STATS 101C hw6

Lucy Shao 304575686

5/31/2017

Q1

```
data<-read.csv("//Users/lucy/Downloads/better2000births.csv",header = T)
dim(data)
```

```
## [1] 1998   21
```

```
#a
set.seed(9876)
index<-sample(nrow(data),1000,rep=F)
train<-data[index,]
dim(train)
```

```
## [1] 1000   21
```

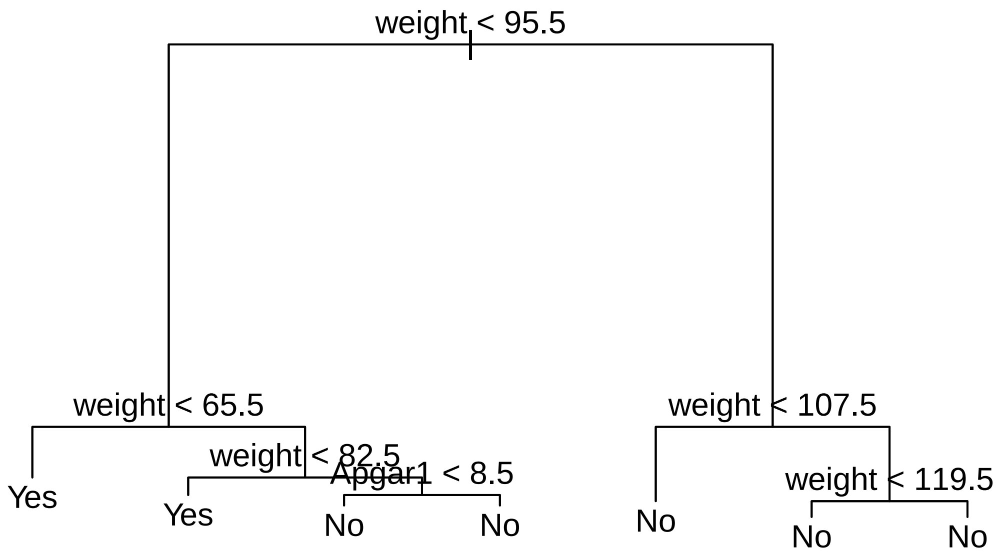
```
test<-data[-index,]
dim(test)
```

```
## [1] 998  21
```

```
library(tree)
model<-tree(factor(Premie)~.,data<-train)
model
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 1000 605.10 No ( 0.910000 0.090000 )
##    2) weight < 95.5 118 161.90 Yes ( 0.440678 0.559322 )
##      4) weight < 65.5 22   0.00 Yes ( 0.000000 1.000000 ) *
##      5) weight > 65.5 96 132.40 No ( 0.541667 0.458333 )
##       10) weight < 82.5 34  42.81 Yes ( 0.323529 0.676471 ) *
##       11) weight > 82.5 62  79.38 No ( 0.661290 0.338710 )
##         22) Apgar1 < 8.5 34  47.02 No ( 0.529412 0.470588 ) *
##         23) Apgar1 > 8.5 28  26.28 No ( 0.821429 0.178571 ) *
##    3) weight > 95.5 882 220.30 No ( 0.972789 0.027211 )
##      6) weight < 107.5 163 117.40 No ( 0.883436 0.116564 ) *
##      7) weight > 107.5 719  59.65 No ( 0.993046 0.006954 )
##       14) weight < 119.5 289  50.48 No ( 0.982699 0.017301 ) *
##       15) weight > 119.5 430   0.00 No ( 1.000000 0.000000 ) *
```

```r
plot(model)
text(model)
```




```r
predict<-predict(model,test,type="class")
head(predict)
```

```
## [1] No No No No No No
## Levels: No Yes
```

```r
summary(model)
```

```
##
## Classification tree:
## tree(formula = factor(Premie) ~ ., data = data <- train)
## Variables actually used in tree construction:
## [1] "weight" "Apgar1"
## Number of terminal nodes:  7
## Residual mean deviance:  0.286 = 283.9 / 993
## Misclassification error rate: 0.056 = 56 / 1000
```

