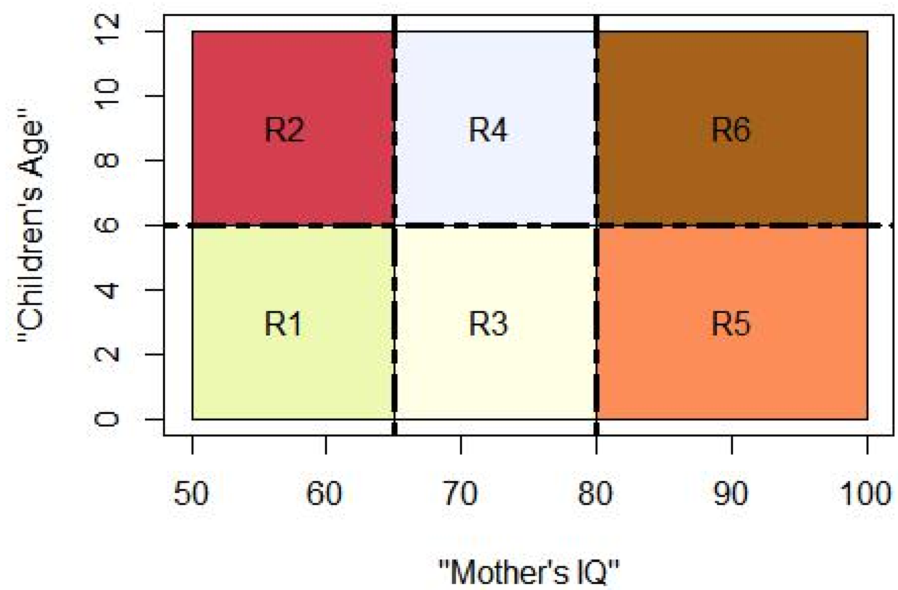
Tree Homework

Jiachen Feng

2021/3/5

8.1

```
plot("Mother's IQ","Children's Age",xlim = c(50,100),ylim = c(0,12))
rect(xleft = 50,ybottom = 0,xright = 100,ytop = 12)
rect(xleft = 50,ybottom = 0,xright = 65,ytop = 6,col = brewer.pal(3,"YlG
nBu"))
rect(xleft = 50,ybottom = 6,xright = 65,ytop = 12,col=brewer.pal(7,"Spec
tral"))
rect(xleft = 65,ybottom = 0,xright = 80,ytop = 6,col=brewer.pal(9,"YlGn
"))
rect(xleft = 65,ybottom = 6,xright = 80,ytop = 12,col=brewer.pal(4,"Blue
s"))
rect(xleft = 80,ybottom = 0,xright = 100,ytop = 6,col=brewer.pal(2,"Spec
tral"))
rect(xleft = 80,ybottom = 6,xright = 100,ytop = 12,col=brewer.pal(4,"BrB
G"))
text(57,3,"R1")
text(57,9,"R2")
text(72,3,"R3")
text(72,9,"R4")
text(90,3,"R5")
text(90,9,"R6")
abline(v=c(65,80),lwd=3,lty=6)
abline(h=6,lwd=3,lty=6)
```



