HW7

Xinyue Lu

2020/3/8

Problem 1. Use function gam discussed in class to build a model for the prostate cancer data.

```
library('lasso2')
```

```
## R Package to solve regression problems while imposing
##   an L1 constraint on the parameters. Based on S-plus Release 2.1
## Copyright (C) 1998, 1999
## Justin Lokhorst   <jlokhors@stats.adelaide.edu.au>
## Berwin A. Turlach <bturlach@stats.adelaide.edu.au>
## Bill Venables     <wvenable@stats.adelaide.edu.au>
##
## Copyright (C) 2002
## Martin Maechler <maechler@stat.math.ethz.ch>
```

```
data(Prostate)
library(gam)
```

```
## Warning: package 'gam' was built under R version 3.6.3
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 3.6.2
```

```
## Loaded gam 1.16.1
```

Pick 3 continuous predictors lcavol, lweight, lcp. Build 3 univariate models (one predictor at a time) using smoothing spline.

```
gam.1 = gam(lpsa~s(lcavol,3),data=Prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

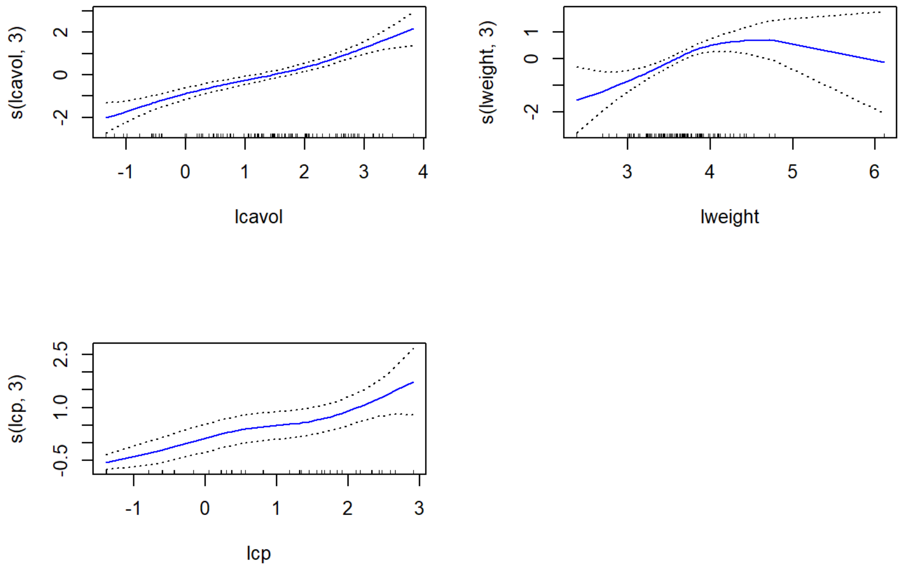
```
gam.2 = gam(lpsa~s(lweight,3),data=Prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
gam.3 = gam(lpsa~s(lcp,3),data=Prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
par(mfrow=c(2,2))
plot(gam.1, se=TRUE ,col="blue")
plot(gam.2, se=TRUE ,col="blue")
plot(gam.3, se=TRUE ,col="blue")
```



Then put all three predictors into gam(), use smoothing splines (option s()).Discuss whether the model with all three predictors improves the fit significantly.

The model with 3 predictors improved the fit significantly. First, this model had the smallest AIC. Second, accroding to the ANOVA tests, this model was better than any of its sub models.

```
gam.4 = gam(lpsa~s(lcavol,3)+s(lweight,3)+s(lcp,3),data=Prostate)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
par(mfrow=c(2,2))
plot(gam.4, se=TRUE ,col="blue")
summary(gam.1)$aic
```

