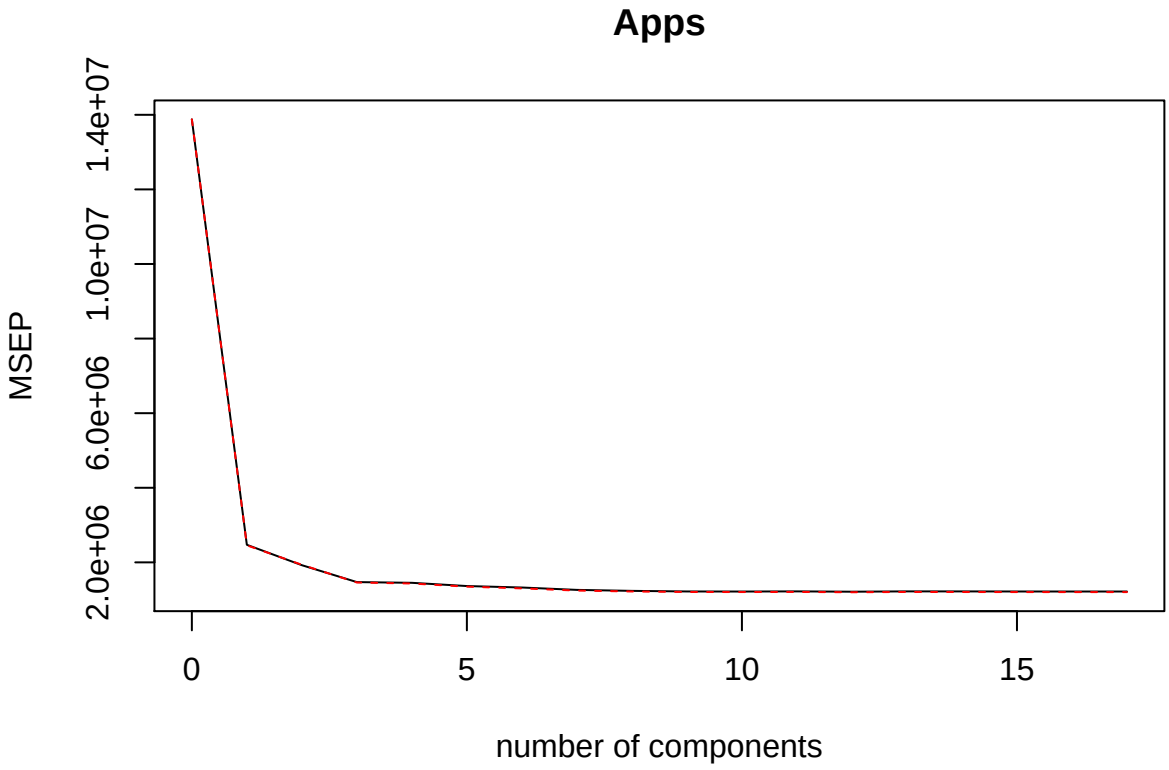

```
pred.pcr <- predict(fit.pcr, College.test, ncomp = 10)
mean((pred.pcr - College.test$Apps)^2)
```

```
## [1] 3014496
```

The test error obtained is 3014496.

(f) Fit a PLS model on the training set, with M chosen by cross-validation. Report the test error obtained, along with the value of M selected by cross-validation.

```
fit.pls <- plsr(Apps ~ ., data = College.train, scale = TRUE, validation = "CV")
validationplot(fit.pls, val.type = "MSEP")
```



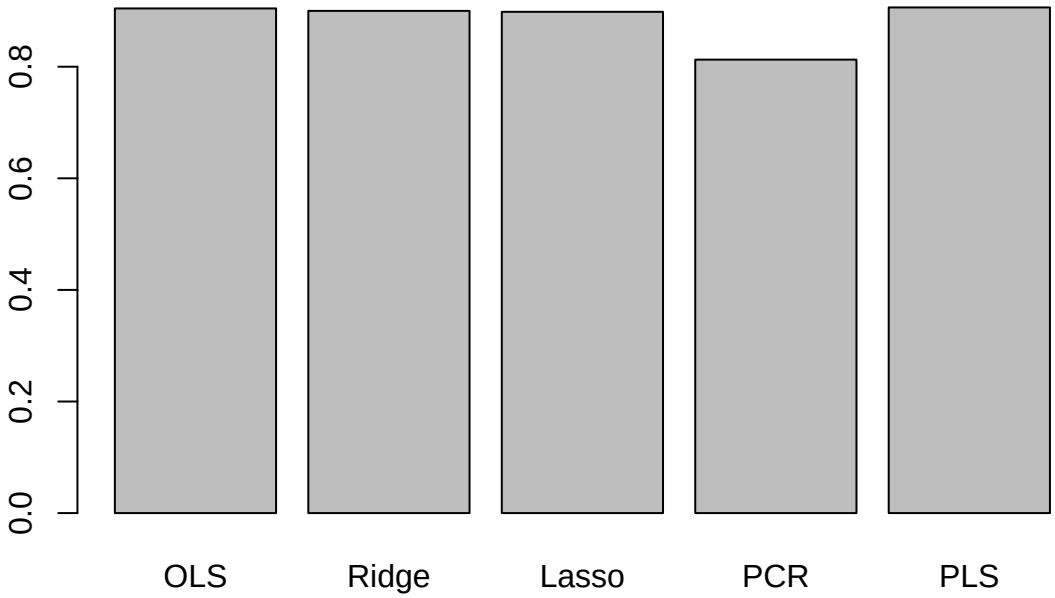

```
pred.pls <- predict(fit.pls, College.test, ncomp = 10)
mean((pred.pls - College.test$Apps)^2)
```

```
## [1] 1508987
```

The test error obtained is 1508987.

(g) Comment on the results obtained. How accurately can we predict the number of college applications received ? Is there much difference among the test errors resulting from these five approaches ?

```
test.avg <- mean(College.test$Apps)
lm.r2 <- 1 - mean((pred.lm - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
ridge.r2 <- 1 - mean((pred.ridge - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
lasso.r2 <- 1 - mean((pred.lasso - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
pcr.r2 <- 1 - mean((pred.pcr - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
pls.r2 <- 1 - mean((pred.pls - College.test$Apps)^2) / mean((test.avg - College.test$Apps)^2)
barplot(c(lm.r2, ridge.r2, lasso.r2, pcr.r2, pls.r2),
        names.arg=c("OLS","Ridge","Lasso", "PCR", "PLS"), main="Test R-squared")
```



So the test R^2 for least squares is 0.9044281, the test R^2 for ridge is 0.9000536, the test R^2 for lasso is 0.8984123, the test R^2 for pcr is 0.8127319 and the test R^2 for pls is 0.9062579.

The plot shows that test R^2 for all models except PCR are around 0.9, with PLS having slightly higher test R^2 than others. PCR has a smaller test R^2 of less than 0.8. All models except PCR predict college applications with high accuracy.

