install.packages("rmarkdown",repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/tswaminathan/Documents/R/win-library/3.5'

## (as 'lib' is unspecified)

## package 'rmarkdown' successfully unpacked and MD5 sums checked

##

## The downloaded binary packages are in

## C:\Users\tswaminathan\AppData\Local\Temp\Rtmp2xR758\downloaded\_packages

install.packages("C50",repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/tswaminathan/Documents/R/win-library/3.5'

## (as 'lib' is unspecified)

## package 'C50' successfully unpacked and MD5 sums checked

##

## The downloaded binary packages are in

## C:\Users\tswaminathan\AppData\Local\Temp\Rtmp2xR758\downloaded\_packages

install.packages("tree",repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/tswaminathan/Documents/R/win-library/3.5'

## (as 'lib' is unspecified)

## package 'tree' successfully unpacked and MD5 sums checked

##

## The downloaded binary packages are in

## C:\Users\tswaminathan\AppData\Local\Temp\Rtmp2xR758\downloaded\_packages

install.packages("caret",repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/tswaminathan/Documents/R/win-library/3.5'

## (as 'lib' is unspecified)

## package 'caret' successfully unpacked and MD5 sums checked

##

## The downloaded binary packages are in

## C:\Users\tswaminathan\AppData\Local\Temp\Rtmp2xR758\downloaded\_packages

install.packages("gmodels",repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/tswaminathan/Documents/R/win-library/3.5'

## (as 'lib' is unspecified)

## package 'gmodels' successfully unpacked and MD5 sums checked

##

## The downloaded binary packages are in

## C:\Users\tswaminathan\AppData\Local\Temp\Rtmp2xR758\downloaded\_packages

install.packages("party",repos = "http://cran.us.r-project.org")

## Installing package into 'C:/Users/tswaminathan/Documents/R/win-library/3.5'

## (as 'lib' is unspecified)

## package 'party' successfully unpacked and MD5 sums checked

##

## The downloaded binary packages are in

## C:\Users\tswaminathan\AppData\Local\Temp\Rtmp2xR758\downloaded\_packages

**library**(party)

## Loading required package: grid

## Loading required package: mvtnorm

## Loading required package: modeltools

## Loading required package: stats4

## Loading required package: strucchange

## Loading required package: zoo

##

## Attaching package: 'zoo'

## The following objects are masked from 'package:base':

##

## as.Date, as.Date.numeric

## Loading required package: sandwich

**library**(caret)

## Loading required package: lattice

## Loading required package: ggplot2

**library**(C50)

**library**(tree)

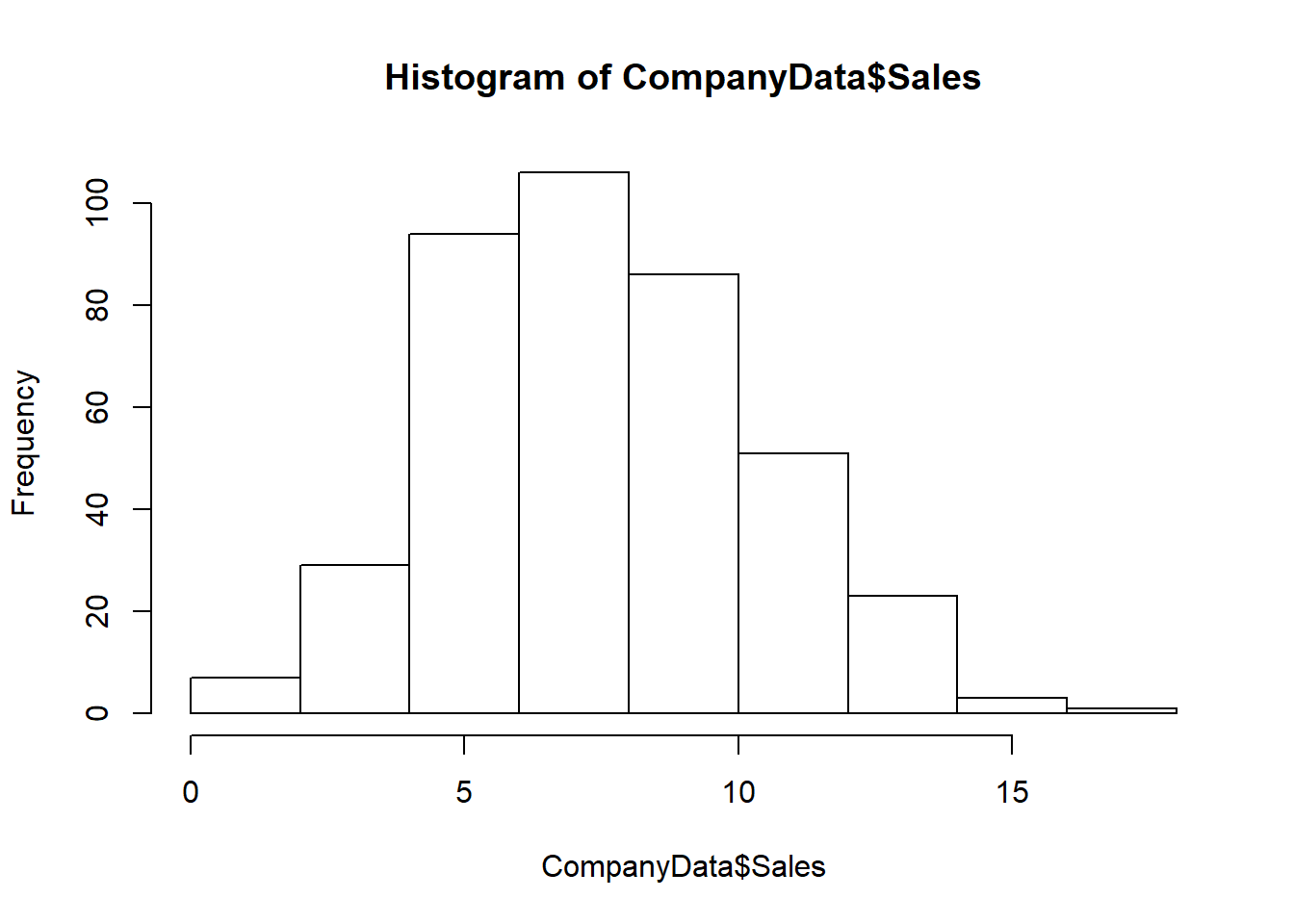
**library**(gmodels)