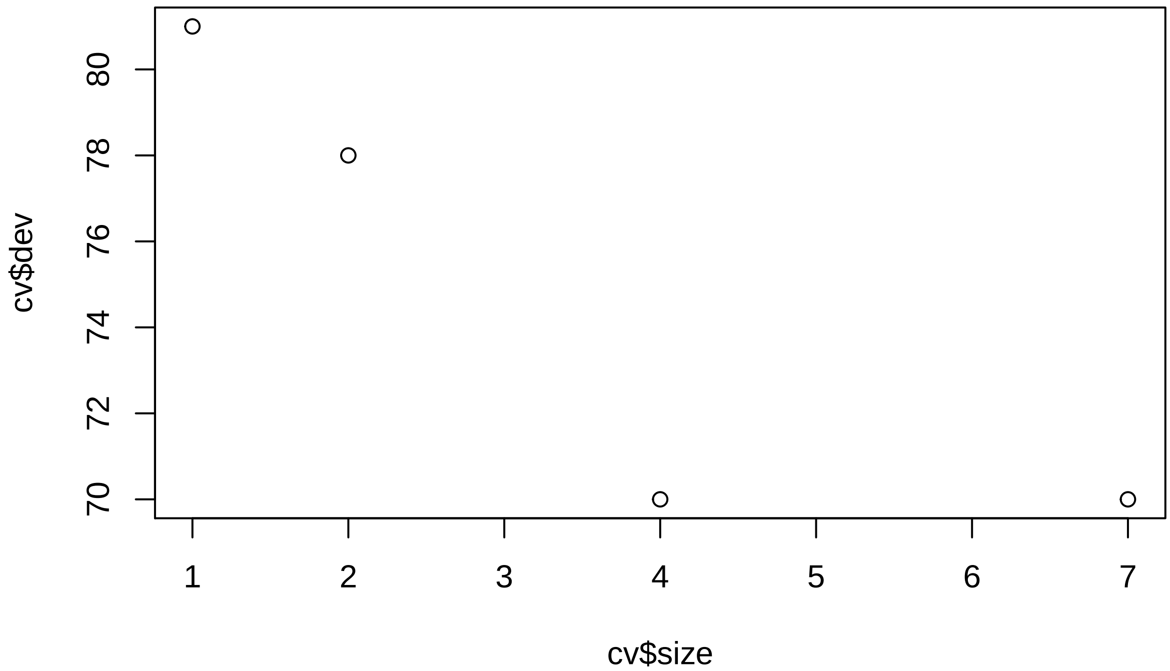
```r
table(test$Premie,predict)
```

```
##      predict
##         No Yes
##   No   903   4
##   Yes   51  40
```

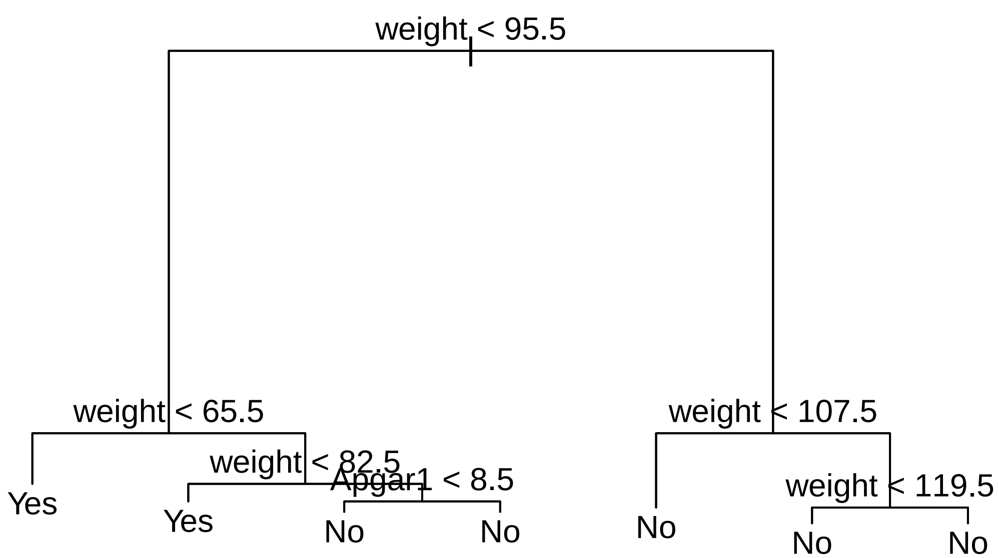
```r
(51+4)/length(predict)
```

```
## [1] 0.05511022
```