Hybrid stepwise variable selection

```
regfit.fwd = regsubsets(Apps ~ ., data = College.train, method ="forward")
summary(regfit.fwd)
```

```
## Subset selection object
## Call: regsubsets.formula(Apps ~ ., data = College.train, method = "forward")
## 17 Variables  (and intercept)
##             Forced in Forced out
## PrivateYes      FALSE      FALSE
## Accept          FALSE      FALSE
## Enroll          FALSE      FALSE
## Top10perc       FALSE      FALSE
## Top25perc       FALSE      FALSE
## F.Undergrad     FALSE      FALSE
## P.Undergrad     FALSE      FALSE
## Outstate        FALSE      FALSE
## Room.Board      FALSE      FALSE
## Books           FALSE      FALSE
## Personal        FALSE      FALSE
## PhD             FALSE      FALSE
## Terminal        FALSE      FALSE
## S.F.Ratio       FALSE      FALSE
```

```
## perc.alumni     FALSE      FALSE
## Expend          FALSE      FALSE
## Grad.Rate       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: forward
##          PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 ) " "        "*"    " "    " "        " "        " "
## 2  ( 1 ) " "        "*"    " "    " "        " "        " "
## 3  ( 1 ) "*"        "*"    " "    " "        " "        " "
## 4  ( 1 ) "*"        "*"    " "    "*"        " "        " "
## 5  ( 1 ) "*"        "*"    " "    "*"        " "        " "
## 6  ( 1 ) "*"        "*"    " "    "*"        " "        " "
## 7  ( 1 ) "*"        "*"    " "    "*"        " "        "*"
## 8  ( 1 ) "*"        "*"    "*"    "*"        " "        "*"
##          P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## 1  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 2  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 3  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 4  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 5  ( 1 ) " "          "*"      " "        " "   " "      " " " "
## 6  ( 1 ) " "          "*"      " "        " "   " "      " " "*"
## 7  ( 1 ) " "          "*"      " "        " "   " "      " " "*"
## 8  ( 1 ) " "          "*"      " "        " "   " "      " " "*"
##          S.F.Ratio perc.alumni Expend Grad.Rate
## 1  ( 1 ) " "       " "          " "    " "
## 2  ( 1 ) " "       " "          "*"    " "
## 3  ( 1 ) " "       " "          "*"    " "
## 4  ( 1 ) " "       " "          "*"    " "
## 5  ( 1 ) " "       " "          "*"    " "
## 6  ( 1 ) " "       " "          "*"    " "
## 7  ( 1 ) " "       " "          "*"    " "
## 8  ( 1 ) " "       " "          "*"    " "
```

```
regfit.bwd = regsubsets(Apps ~ ., data = College.train, method ="backward")
summary(regfit.bwd)
```

```
## Subset selection object
## Call: regsubsets.formula(Apps ~ ., data = College.train, method = "backward")
## 17 Variables  (and intercept)
##             Forced in Forced out
## PrivateYes      FALSE      FALSE
## Accept          FALSE      FALSE
## Enroll          FALSE      FALSE
## Top10perc       FALSE      FALSE
## Top25perc       FALSE      FALSE
## F.Undergrad     FALSE      FALSE
## P.Undergrad     FALSE      FALSE
## Outstate        FALSE      FALSE
## Room.Board      FALSE      FALSE
## Books           FALSE      FALSE
## Personal        FALSE      FALSE
## PhD             FALSE      FALSE
## Terminal        FALSE      FALSE
## S.F.Ratio       FALSE      FALSE
```

```
## perc.alumni     FALSE      FALSE
## Expend          FALSE      FALSE
## Grad.Rate       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: backward
##          PrivateYes Accept Enroll Top10perc Top25perc F.Undergrad
## 1  ( 1 ) " "        "*"    " "    " "        " "        " "
## 2  ( 1 ) " "        "*"    " "    " "        " "        " "
## 3  ( 1 ) " "        "*"    " "    " "        " "        " "
## 4  ( 1 ) " "        "*"    " "    "*"        " "        " "
## 5  ( 1 ) " "        "*"    " "    "*"        " "        "*"
## 6  ( 1 ) " "        "*"    "*"    "*"        " "        "*"
## 7  ( 1 ) " "        "*"    "*"    "*"        " "        "*"
## 8  ( 1 ) "*"        "*"    "*"    "*"        " "        "*"
##          P.Undergrad Outstate Room.Board Books Personal PhD Terminal
## 1  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 2  ( 1 ) " "          " "      " "        " "   " "      " " " "
## 3  ( 1 ) " "          "*"      " "        " "   " "      " " " "
## 4  ( 1 ) " "          "*"      " "        " "   " "      " " " "
## 5  ( 1 ) " "          "*"      " "        " "   " "      " " " "
## 6  ( 1 ) " "          "*"      " "        " "   " "      " " " "
## 7  ( 1 ) " "          "*"      " "        " "   " "      " " "*"
## 8  ( 1 ) " "          "*"      " "        " "   " "      " " "*"
##          S.F.Ratio perc.alumni Expend Grad.Rate
## 1  ( 1 ) " "       " "         " "    " "
## 2  ( 1 ) " "       " "         "*"    " "
## 3  ( 1 ) " "       " "         "*"    " "
## 4  ( 1 ) " "       " "         "*"    " "
## 5  ( 1 ) " "       " "         "*"    " "
## 6  ( 1 ) " "       " "         "*"    " "
## 7  ( 1 ) " "       " "         "*"    " "
## 8  ( 1 ) " "       " "         "*"    " "
```

```
coef(regfit.fwd,3)
```

```
##  (Intercept)   PrivateYes       Accept       Expend
## -912.6286874 -722.4973607    1.3146726    0.1835056
```

```
coef(regfit.bwd,3)
```

```
##   (Intercept)        Accept      Outstate        Expend
## -923.40880186    1.36393218   -0.09463387    0.22354203
```

Using forward stepwise selection, the best one variable model contains only Accept, and the best two-variable model additionally includes Expend. For this data, the best one-variable andtwo-variable models are each identical for forward and backward selection. However, the best three-variable models identified by forward stepwise selection and backward stepwise selection are different.

Elastic net

```
fit.elastic <- glmnet(train.mat, College.train$Apps, alpha = 0.5, lambda = grid, thresh = 1e-12)
cv.elastic <- cv.glmnet(train.mat, College.train$Apps, alpha = 0.5, lambda = grid, thresh = 1e-12)
bestlam.elastic <- cv.elastic$lambda.min
bestlam.elastic
```

```
## [1] 37.64936
```

```
pred.elastic <- predict(fit.elastic, s = bestlam.elastic, newx = test.mat)
mean((pred.elastic - College.test$Apps)^2)
```

```
## [1] 1685379
```

The test error obtained is 1685379.

Exercise 4 (Applied: Bootstrap and Lasso; to be graded in detail)

Here we use the bootstrap as the basis for inference with the lasso.