8.2

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

Since we're using depth-one trees,

$$f_j(X_j) = I_1 * \ c_m + I_2 * \ c_m$$

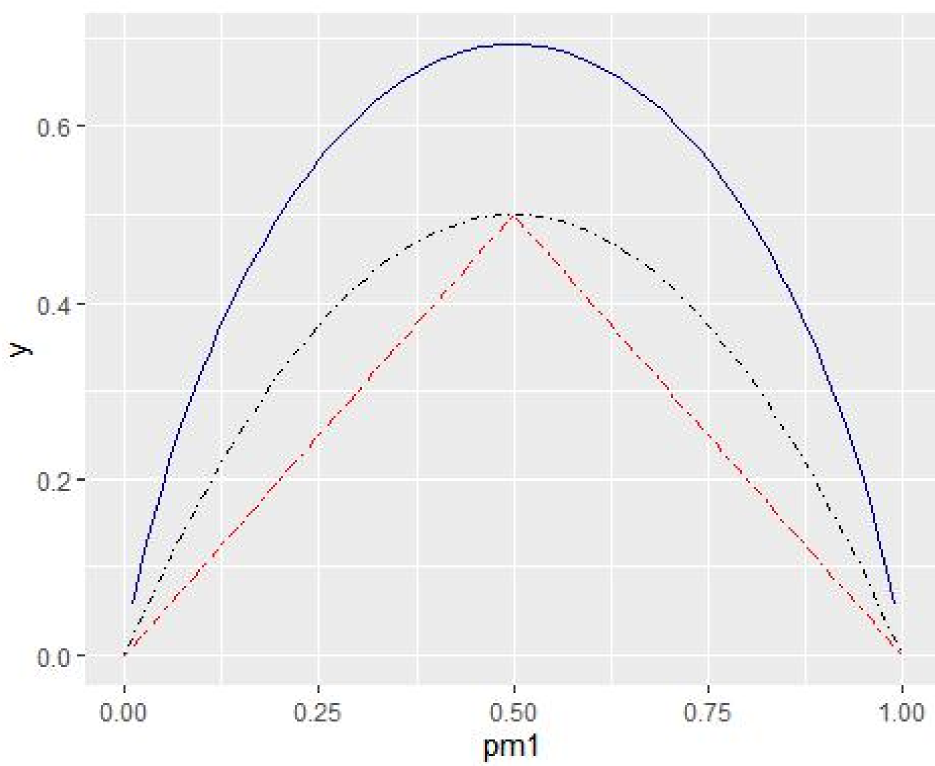
It is additive, so that

$$f(X) = \sum_{j=1}^{p} f_j\,(X_j)$$

8.3

```
errorrate <- function(pm1){
  max <- pmax(pm1,1-pm1)
  E <- 1-max
  return(E)
}
Gini <- function(pm1){
  G <- 2*pm1*(1-pm1)
  return(G)
}
entropy <- function(pm1){
```

```
  D <- -pm1*log(pm1)-(1-pm1)*log(1-pm1)
  return(D)
}

ggplot()+
  stat_function(fun=errorrate,lty=6,col="red")+
  xlim(0,1)+
  stat_function(fun = Gini,lty=4)+
  xlim(0,1)+
  stat_function(fun = entropy,col="darkblue")+
  xlim(0,1)+
  xlab("pm1")

## Scale for 'x' is already present. Adding another scale for 'x', which
 will
## replace the existing scale.
## Scale for 'x' is already present. Adding another scale for 'x', which
 will
## replace the existing scale.
```

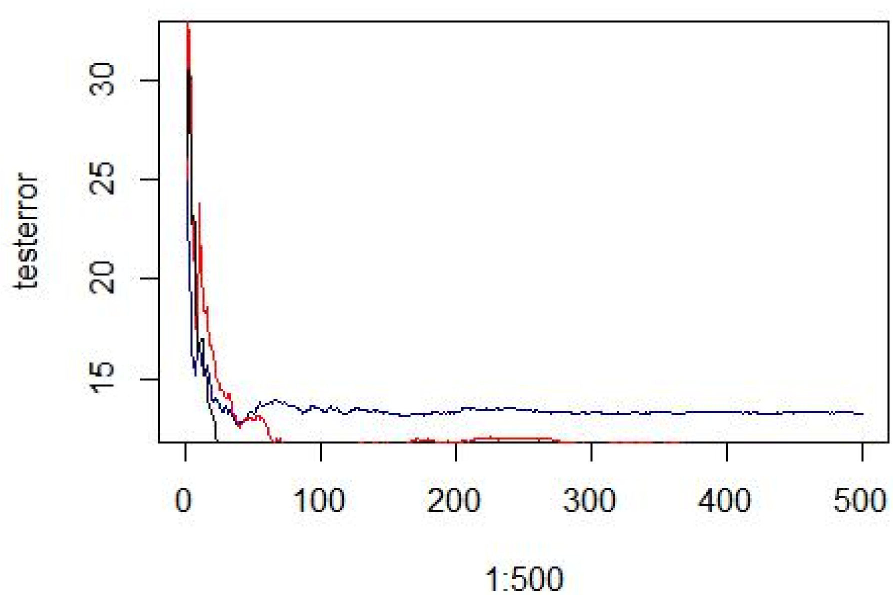


8.5

The left one suggests that the class appears most, whereas the right one suggests that the average of the appearance.

