9.2

代码：

*#读取数据*

*setwd("C:\\Users\\Mac\\Desktop\\过程\\学业\\本科\\专业课\\多元统计\\CART\\data")*

*CHD<-read.table("cleveland.txt", header=T)*

*CHD<-CHD[, -15]*

*#探索age的最佳split*

*CHD2<-cbind(CHD$age, CHD$diag)*

*Age<-sort(unique(CHD2[,1]))*

*T<-length(Age)*

*n<-matrix(nrow=2, ncol=2)*

*tau\_L<-numeric(T)*

*tau\_R<-numeric(T)*

*delta<-numeric(T)*

*delta\_max<-0*

*threshold\_best<-0*

*for (i in 1:T)*

*{*

*#当前分法下的左右节点*

*CHD2\_left<-matrix(CHD2[CHD2[,1]<=Age[i], ],ncol=2)*

*CHD2\_right<-matrix(CHD2[CHD2[,1]>Age[i], ],ncol=2)*

*#左右节点和分类列联表*

*n[1, 1]<-sum(CHD2\_left[, 2]==1)*

*n[1, 2]<-sum(CHD2\_left[, 2]==2)*

*n[2, 1]<-sum(CHD2\_right[, 2]==1)*

*n[2, 2]<-sum(CHD2\_right[, 2]==2)*

*n\_row<-rowSums(n)*

*n\_col<-colSums(n)*

*n\_sum<-sum(n)*

*#根节点和左右节点的基尼系数*

*root <- 1-(n\_col[1]/n\_sum)^2-(n\_col[2]/n\_sum)^2*

*tau\_L[i] <- 1-(n[1,1]/n\_row[1])^2-(n[1,2]/n\_row[1])^2*

*tau\_R[i] <- 1-(n[2,1]/n\_row[2])^2-(n[2,2]/n\_row[2])^2*

*#当前分法的减少基尼系数*

*delta[i] <- root-n\_row[1]/n\_sum\*tau\_L[i]-n\_row[2]/n\_sum\*tau\_R[i]*

*#找出最好的split*

*if (is.na(delta[i])) {delta[i]<-0}*

*if (delta[i]> delta\_max)*

*{*

*delta\_max<-delta[i]*

*split\_best<-n*

*threshold\_best<-Age[i]*

*}*

*}*

*#最佳的split阈值及其效果*

*split\_best*

*threshold\_best*

*par(mfrow=c(1,2))*

*#左右节点基尼系数图*

*plot(Age, tau\_R, type='l', col='red', xlab="Age at Split", ylab="i(tau)")*

*points(Age, tau\_L, type='l', col="blue")*

*legend("bottom", c("left", "right"), col=c('blue', 'red'), lty=c(1,1), bty='n')*

*#减少基尼系数图*

*plot(Age, delta, type='l', col='red', xlab="Age at Split", ylab="Goodness of Split")*