(a)

```
library(boot)
set.seed(1)
coeffcients.fn=function(data,index){
  coef.lasso <- predict(glmnet(train.mat[index,], College.train[index,]$Apps,
                               alpha = 1, lambda = grid, thresh = 1e-12), s =
                          bestlam.lasso, type = "coefficients")
  return (coef.lasso[1:19])}
coeffcients.fn(College.train,1:100)
```

```
##  [1] -1.973681e+02  0.000000e+00 -1.051225e+03  1.203868e+00  0.000000e+00
##  [6]  3.019600e+01  5.403740e+00  1.170494e-02  0.000000e+00  0.000000e+00
## [11] -8.064610e-03  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
## [16]  0.000000e+00 -1.490155e+01  1.036121e-01 -2.391826e+00
```

```
coeffcients.fn(College.train,sample(100,100, replace =T))
```

```
##  [1] -3.029779e+02  0.000000e+00 -8.589471e+02  1.059071e+00  1.915848e+00
##  [6]  3.896378e+01  0.000000e+00  0.000000e+00 -5.172074e-02  1.832240e-02
## [11] -1.899018e-03 -2.592151e-01 -2.155388e-01 -3.530466e+00 -3.238002e+00
## [16]  0.000000e+00  0.000000e+00  3.118203e-02  1.206466e+00
```

```
boot(College.train,coeffcients.fn, R=100)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
```

```
## Call:
## boot(data = College.train, statistic = coeffcients.fn, R = 100)
##
##
## Bootstrap Statistics :
##           original       bias     std. error
## t1*  -836.50402310 30.908664464 466.58005670
## t2*     0.00000000  0.000000000   0.00000000
## t3*  -385.73749394  8.166822334 212.28386071
## t4*     1.17935134  0.019198514   0.10566225
## t5*     0.00000000  0.028762784   0.15713778
## t6*    22.70211938  2.509317748  11.05479377
## t7*     0.00000000 -1.737022492   4.97875659
## t8*     0.07062149 -0.014014833   0.05799887
## t9*     0.01366763  0.010483327   0.04798777
## t10*   -0.03424677 -0.001234856   0.02819913
## t11*    0.01281659  0.012008024   0.03140475
## t12*   -0.02167770 -0.048060081   0.18757730
## t13*    0.00000000 -0.000274648   0.04027511
## t14*   -1.46396964 -1.411275675   3.09815431
## t15*   -5.17281004  1.482201659   3.25344415
## t16*    5.70969524  1.283007706  17.99745883
## t17*   -9.95007567  0.535643275   4.93765126
## t18*    0.14852541 -0.010701314   0.04545214
## t19*    5.79789861  0.142461989   3.88968418
```

(b)

```
set.seed(1)
coeffcients.fn2=function(data,index){
  cv.lasso2 <- cv.glmnet(train.mat[index,], College.train[index,]$Apps,
                         alpha =1, lambda = grid, thresh = 1e-12)
  bestlam.lasso2 <- cv.lasso2$lambda.min
  coef.lasso <- predict(glmnet(train.mat[index,], College.train[index,]$Apps,
                               alpha = 1, lambda = grid, thresh = 1e-12), s =
                          bestlam.lasso2, type = "coefficients")
  return (coef.lasso[1:19])}
boot(College.train,coeffcients.fn2, R=100)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = College.train, statistic = coeffcients.fn2, R = 100)
##
##
## Bootstrap Statistics :
##           original        bias     std. error
## t1*  -836.50402310 161.216426511 545.31687568
## t2*     0.00000000   0.000000000   0.00000000
## t3*  -385.73749394  17.871251422 225.66322414
```

```
## t4*     1.17935134   0.082557347   0.11861897
## t5*     0.00000000  -0.265252710   0.52612812
## t6*    22.70211938   8.686755939  13.97252074
## t7*     0.00000000  -5.887133970   8.20098248
## t8*     0.07062149   0.012416577   0.09353858
## t9*     0.01366763   0.014603857   0.06393148
## t10*   -0.03424677  -0.020786973   0.03512853
## t11*    0.01281659   0.018136515   0.04066008
## t12*   -0.02167770  -0.098792952   0.24527478
## t13*    0.00000000  -0.001413179   0.06050959
## t14*   -1.46396964  -2.072434879   3.94908823
## t15*   -5.17281004   0.516000032   4.02643528
## t16*    5.70969524   3.797232605  22.59197899
## t17*   -9.95007567   0.709822954   5.17441564
## t18*    0.14852541  -0.002047539   0.04818691
## t19*    5.79789861   2.177432866   4.30438779
```

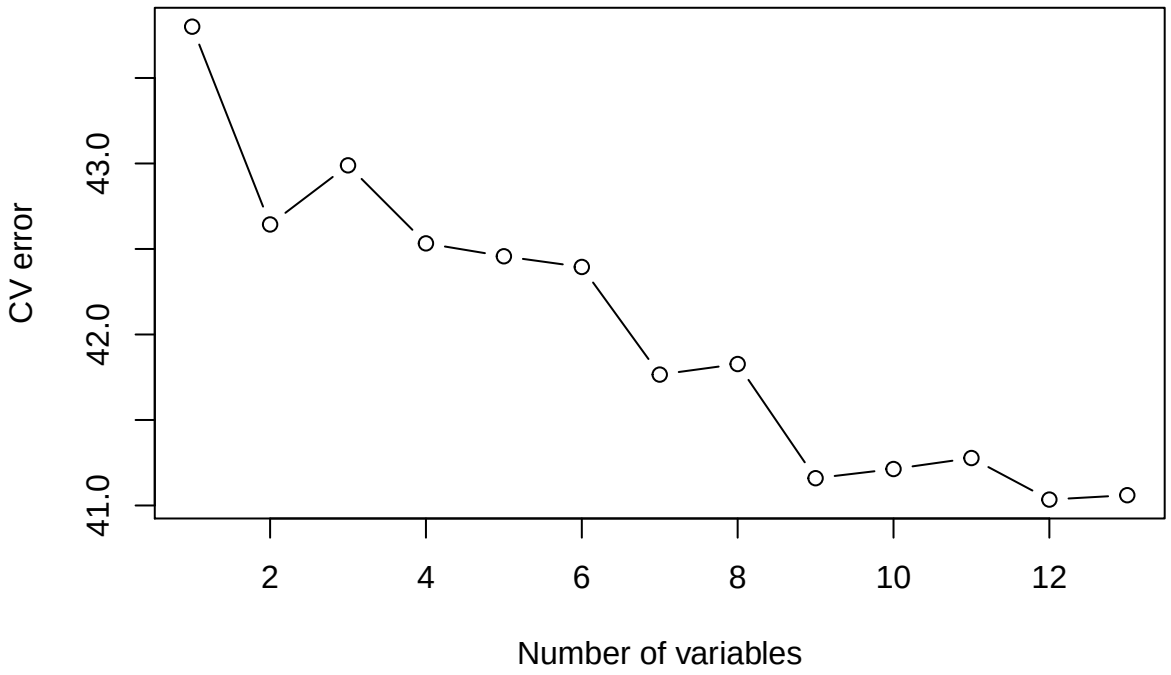
Exercise 5 (Applied: Model Comparison)

Exercise 11 (p. 264): Compare different methods to predict per capita crime rate in the Boston data set.

(a) Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression and PCR. Present and discuss results for the approaches that you consider.