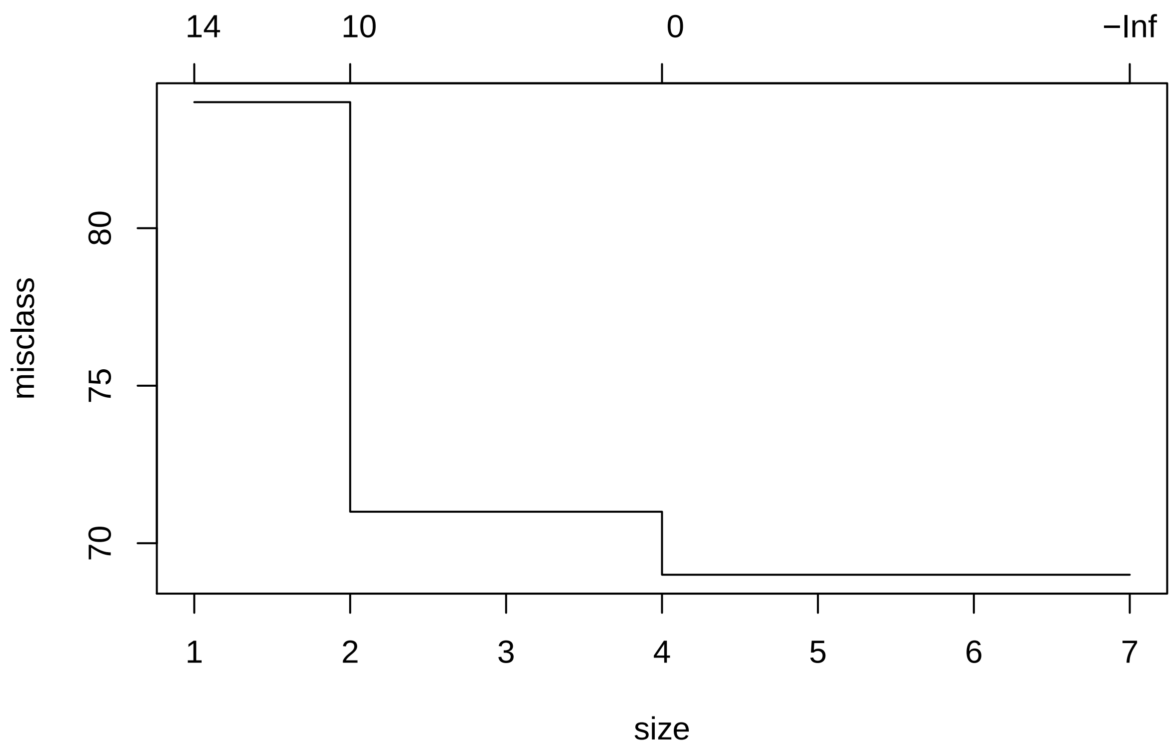
```r
#misclassification error


#b
#cv=cv.tree(model,FUN=prune.tree)
#plot(cv)
cv<-cv.tree(model,FUN=prune.misclass)
plot(cv$size,cv$dev)
```




```r
pruned<-prune.misclass(model,best=6) #6 or?
plot(pruned)
text(pruned)
```



```r
cv<-cv.tree(model,FUN=prune.misclass, K=10)
plot(cv) ###four should be the best one
```




```r
predict2<-predict(pruned,test,type="class")
table(test$Premie,predict2)
```

```
##      predict2
##         No Yes
##   No   903   4
##   Yes   51  40
```