```
## [1] 233.7284
```

```
summary(gam.2)$aic
```

```
## [1] 289.0693
```

```
summary(gam.3)$aic
```

```
## [1] 275.2289
```

```
summary(gam.4)$aic
```

```
## [1] 226.3297
```

```
anova(gam.1,gam.4)
```

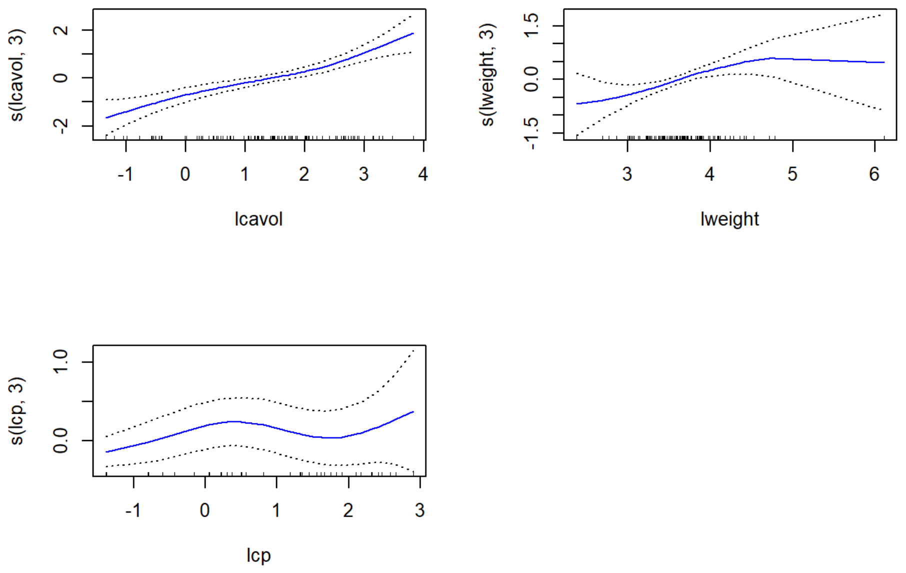
```
## Analysis of Deviance Table
##
## Model 1: lpsa ~ s(lcavol, 3)
## Model 2: lpsa ~ s(lcavol, 3) + s(lweight, 3) + s(lcp, 3)
##   Resid. Df Resid. Dev     Df Deviance Pr(>Chi)
## 1        93     57.015
## 2        87     46.680 6.0001   10.335 0.003745 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(gam.2,gam.4)
```

```
## Analysis of Deviance Table
##
## Model 1: lpsa ~ s(lweight, 3)
## Model 2: lpsa ~ s(lcavol, 3) + s(lweight, 3) + s(lcp, 3)
##   Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
## 1        93     100.87
## 2        87      46.68 6.0001    54.19 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(gam.3,gam.4)
```

```
## Analysis of Deviance Table
##
## Model 1: lpsa ~ s(lcp, 3)
## Model 2: lpsa ~ s(lcavol, 3) + s(lweight, 3) + s(lcp, 3)
##   Resid. Df Resid. Dev     Df Deviance  Pr(>Chi)
## 1        93     87.457
## 2        87     46.680 6.0002   40.777 2.392e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```



Use 10 folds cross validation to estimate predictive MSE.