Best subset selection:

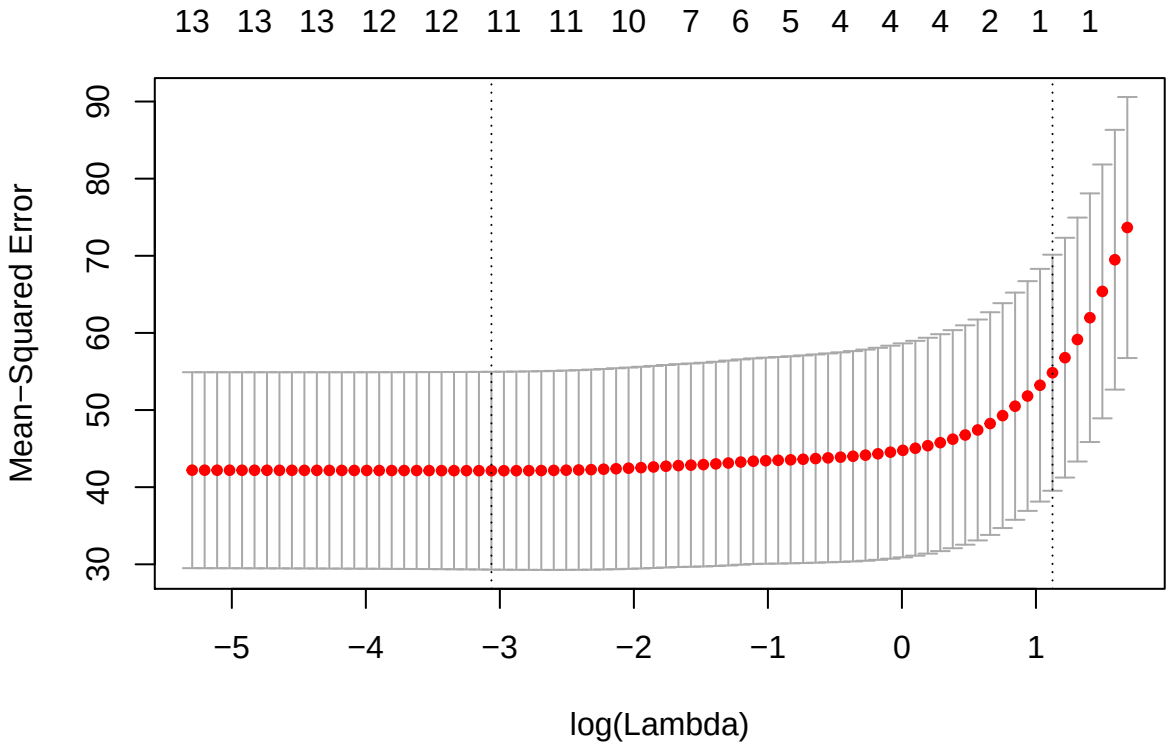
```r
library(MASS)
data(Boston)
set.seed(1)

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
}

k = 10
folds <- sample(1:k, nrow(Boston), replace = TRUE)
cv.errors <- matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))
for (j in 1:k) {
    best.fit <- regsubsets(crim ~ ., data = Boston[folds != j, ], nvmax = 13)
    for (i in 1:13) {
        pred <- predict(best.fit, Boston[folds == j, ], id = i)
        cv.errors[j, i] <- mean((Boston$crim[folds == j] - pred)^2)
    }
}
mean.cv.errors <- apply(cv.errors, 2, mean)
plot(mean.cv.errors, type = "b", xlab = "Number of variables", ylab = "CV error")
```



We may see that cross-validation selects an 12-variables model. We have a CV estimate for the test MSE equal to 41.0345657.

Lasso:

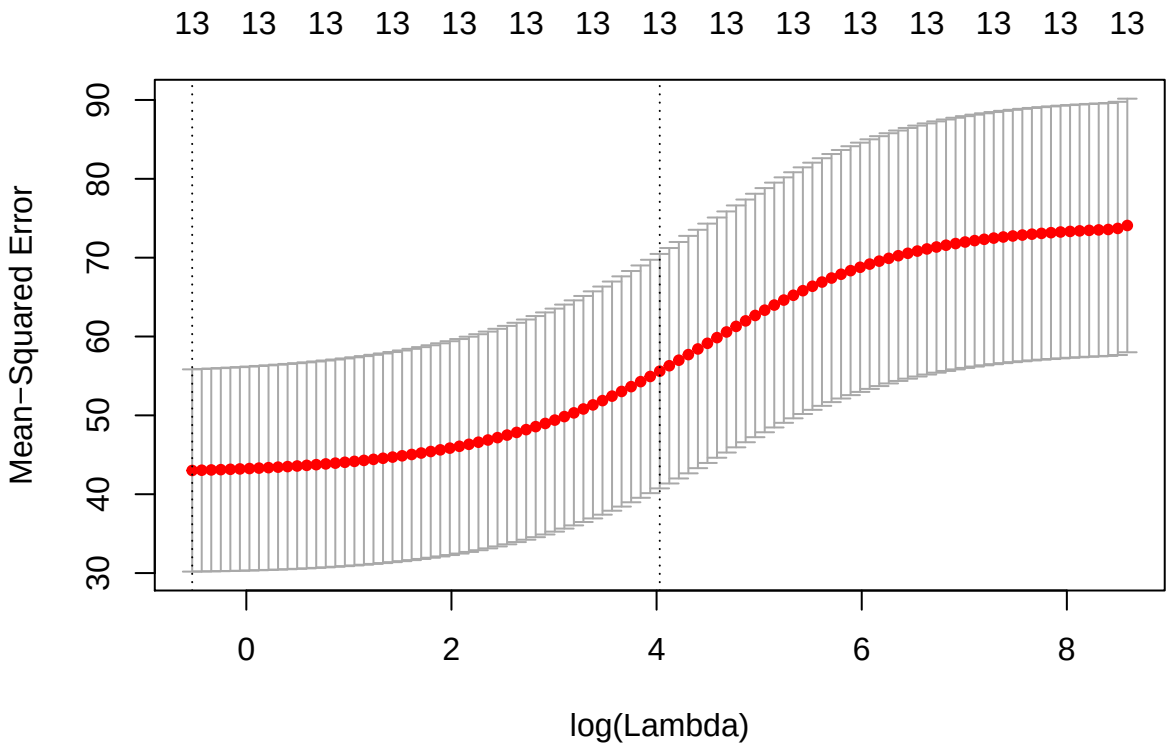
```
x <- model.matrix(crim ~ ., Boston)[, -1]
y <- Boston$crim
cv.out <- cv.glmnet(x, y, alpha = 1, type.measure = "mse")
plot(cv.out)
```



Here cross-validation selects a λ equal to 0.0467489. We have a CV estimate for the test MSE equal to 42.1346961.

Ridge regression:

```
cv.out <- cv.glmnet(x, y, alpha = 0, type.measure = "mse")
plot(cv.out)
```



Here cross-validation selects a λ equal to 0.5899047. We have a CV estimate for the test MSE equal to 43.0116767.