8.7

```r
data("Boston")
train <- sample(1:nrow(Boston),nrow(Boston)/2)
set.seed(1)
rf1 <- randomForest(medv~.,data = Boston,subset = train,mtry=ncol(Boston)-1,importance=T)
rf2 <- randomForest(medv~.,data = Boston,subset = train,mtry=ncol(Boston)/2,importance=T)
rf3 <- randomForest(medv~.,data = Boston,subset = train,mtry=sqrt(ncol(Boston)),importance=T)
plot(1:500,rf1$mse,col="darkblue",type = "l",ylab = "testerror")
lines(1:500,rf2$mse,col="red",type = "l")
lines(1:500,rf3$mse,type = "l")
```



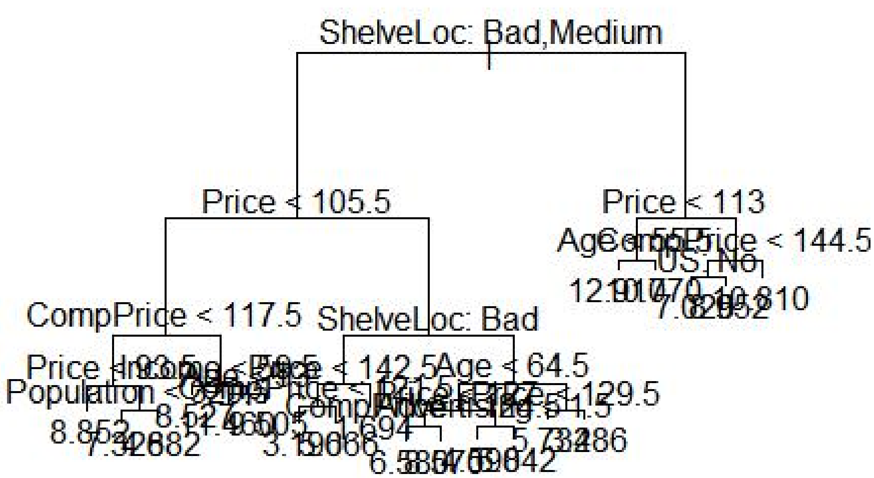
8.8

(a)

```r
data("Carseats")
Carseats$High <- ifelse(Carseats$Sales<=8,"No","Yes")

trainset <- sample(1:nrow(Carseats),nrow(Carseats)/2)
train <- Carseats[trainset,]
test <- Carseats[-trainset,]
```

(b)

```
tree1 <- tree(Sales~.,data = train)
summary(tree1)

##
## Regression tree:
## tree(formula = Sales ~ ., data = train)
## Variables actually used in tree construction:
## [1] "ShelveLoc"   "Price"       "CompPrice"   "Population"  "Income"

## [6] "Age"         "Advertising" "US"
## Number of terminal nodes:  20
## Residual mean deviance:  1.826 = 328.7 / 180
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -4.1440 -0.9042  0.1647  0.0000  0.7964  3.2580

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



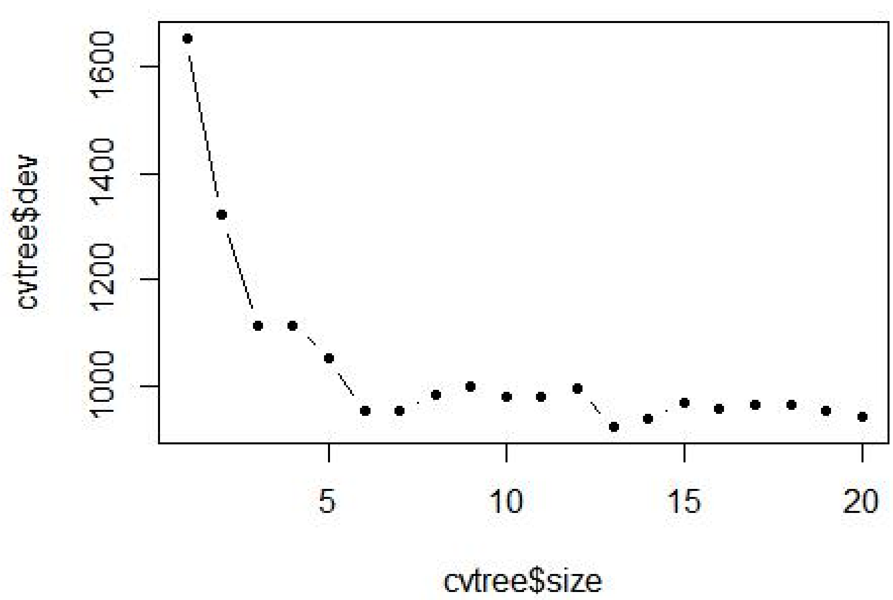