Accroding to 10-fold cross validation, the model with all the 3 predictors had the smallest cross validation MSE 0.6118.

```
CVgam <- function (formula, data, nfold = 10, debug.level = 0, printit = TRUE, cvparts = NULL, gamma = 1, seed = 29){
    if (is.null(cvparts)) {
        set.seed(seed)
        cvparts <- sample(1:nfold, nrow(data), replace = TRUE)
    }
    folds <- unique(cvparts)
    khat <- hat <- numeric(nrow(data))
    scale.gam <- summary(gam(formula, data = data))$scale
    for (i in folds) {
        trainrows <- cvparts != i
        testrows <- cvparts == i
        elev.gam <- gam(formula, data = data[trainrows, ],
                        gamma = gamma)
        hat[testrows] <- predict(elev.gam, newdata = data[testrows,], select = TRUE)
        res <- residuals(elev.gam)
    }
    y <- eval(formula[[2]], envir = as.data.frame(data))
    res <- y - hat
    cvscale <- sum(res^2)/length(res)
    prntvec <- c(GAMscale = scale.gam, `CV-mse-GAM ` = cvscale)
    if (printit)
        print(round(prntvec, 4))
    invisible(list(fitted = hat, resid = res, cvscale = cvscale, scale.gam = scale.gam))
}
print("lcavol-model")
```

```
## [1] "lcavol-model"
```

```
CVgam.1<-CVgam(lpsa~s(lcavol), data=Prostate, nfold = 10,  seed = 666)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
## CV-mse-GAM
##      0.6493
```

```
print("lweight-model")
```

```
## [1] "lweight-model"
```

```
CVgam.2<-CVgam(lpsa~s(lweight), data=Prostate, nfold = 10,  seed = 666)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
## CV-mse-GAM
##     1.1782
```

```
print("lcp-model")
```

```
## [1] "lcp-model"
```

```
CVgam.3<-CVgam(lpsa~s(lcp), data=Prostate, nfold = 10,  seed = 666)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
## CV-mse-GAM
##     0.9721
```

```
print("lcavol-lweight-lcp-model")
```

```
## [1] "lcavol-lweight-lcp-model"
```

```
CVgam.4<-CVgam(lpsa~s(lcavol)+s(lweight)+s(lcp), data=Prostate, nfold = 10,  seed = 666)
```

```
## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored

## Warning in model.matrix.default(mt, mf, contrasts): non-list contrasts
## argument ignored
```

```
## CV-mse-GAM
##     0.6716
```

Problem 2 Using the dataset Carseats (see the code below), Predict Sales using regression trees and related approaches, treating the response as a quantitative variable

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.2
```

```
#data(package="ISLR")
attach(Carseats)
```

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

```
set.seed(666)
test.idx = sample(1:nrow(Carseats),nrow(Carseats)/3)
train = Carseats[-test.idx,]
test = Carseats[test.idx,]
```

b). Fit a regression tree to the training set. Plot the tree, and interpret the results. What test error rate do you obtain?

The default tree used 7 variables and had 18 terminal nodes. The MSE of the train set was 2.482 while the MSE of the test set was 5.2386. The test MSE was over 1 time larger than the train MSE, which indicated that the tree was overfitting.

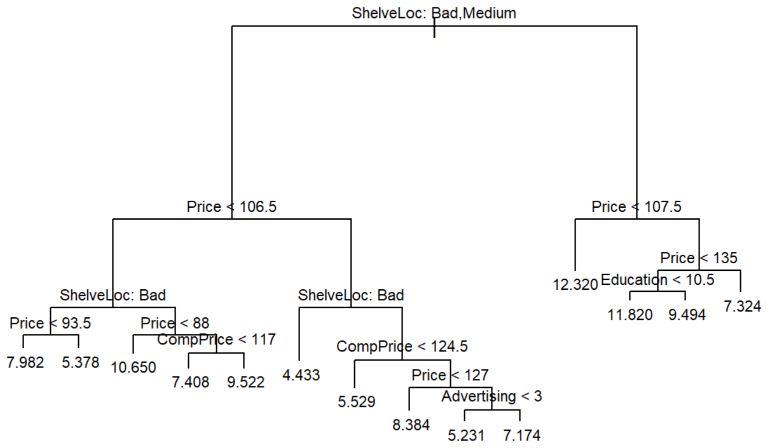
```
library(tree)
```

```
## Warning: package 'tree' was built under R version 3.6.3
```

```
#default tree
set.seed(555)
tree1=tree(Sales~.,data=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = Sales ~ ., data = train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Advertising" "Education"
## Number of terminal nodes:  14
## Residual mean deviance:  2.653 = 671.2 / 253
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -5.23100 -1.03900 -0.01391  0.00000  0.96060  3.95300
```