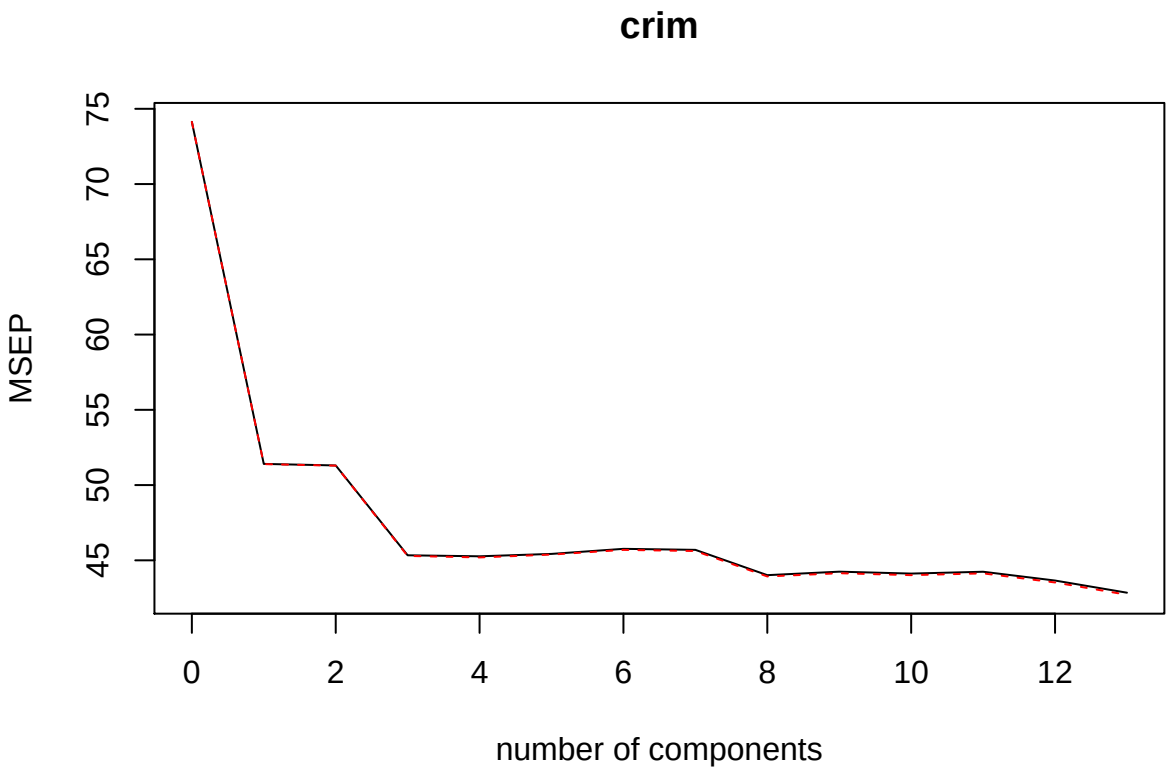
PCR:

```
pcr.fit <- pcr(crim ~ ., data = Boston, scale = TRUE, validation = "CV")
summary(pcr.fit)
```

```
## Data:    X dimension: 506 13
##  Y dimension: 506 1
## Fit method: svdpc
## Number of components considered: 13
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            8.61    7.170    7.163    6.733    6.728    6.740    6.765
## adjCV         8.61    7.169    7.162    6.730    6.723    6.737    6.760
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       6.760    6.634    6.652     6.642     6.652     6.607     6.546
## adjCV    6.754    6.628    6.644     6.635     6.643     6.598     6.536
##
```

```
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       47.70    60.36    69.67    76.45    82.99    88.00    91.14
## crim    30.69    30.87    39.27    39.61    39.61    39.86    40.14
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## X       93.45    95.40     97.04     98.46     99.52     100.0
## crim    42.47    42.55     42.78     43.04     44.13      45.4
```

```r
validationplot(pcr.fit, val.type = "MSEP")
```



Here cross-validation selects M to be equal to 14 (so, no dimension reduction). We have a CV estimate for the test MSE equal to 45.693568.

(b) Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, cross-validation, or some other reasonable alternative, as opposed to using training error.

As computed above the model with the lower cross-validation error is the one chosen by the best subset selection method.

(c) Does your chosen model involve all of the features in the data set ? Why or why not ?

No, the model chosen by the best subset selection method has only 13 predictors.

Exercise 6 (Applied: Model Selection and GAMs)

(a) Get an overview of the data and do simple descriptive analysis including a correlation analysis and a scatterplot matrix.

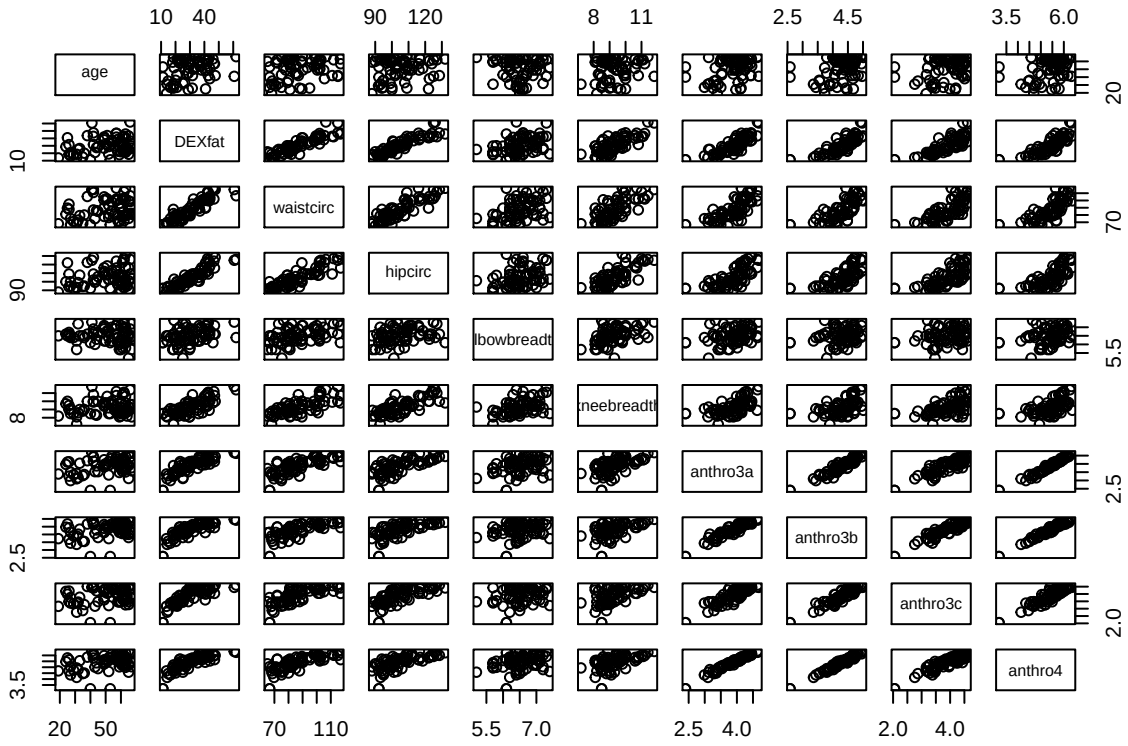
```
library(TH.data)
```

```
## Loading required package: survival
##
## Attaching package: 'survival'
##
## The following object is masked from 'package:boot':
##
##     aml
##
##
## Attaching package: 'TH.data'
##
## The following object is masked from 'package:MASS':
##
##     geyser
```

```
attach(bodyfat)
cor(subset(bodyfat))
```

```
##                      age    DEXfat waistcirc   hipcirc elbowbreadth
## age           1.00000000 0.2710550 0.2385017 0.1804920  -0.06674681
## DEXfat        0.27105504 1.0000000 0.8986535 0.9021881   0.35357317
## waistcirc     0.23850168 0.8986535 1.0000000 0.8712982   0.40092902
## hipcirc       0.18049198 0.9021881 0.8712982 1.0000000   0.33345442
## elbowbreadth -0.06674681 0.3535732 0.4009290 0.3334544   1.00000000
## kneebreadth   0.12814505 0.7680517 0.7316143 0.7588811   0.46257902
## anthro3a      0.33449669 0.8370327 0.7554789 0.7109292   0.32531031
## anthro3b      0.33236952 0.8090941 0.7073444 0.6663164   0.25326405
## anthro3c      0.28129947 0.8095437 0.7399948 0.6888104   0.24064489
## anthro4       0.34451978 0.8230120 0.7398512 0.6876481   0.29374463
##               kneebreadth  anthro3a  anthro3b  anthro3c   anthro4
## age             0.1281450 0.3344967 0.3323695 0.2812995 0.3445198
## DEXfat          0.7680517 0.8370327 0.8090941 0.8095437 0.8230120
## waistcirc       0.7316143 0.7554789 0.7073444 0.7399948 0.7398512
## hipcirc         0.7588811 0.7109292 0.6663164 0.6888104 0.6876481
## elbowbreadth    0.4625790 0.3253103 0.2532641 0.2406449 0.2937446
## kneebreadth     1.0000000 0.5831628 0.5115644 0.5375152 0.5400872
## anthro3a        0.5831628 1.0000000 0.9509316 0.8838199 0.9819509
## anthro3b        0.5115644 0.9509316 1.0000000 0.9333474 0.9838199
## anthro3c        0.5375152 0.8838199 0.9333474 1.0000000 0.9170213
## anthro4         0.5400872 0.9819509 0.9838199 0.9170213 1.0000000
```

```
pairs(bodyfat)
```