```
yhat <- predict(tree1,newdata = test)
y <- test[,"Sales"]
mean((yhat-y)^2)

## [1] 5.312313
```

(c)

```
cvtree <- cv.tree(tree1)
summary(cvtree)
```

```
##        Length Class  Mode
## size   20     -none- numeric
## dev    20     -none- numeric
## k      20     -none- numeric
## method  1     -none- character
```

```
plot(cvtree$size,cvtree$dev,type = 'b',pch=20)
```




```
prune <- prune.tree(tree1,best = 15)

yhat <- predict(prune,newdata = test)
y <- test[,"Sales"]
mean((yhat-y)^2)
```

```
## [1] 5.881498
```

(d)

```
set.seed(1)
bag <- randomForest(Sales~.,data = train,mtry=11,importance=T)

yhat <- predict(bag,newdata = test)
y <- test[,"Sales"]
mean((yhat-y)^2)
```

```
## [1] 2.046747

importance(bag)

##                %IncMSE IncNodePurity
## CompPrice    5.3376162     82.577219
## Income      -0.8785209     37.794006
## Advertising  3.1977266     41.330555
## Population  -0.7102277     47.981835
## Price       21.0112129    124.640615
## ShelveLoc   23.4248611     90.861156
## Age          3.7656413     70.607100
## Education   -1.0579893     31.051055
## Urban       -3.1100704      4.580155
## US           3.1322490      5.296530
## High        84.4521601   1036.120281
```

(e)

```
set.seed(1)
rf <- randomForest(Sales~.,data = train,mtry=3,importance=T)

yhat <- predict(rf,newdata = test)
y <- test[,"Sales"]
mean((yhat-y)^2)

## [1] 1.868757

importance(rf)

##                %IncMSE IncNodePurity
## CompPrice    7.5468056      96.47039
## Income       0.1316241      68.09053
## Advertising  5.8893159      91.81305
## Population   0.9819746      71.78220
## Price       17.7178407     219.04966
## ShelveLoc   19.6995728     196.63083
## Age          6.6400824      98.55016
## Education    0.8340787      46.48212
## Urban        0.2099158      10.01687
## US           1.8698812      12.87469
## High        50.9487507     613.52441
```

8.11

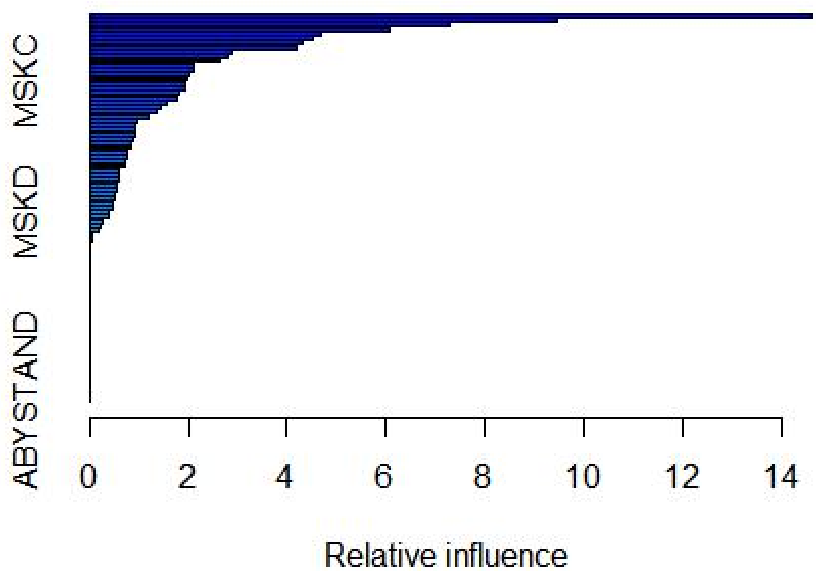
(a)

```
data("Caravan")
Caravan$Purchase <- as.character(Caravan$Purchase)
Caravan$Purchase[Caravan$Purchase=="No"] <- 0
Caravan$Purchase[Caravan$Purchase=="Yes"] <- 1
```

```r
train <- Caravan[1:1000,]
test <- Caravan[1001:nrow(Caravan),]
```

(b)

```r
set.seed(1)

boost <- gbm(Purchase~.,data = train,distribution = "bernoulli",n.trees
= 1000,shrinkage = .01)
summary(boost)
```