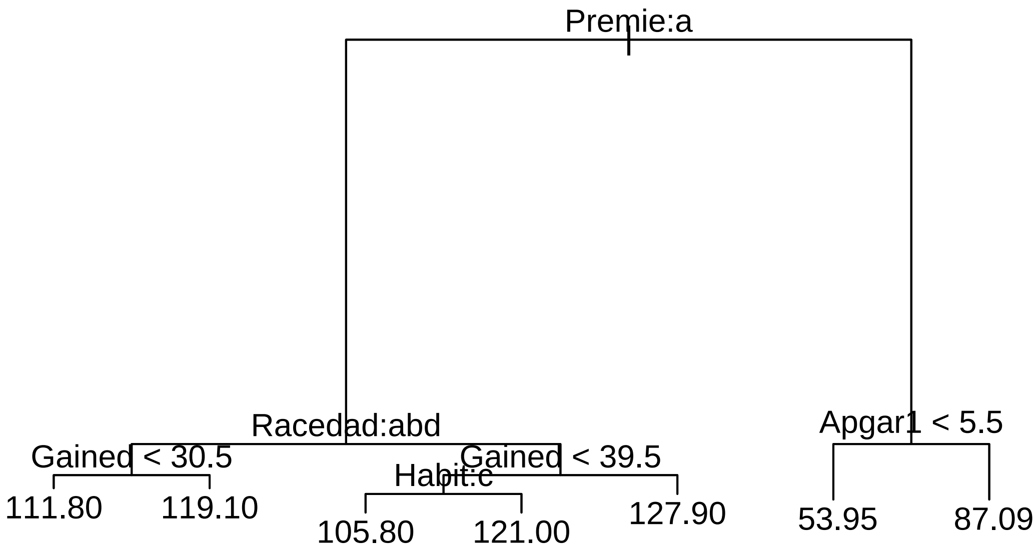
```r
#c Interpret your pruned tree (or your tree in (a) if you did not need to prune). In particular, does i

#weight and Apgar1 are associate with premature births


#d
#did better, with 6%
```

2

```r
nrow(train)
```

```
## [1] 1000
```

```r
model<-tree(weight~.,data = train)
plot(model)
text(model)
```




```r
predict2<-predict(model,test,type="vector")
head(predict2)
```

```
##        4        5        7        8       11       12
## 127.9302 119.0780 121.0215 111.8083 121.0215 121.0215
```