```
plot(tree1)
text(tree1,pretty=0,cex=0.7)
```



```
#tree1
tree1.pred=predict(tree1, test)
mean((tree1.pred-test$Sales)^2)
```

```
## [1] 4.594368
```

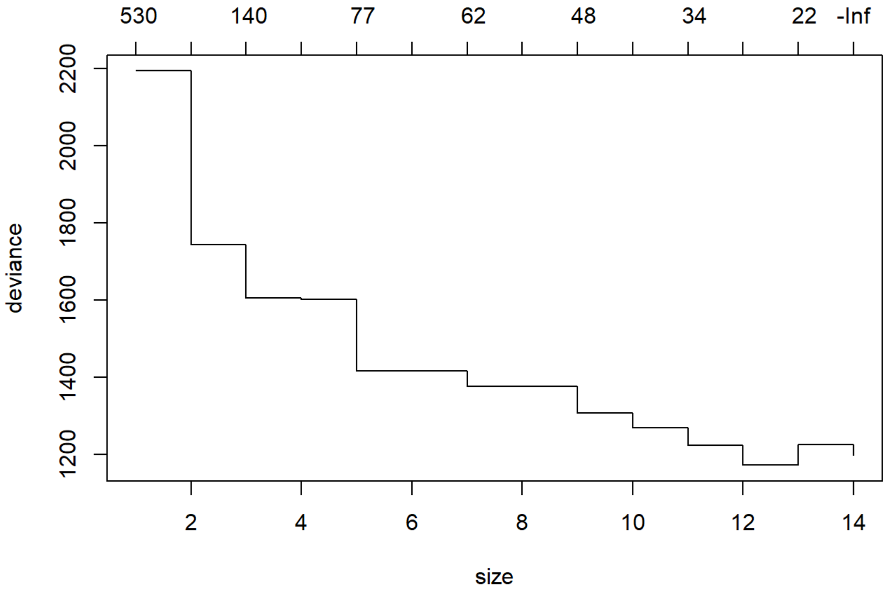
c). Use cross-validation in order to determine the optimal level of tree complexity. Does pruning the tree improve the test error rate?

Accroding to the cross validation results, the tree with the lowest cross validation error used 2 variables and had 5 leaves. The MSE of the train set was 4.262 while the MSE of the test set was 5.0305. Pruning the tree did improve the test error rate. The model also became simpler and more robust.

```
#cross validation
set.seed(666)
tree1.cv <- cv.tree(tree1)
summary(tree1.cv)
```

```
##        Length Class  Mode
## size   14     -none- numeric
## dev    14     -none- numeric
## k      14     -none- numeric
## method  1     -none- character
```

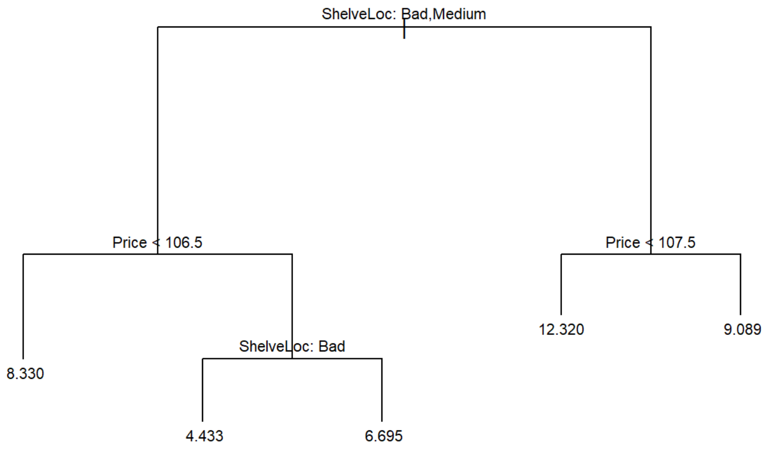
```
plot(tree1.cv)
```



```
# pick the best size from cross validation
tree1.prune<-prune.tree(tree1, best=5)
summary(tree1.prune)
```

```
##
## Regression tree:
## snip.tree(tree = tree1, nodes = c(7L, 11L, 4L))
## Variables actually used in tree construction:
## [1] "ShelveLoc" "Price"
## Number of terminal nodes:  5
## Residual mean deviance:  4.263 = 1117 / 262
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -6.695  -1.367  -0.053   0.000   1.398   5.030
```

```
plot(tree1.prune)
text(tree1.prune,pretty=0,cex=0.7)
```



```
tree1.prune.pred=predict(tree1.prune, test)
mean((tree1.prune.pred-test$Sales)^2)
```

```
## [1] 4.92084
```