```
##                 var     rel.inf
## PPERSAUT PPERSAUT 14.63504779
## MKOOPKLA MKOOPKLA  9.47091649
## MOPLHOOG MOPLHOOG  7.31457416
## MBERMIDD MBERMIDD  6.08651965
## PBRAND     PBRAND  4.66766122
## MGODGE     MGODGE  4.49463264
## ABRAND     ABRAND  4.32427755
## MINK3045 MINK3045  4.17590619
## MOSTYPE   MOSTYPE  2.86402583
## PWAPART   PWAPART  2.78191075
## MAUT1       MAUT1  2.61929152
## MBERARBG MBERARBG  2.10480508
## MSKA         MSKA  2.10185152
## MAUT2       MAUT2  2.02172510
## MSKC         MSKC  1.98684345
## MINKGEM   MINKGEM  1.92122708
```

```
## MGODPR     MGODPR  1.91777542
## MBERHOOG MBERHOOG  1.80710618
## MGODOV     MGODOV  1.78693913
## PBYSTAND PBYSTAND  1.57279593
## MSKB1       MSKB1  1.43551401
## MFWEKIND MFWEKIND  1.37264255
## MRELGE     MRELGE  1.20805179
## MOPLMIDD MOPLMIDD  0.93791970
## MINK7512 MINK7512  0.92590720
## MINK4575 MINK4575  0.91745993
## MGODRK     MGODRK  0.90765539
## MFGEKIND MFGEKIND  0.85745374
## MZPART     MZPART  0.82531066
## MRELOV     MRELOV  0.80731252
## MINKM30   MINKM30  0.74126812
## MHKOOP     MHKOOP  0.73690793
## MZFONDS   MZFONDS  0.71638323
## MAUT0       MAUT0  0.71388052
## MHHUUR     MHHUUR  0.59287247
## APERSAUT APERSAUT  0.58056986
## MOSHOOFD MOSHOOFD  0.58029563
## MSKB2       MSKB2  0.53885275
## PLEVEN     PLEVEN  0.53052444
## MINK123M MINK123M  0.50660603
## MBERARBO MBERARBO  0.48596479
## MGEMOMV   MGEMOMV  0.47614792
## PMOTSCO   PMOTSCO  0.46163590
## MSKD         MSKD  0.39735297
## MBERBOER MBERBOER  0.36417546
## MGEMLEEF MGEMLEEF  0.26166240
## MFALLEEN MFALLEEN  0.21448118
## MBERZELF MBERZELF  0.15906143
## MOPLLAAG MOPLLAAG  0.05263665
## MAANTHUI MAANTHUI  0.03766014
## MRELSA     MRELSA  0.00000000
## PWABEDR   PWABEDR  0.00000000
## PWALAND   PWALAND  0.00000000
## PBESAUT   PBESAUT  0.00000000
## PVRAAUT   PVRAAUT  0.00000000
## PAANHANG PAANHANG  0.00000000
## PTRACTOR PTRACTOR  0.00000000
## PWERKT     PWERKT  0.00000000
## PBROM       PBROM  0.00000000
## PPERSONG PPERSONG  0.00000000
## PGEZONG   PGEZONG  0.00000000
## PWAOREG   PWAOREG  0.00000000
## PZEILPL   PZEILPL  0.00000000
## PPLEZIER PPLEZIER  0.00000000
## PFIETS     PFIETS  0.00000000
## PINBOED   PINBOED  0.00000000
```

```
## AWAPART   AWAPART  0.00000000
## AWABEDR   AWABEDR  0.00000000
## AWALAND   AWALAND  0.00000000
## ABESAUT   ABESAUT  0.00000000
## AMOTSCO   AMOTSCO  0.00000000
## AVRAAUT   AVRAAUT  0.00000000
## AAANHANG AAANHANG  0.00000000
## ATRACTOR ATRACTOR  0.00000000
## AWERKT     AWERKT  0.00000000
## ABROM       ABROM  0.00000000
## ALEVEN     ALEVEN  0.00000000
## APERSONG APERSONG  0.00000000
## AGEZONG   AGEZONG  0.00000000
## AWAOREG   AWAOREG  0.00000000
## AZEILPL   AZEILPL  0.00000000
## APLEZIER APLEZIER  0.00000000
## AFIETS     AFIETS  0.00000000
## AINBOED   AINBOED  0.00000000
## ABYSTAND ABYSTAND  0.00000000
```

(c)

```
yhat <- predict(boost,newdata = test,n.trees = 1000)
```