(b) Fit a linear model assuming normal errors. Are all potential covariates informative? Check the results against a model that underwent AIC-based variable selection.

```
lm1 = lm(DEXfat~., data=bodyfat)
summary(lm1)
```

```
##
## Call:
## lm(formula = DEXfat ~ ., data = bodyfat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.954 -1.949 -0.219  1.169 10.812
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -69.02828    7.51686  -9.183 4.18e-13 ***
## age            0.01996    0.03221   0.620  0.53777
## waistcirc      0.21049    0.06714   3.135  0.00264 **
## hipcirc        0.34351    0.08037   4.274 6.85e-05 ***
## elbowbreadth  -0.41237    1.02291  -0.403  0.68826
## kneebreadth    1.75798    0.72495   2.425  0.01829 *
## anthro3a       5.74230    5.20752   1.103  0.27449
## anthro3b       9.86643    5.65786   1.744  0.08622 .
## anthro3c       0.38743    2.08746   0.186  0.85338
## anthro4       -6.57439    6.48918  -1.013  0.31500
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.281 on 61 degrees of freedom
## Multiple R-squared:  0.9231, Adjusted R-squared:  0.9117
## F-statistic: 81.35 on 9 and 61 DF,  p-value: < 2.2e-16
```

There is a relatioship between the predictors and the response by testing the null hypothesis of whether all the regression coefficients are zero. The F -statistic is has a small p-value, less than 0.01, indicating that we can reject the null hypothesis. Therefore, there is a significant relatioship between the predictors and the response DEXfat.

From the p-values of each predictor, waistcirc, hipcirc and kneebreadth have a statistically significant relationship to the response(since they have small p-values < 0.01).

The potential covariates are informative. From the correlation matrix, waistcirc and hipcirc are higher correlated with DEXfat($>=0.9$).