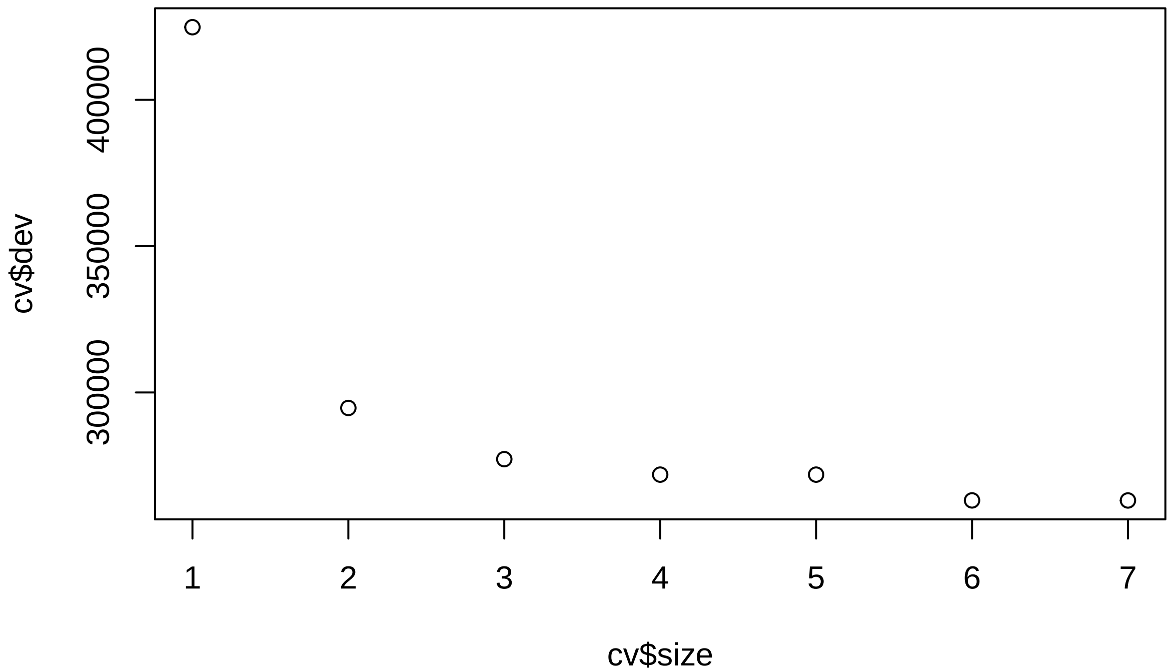
```r
MSE<-mean((as.numeric(predict2) - test$weight)^2)
MSE
```

```
## [1] 271.7413
```

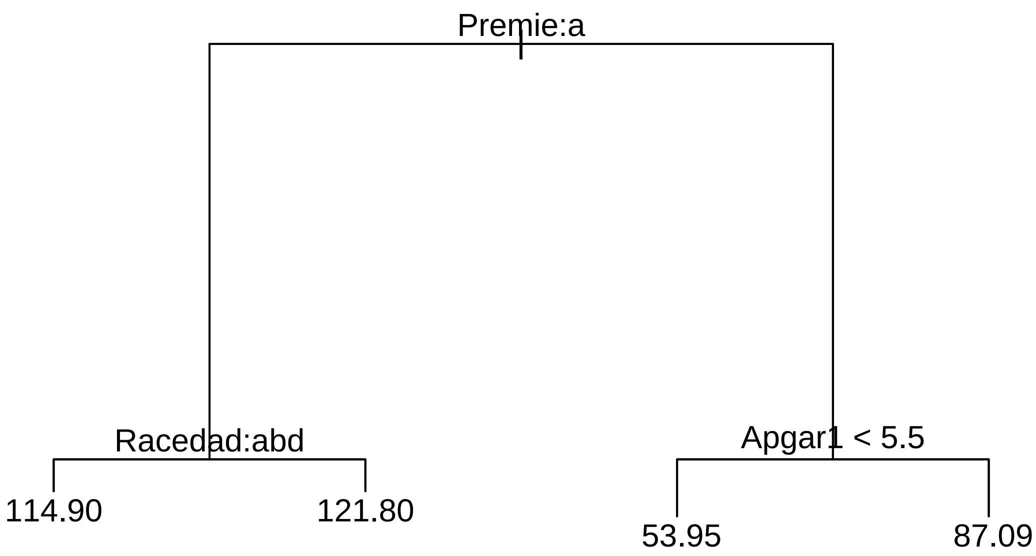
```r
#b
cv<-cv.tree(model)
cv$dev
```

```
## [1] 263094.4 263108.9 271923.0 271923.0 277208.2 294698.4 424818.0
```

```r
plot(cv$size,cv$dev)
```




```r
pruned<-prune.tree(model,best=4)
plot(pruned)
text(pruned)
```




```r
#c
predict3<-predict(pruned,test)
MSE<-mean((as.numeric(predict3)-test$weight)^2)
MSE
```

```
## [1] 274.0873
```

```
# MSE decreases
```