*#定义一个计算最佳阈值的函数*

*tree <-function(column)*

*{*

*xy<-cbind(CHD[[column]], CHD$diag)*

*x\_unique<-sort(unique(xy[,1]))*

*T<-length(x\_unique)*

*delta\_max<-0*

*for (i in 1:T)*

*{*

*#当前分法下的左右节点*

*xy\_left<-matrix(xy[xy[,1]<=x\_unique[i], ],ncol=2)*

*xy\_right<-matrix(xy[xy[,1]>x\_unique[i], ],ncol=2)*

*#左右节点和分类列联表*

*n[1, 1]<-sum(xy\_left[, 2]==1)*

*n[1, 2]<-sum(xy\_left[, 2]==2)*

*n[2, 1]<-sum(xy\_right[, 2]==1)*

*n[2, 2]<-sum(xy\_right[, 2]==2)*

*n\_row<-rowSums(n)*

*n\_col<-colSums(n)*

*n\_sum<-sum(n)*

*#根节点和左右节点的基尼系数*

*root <- 1-(n\_col[1]/n\_sum)^2-(n\_col[2]/n\_sum)^2*

*tau\_L <- 1-(n[1,1]/n\_row[1])^2-(n[1,2]/n\_row[1])^2*

*tau\_R <- 1-(n[2,1]/n\_row[2])^2-(n[2,2]/n\_row[2])^2*

*#当前分法的减少基尼系数*

*delta <- root-n\_row[1]/n\_sum\*tau\_L-n\_row[2]/n\_sum\*tau\_R*

*#找出最好的split*

*if (is.na(delta)) {delta <- 0}*

*if (delta > delta\_max) {delta\_max<-delta}*

*}*

*print(column)*

*print(delta\_max)*

*}*

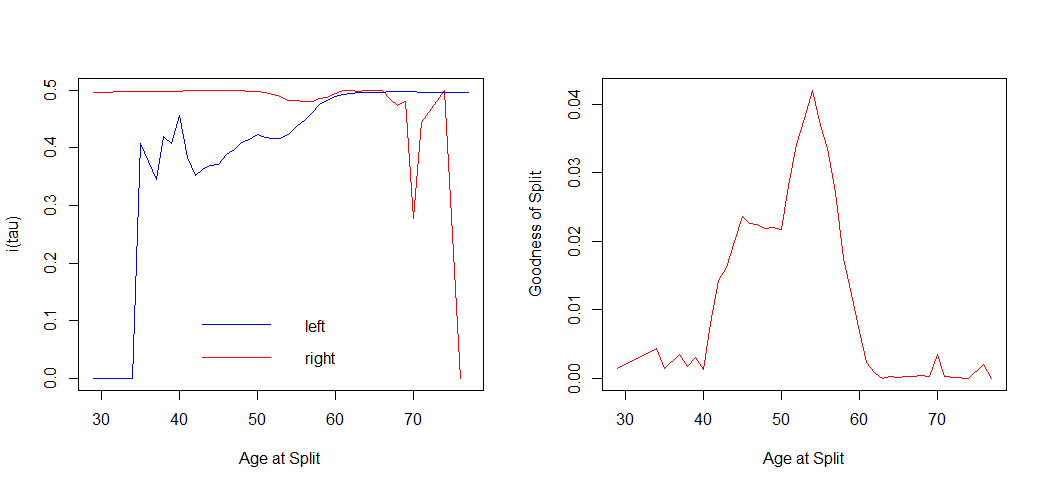
*#打印变量名及最佳阈值*

*for (col in colnames(CHD)) {if (col != 'diag') {tree(col)}}*

结果如下：

*the best split on the variable “age”*

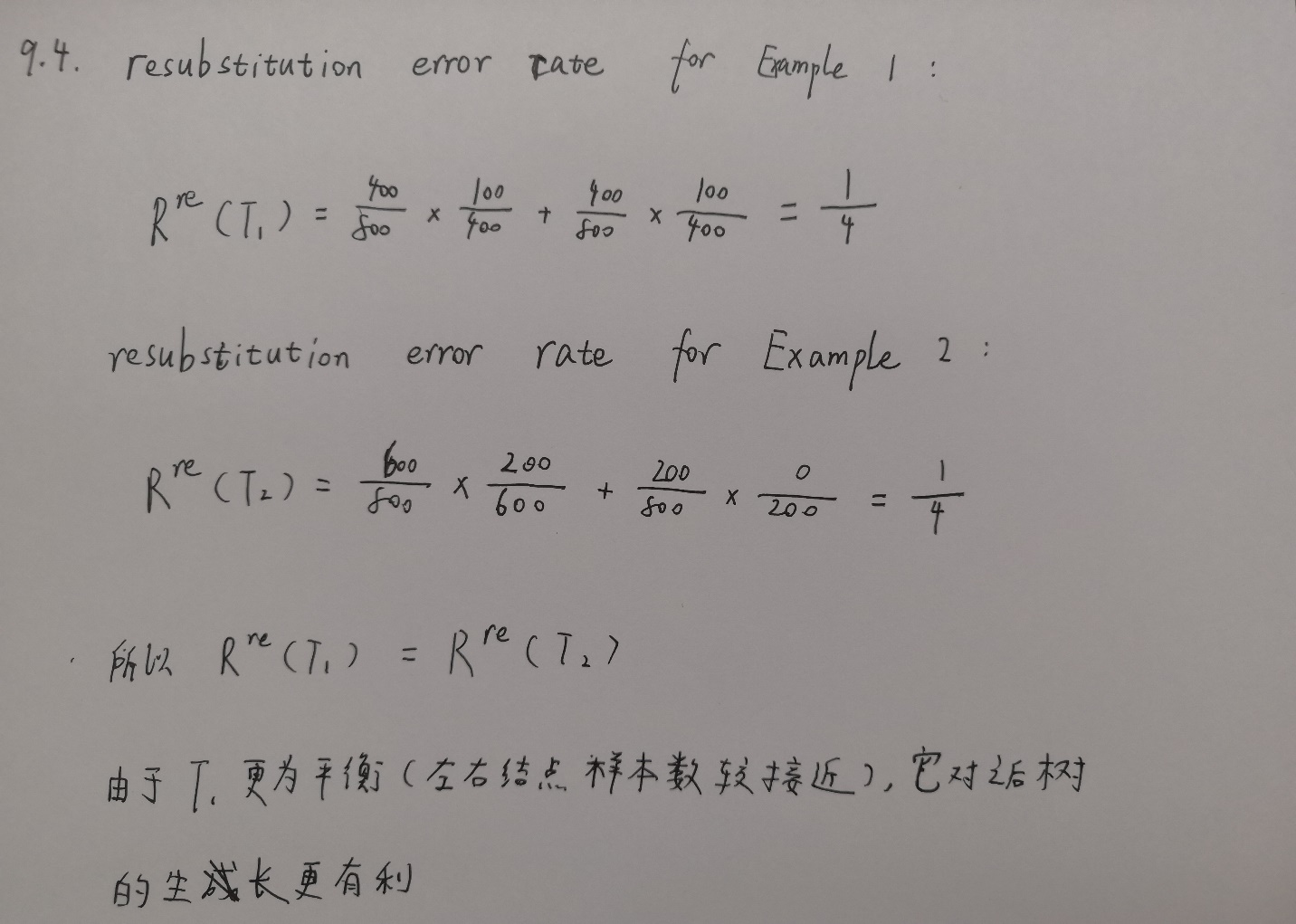
|  |  |  |  |
| --- | --- | --- | --- |
|  | Buff | Sick | Row Total |
| Age<=54 | 96 | 42 | 138 |
| Age>54 | 64 | 94 | 158 |
| Column Total | 160 | 136 | 296 |