AIC:

```
null=lm(DEXfat~1, data=bodyfat)
null
```

```
##
## Call:
## lm(formula = DEXfat ~ 1, data = bodyfat)
##
## Coefficients:
## (Intercept)
##       30.78
```

```
full=lm(DEXfat~., data=bodyfat)
full
```

```
##
## Call:
## lm(formula = DEXfat ~ ., data = bodyfat)
##
## Coefficients:
##  (Intercept)           age     waistcirc       hipcirc  elbowbreadth
##    -69.02828       0.01996       0.21049       0.34351      -0.41237
##  kneebreadth      anthro3a      anthro3b      anthro3c       anthro4
##      1.75798       5.74230       9.86643       0.38743      -6.57439
```

```
step(null, scope=list(lower=null, upper=full), direction="forward")
```

```
## Start:  AIC=342.04
## DEXfat ~ 1
##
##               Df Sum of Sq    RSS    AIC
## + hipcirc      1    6947.8 1588.2 224.64
## + waistcirc    1    6893.5 1642.5 227.03
## + anthro3a     1    5980.5 2555.5 258.42
## + anthro4      1    5781.8 2754.1 263.73
## + anthro3c     1    5594.2 2941.8 268.41
```

```
## + anthro3b      1    5587.9 2948.0 268.56
## + kneebreadth   1    5035.4 3500.6 280.76
## + elbowbreadth  1    1067.1 7468.9 334.56
## + age           1     627.1 7908.8 338.63
## <none>                      8536.0 342.04
##
## Step:  AIC=224.64
## DEXfat ~ hipcirc
##
##                Df Sum of Sq     RSS    AIC
## + anthro4       1    664.83  923.35 188.14
## + anthro3b      1    663.87  924.30 188.21
## + anthro3a      1    660.60  927.58 188.46
## + anthro3c      1    574.72 1013.45 194.75
## + waistcirc     1    449.20 1138.98 203.04
## + kneebreadth   1    139.99 1448.19 220.09
## + age           1    103.33 1484.85 221.87
## <none>                      1588.18 224.64
## + elbowbreadth  1     26.71 1561.47 225.44
##
## Step:  AIC=188.14
## DEXfat ~ hipcirc + anthro4
##
##                Df Sum of Sq    RSS    AIC
## + waistcirc     1   143.673 779.67 178.13
## + kneebreadth   1   117.613 805.74 180.46
## <none>                      923.35 188.14
## + anthro3c      1    17.492 905.86 188.78
## + anthro3a      1     9.548 913.80 189.40
## + anthro3b      1     9.341 914.01 189.42
## + elbowbreadth  1     7.577 915.77 189.55
## + age           1     5.385 917.96 189.72
##
## Step:  AIC=178.13
## DEXfat ~ hipcirc + anthro4 + waistcirc
##
##                Df Sum of Sq    RSS    AIC
## + kneebreadth   1    70.214 709.46 173.43
## + anthro3b      1    26.431 753.24 177.68
## <none>                      779.67 178.13
## + anthro3c      1     7.509 772.17 179.44
## + anthro3a      1     5.993 773.68 179.58
## + age           1     2.843 776.83 179.87
## + elbowbreadth  1     0.013 779.66 180.13
##
## Step:  AIC=173.43
## DEXfat ~ hipcirc + anthro4 + waistcirc + kneebreadth
##
##                Df Sum of Sq    RSS    AIC
## + anthro3b      1    33.065 676.40 172.04
## <none>                      709.46 173.43
## + anthro3c      1     8.980 700.48 174.53
## + elbowbreadth  1     6.575 702.89 174.77
## + age           1     4.091 705.37 175.02
```

```
## + anthro3a      1     0.411 709.05 175.39
##
## Step:  AIC=172.04
## DEXfat ~ hipcirc + anthro4 + waistcirc + kneebreadth + anthro3b
##
##                Df Sum of Sq    RSS    AIC
## <none>                      676.40 172.04
## + anthro3a      1   12.2733 664.12 172.74
## + age           1    5.0959 671.30 173.50
## + elbowbreadth  1    2.8968 673.50 173.74
## + anthro3c      1    0.0341 676.36 174.04


##
## Call:
## lm(formula = DEXfat ~ hipcirc + anthro4 + waistcirc + kneebreadth +
##     anthro3b, data = bodyfat)
##
## Coefficients:
## (Intercept)      hipcirc      anthro4    waistcirc  kneebreadth
##    -71.8319       0.3537      -0.7684       0.2070       1.8126
##    anthro3b
##      8.0581
```

```
step(full, data=bodyfat, direction="backward")
```

```
## Start:  AIC=177.92
## DEXfat ~ age + waistcirc + hipcirc + elbowbreadth + kneebreadth +
##     anthro3a + anthro3b + anthro3c + anthro4
##
##                Df Sum of Sq    RSS    AIC
## - anthro3c      1     0.371 656.89 175.96
## - elbowbreadth  1     1.749 658.27 176.11
## - age           1     4.133 660.65 176.37
## - anthro4       1    11.047 667.57 177.11
## - anthro3a      1    13.087 669.61 177.32
## <none>                      656.52 177.92
## - anthro3b      1    32.729 689.25 179.38
## - kneebreadth   1    63.289 719.81 182.46
## - waistcirc     1   105.765 762.28 186.53
## - hipcirc       1   196.600 853.12 194.52
##
## Step:  AIC=175.96
## DEXfat ~ age + waistcirc + hipcirc + elbowbreadth + kneebreadth +
##     anthro3a + anthro3b + anthro4
##
##                Df Sum of Sq    RSS    AIC
## - elbowbreadth  1     1.929 658.82 174.17
## - age           1     3.935 660.82 174.39
## - anthro4       1    10.977 667.87 175.14
## - anthro3a      1    12.769 669.66 175.33
## <none>                      656.89 175.96
## - anthro3b      1    42.816 699.71 178.45
## - kneebreadth   1    63.959 720.85 180.56
```

```
## - waistcirc     1   114.954 771.84 185.41
## - hipcirc       1   196.607 853.50 192.55
##
## Step:  AIC=174.17
## DEXfat ~ age + waistcirc + hipcirc + kneebreadth + anthro3a +
##     anthro3b + anthro4
##
##               Df Sum of Sq    RSS    AIC
## - age          1     5.304 664.12 172.74
## - anthro4      1    11.765 670.58 173.43
## - anthro3a     1    12.481 671.30 173.50
## <none>                     658.82 174.17
## - anthro3b     1    46.156 704.97 176.98
## - kneebreadth  1    62.889 721.71 178.65
## - waistcirc    1   113.074 771.89 183.42
## - hipcirc      1   207.704 866.52 191.63
##
## Step:  AIC=172.74
## DEXfat ~ waistcirc + hipcirc + kneebreadth + anthro3a + anthro3b +
##     anthro4
##
##               Df Sum of Sq    RSS    AIC
## - anthro4      1    10.708 674.83 171.88
## - anthro3a     1    12.273 676.40 172.04
## <none>                     664.12 172.74
## - anthro3b     1    44.927 709.05 175.39
## - kneebreadth  1    61.561 725.68 177.03
## - waistcirc    1   116.017 780.14 182.17
## - hipcirc      1   204.154 868.28 189.77
##
## Step:  AIC=171.88
## DEXfat ~ waistcirc + hipcirc + kneebreadth + anthro3a + anthro3b
##
##               Df Sum of Sq    RSS    AIC
## - anthro3a     1     2.038 676.87 170.09
## <none>                     674.83 171.88
## - anthro3b     1    56.512 731.34 175.59
## - kneebreadth  1    70.368 745.20 176.92
## - waistcirc    1   106.780 781.61 180.31
## - hipcirc      1   218.070 892.90 189.76
##
## Step:  AIC=170.09
## DEXfat ~ waistcirc + hipcirc + kneebreadth + anthro3b
##
##               Df Sum of Sq     RSS    AIC
## <none>                      676.87 170.09
## - kneebreadth  1     76.42  753.28 175.69
## - waistcirc    1    114.75  791.61 179.21
## - hipcirc      1    218.67  895.54 187.97
## - anthro3b     1    408.19 1085.05 201.60

##
## Call:
## lm(formula = DEXfat ~ waistcirc + hipcirc + kneebreadth + anthro3b,
```

```
##     data = bodyfat)
##
## Coefficients:
## (Intercept)    waistcirc      hipcirc  kneebreadth     anthro3b
##    -71.7197       0.2037       0.3546       1.8047       7.1264
```

In the forward selection, I start with no predictors, add the variable with the largest t value if it is significant, and stop when none of the t values are significant. The last step is, Step: AIC=172.04 DEXfat ~ hipcirc + anthro4 + waistcirc + kneebreadth + anthro3b

In the backwards elimination, I start with the full model, and eliminate the least significant variable at each stage, until all the variables in the model are significant. The last step is, Step: AIC=170.09 DEXfat ~ waistcirc + hipcirc + kneebreadth + anthro3b

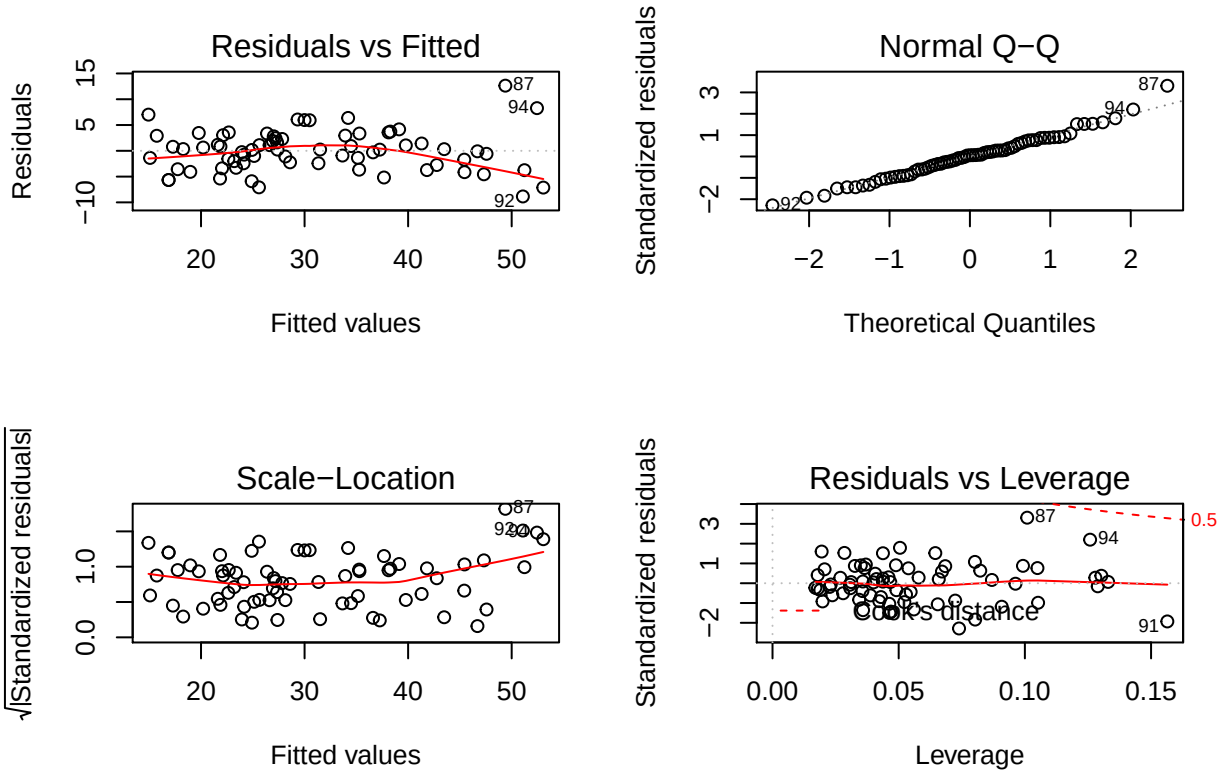
Before we get waistcirc, hipcirc and kneebreadth have a statistically significant relationship to the DEXfat because of small p-values, and these variabels contain in the AIC-based model we selected.

(c) Check the model assumptions of the final model and give short arguments whether they are fullfilled or not.

```
lm2 = lm(DEXfat~waistcirc+hipcirc+kneebreadth, data=bodyfat)
summary(lm2)
```

```
##
## Call:
## lm(formula = DEXfat ~ waistcirc + hipcirc + kneebreadth, data = bodyfat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.8518 -2.5899  0.2273  2.6234 12.6306
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -58.39297    5.71436 -10.219 2.65e-15 ***
## waistcirc     0.33917    0.07163   4.735 1.18e-05 ***
## hipcirc       0.43191    0.09527   4.534 2.46e-05 ***
## kneebreadth   1.51231    0.82878   1.825   0.0725 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.024 on 67 degrees of freedom
## Multiple R-squared:  0.8729, Adjusted R-squared:  0.8672
## F-statistic: 153.4 on 3 and 67 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm2)
```



The model does not fit some assumptions. One is that the curve pattern in the residuals plot indicates non-linearity in the data. The other is from the leverage plot, point 94 and 91 appear to have high leverage, and point 87 appears to be a residual.

(d) Fit a generalized additive model using the results you obtained so far. Use the function gam() in the mgcv package and fitt cubic spline effects using the option bs='cr'.

We'll start with the typical linear model approach.

```
library(mgcv)
```

```
## Loading required package: nlme
## This is mgcv 1.8-11. For overview type 'help("mgcv-package")'.
```

```
gam.m1=gam(DEXfat~waistcirc+hipcirc+kneebreadth,data=bodyfat)
summary(gam.m1)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEXfat ~ waistcirc + hipcirc + kneebreadth
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -58.39297    5.71436 -10.219 2.65e-15 ***
## waistcirc     0.33917    0.07163   4.735 1.18e-05 ***
```

```
## hipcirc       0.43191    0.09527   4.534 2.46e-05 ***
## kneebreadth   1.51231    0.82878   1.825   0.0725 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## R-sq.(adj) =  0.867   Deviance explained = 87.3%
## GCV = 17.162  Scale est. = 16.195    n = 71
```

It appears we have statistical significant effects for waistcirc and hipcirc, but not for kneebreadth, and the adjusted R-squared suggests a notable amount of the variance is accounted for. Now look at the nonlinear effects for each covariate,

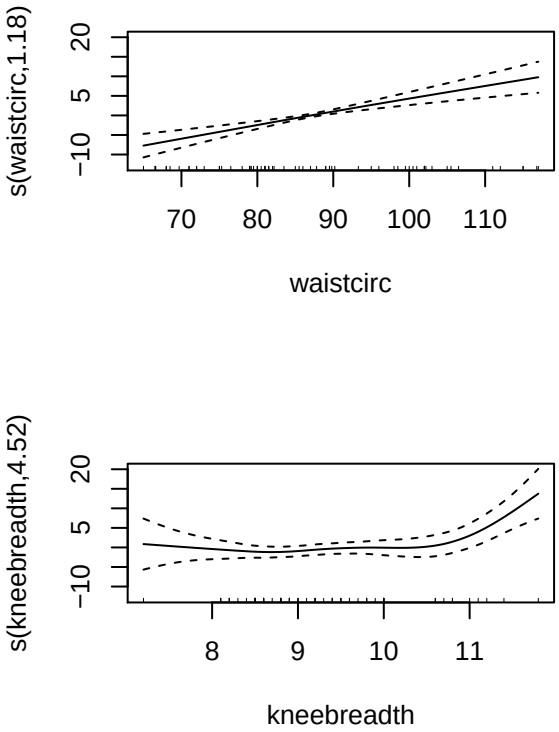
```
gam.m2 <- gam(DEXfat ~ s(waistcirc,bs='cr') + s(hipcirc,bs='cr') + s(kneebreadth,bs='cr'),
              data = bodyfat)
summary(gam.m2)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## DEXfat ~ s(waistcirc, bs = "cr") + s(hipcirc, bs = "cr") + s(kneebreadth,
##     bs = "cr")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  30.7828     0.4217      73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                   edf Ref.df      F  p-value
## s(waistcirc)    1.177  1.321 21.992 7.95e-06 ***
## s(hipcirc)      2.313  2.882 10.423 1.53e-05 ***
## s(kneebreadth) 4.515  5.522  3.581  0.00522 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.896   Deviance explained = 90.8%
## GCV = 14.457  Scale est. = 12.623    n = 71
```

The cubic effects for kneebreadth is significant in the nonlinear model, as well as the effects for waistcirc and hipcirc.

(e) Check again the model assumptions of your final model and compare the results to the one obtained in the linear model.

```
par(mfrow=c(2,2))
plot(gam.m2)
```



The final generalized additive model satisfies the assumptions of the absence of interaction effects and the assumptions of nonlinear which we can see from the plot. The cubic effects for kneebreadth is significant in the nonlinear model, as well as the effects for waistcirc and hipcirc. The conclusion for the nonlinear generalized additive model in is different from the linear generalized additive model regarding the individual effects from $gam.m1$, but it is the same as the linear model assuming normal errors we fit in part b.