*Determination of the best split*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| age | gender | cp | trestbps | chol | fbs | restecg |
| 0.042 | 0.040 | 0.046 | 0.010 | 0.011 | 1.088e-05 | 0.014 |
| thatach | **exang** | **oldpeak** | **slope** | **ca** | **thal** |  |
| 0.090 | 0.090 | 0.082 | 0.074 | 0.118 | 0.119 |  |

9.4



9.8

代码：

*#读取数据*

*setwd("C:\\Users\\Mac\\Desktop\\过程\\学业\\本科\\专业课\\多元统计\\CART\\data")*

*vehicle<-read.table("vehicle3.txt", header=T)*

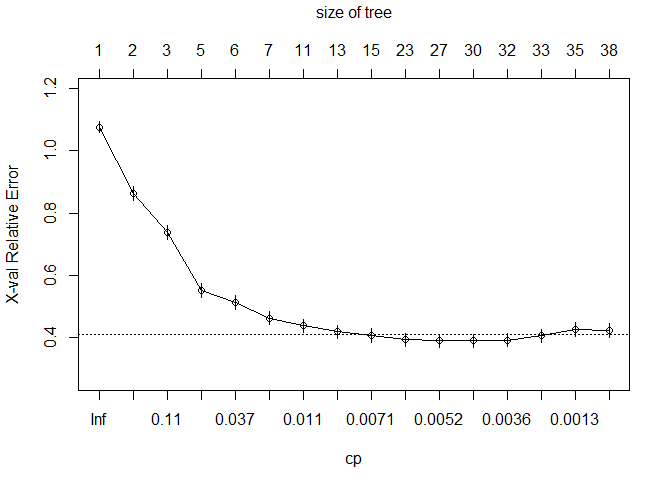
*vehicle$class<-factor(vehicle$class)*

*#绘制阿尔法&树大小-误判率图*

*library(rpart, rpart.plot)*

*out<-rpart(class~.-pam, vehicle, cp=1e-10)*

*plotcp(out)*



#cptable

out$cptable

CP nsplit rel error xerror xstd

1 0.2054140127 0 1.0000000 1.0764331 0.01855893

2 0.1210191083 1 0.7945860 0.8646497 0.02220631

3 0.0955414013 2 0.6735669 0.7388535 0.02304867

4 0.0509554140 4 0.4824841 0.5525478 0.02278086

5 0.0270700637 5 0.4315287 0.5143312 0.02250128

6 0.0127388535 6 0.4044586 0.4633758 0.02200129

7 0.0087579618 10 0.3535032 0.4378981 0.02169400

8 0.0079617834 12 0.3359873 0.4187898 0.02143716

9 0.0063694268 14 0.3200637 0.4076433 0.02127655

10 0.0055732484 22 0.2579618 0.3933121 0.02105800

11 0.0047770701 26 0.2356688 0.3901274 0.02100755

12 0.0039808917 29 0.2213376 0.3901274 0.02100755

13 0.0031847134 31 0.2133758 0.3917197 0.02103286

14 0.0015923567 32 0.2101911 0.4060510 0.02125294

15 0.0010615711 34 0.2070064 0.4267516 0.02154698

16 0.0000000001 37 0.2038217 0.4235669 0.02150354

#选择最佳的阿尔法，进行剪枝

out.prune<-prune(out, cp=out$cptable[9,1])