CompanyData <- read.csv(file.choose())

*# Splitting data into training and testing.*

*# splitting the data based on Sales*

hist(CompanyData$Sales)



High = ifelse(CompanyData$Sales<10, "No", "Yes")

CD = data.frame(CompanyData, High)

*#CD <- CompanyData[,2:12]*

*# View(CD)*

CD\_train <- CD[1:200,]

*# View(CD\_train)*

CD\_test <- CD[201:400,]

*# View(CD\_test)*

*#Using Party Function*

op\_tree = ctree(High ~ CompPrice + Income + Advertising + Population + Price + ShelveLoc

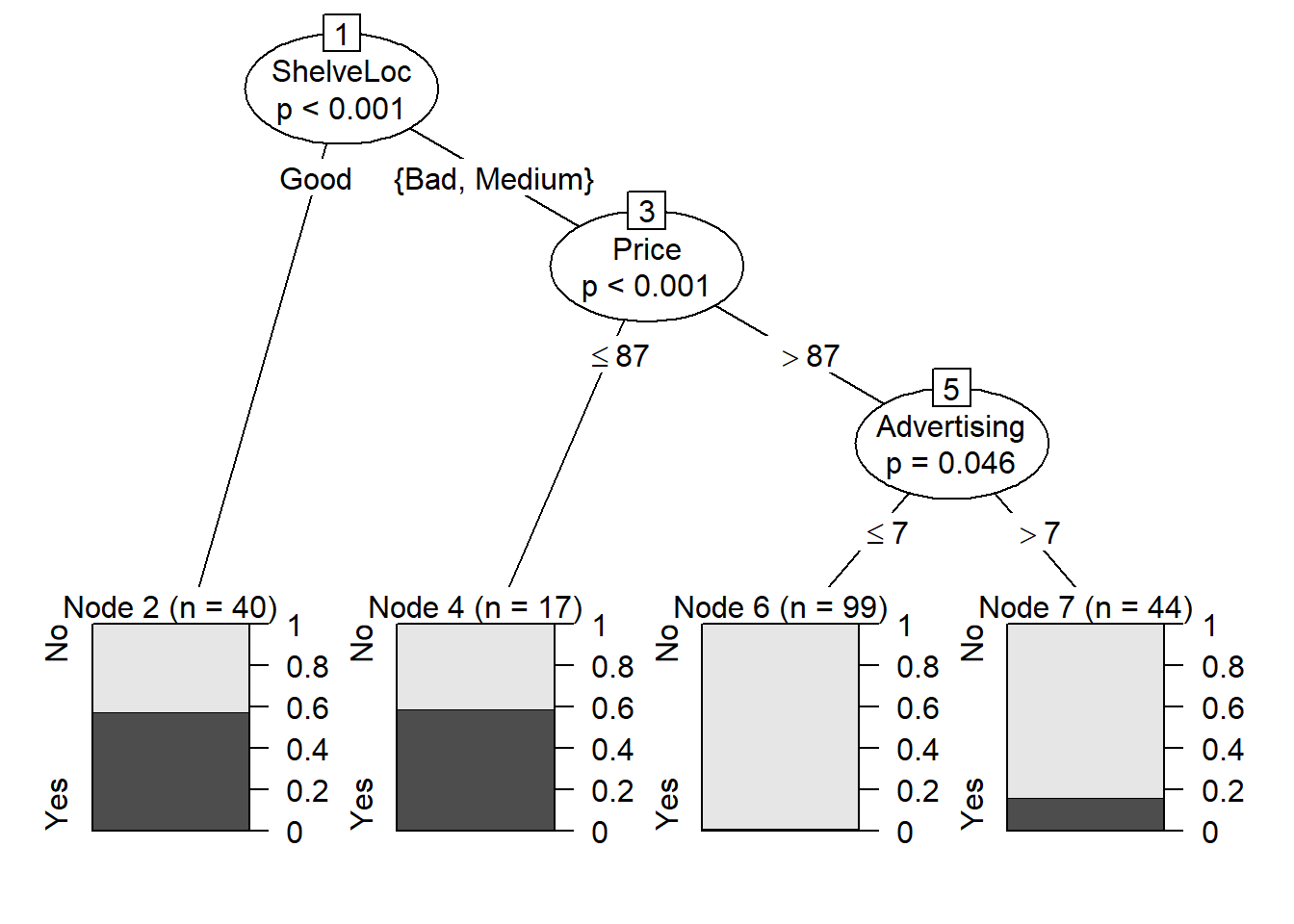
+ Age + Education + Urban + US, data = CD\_train)

summary(op\_tree)

## Length Class Mode

## 1 BinaryTree S4

plot(op\_tree)



*# On looking into the Above tree, i see that if the Location of the Shelv is good,*

*# then there is a probability of 60% chance that the customer will buy.*

*# With ShelveLoc having a Bad or Medium and Price <= 87, the probability of High sales*

*# could be 60%.*

*# If ShelveLoc is Bad or Medium, With Price >= 87 and Advertising less then <= 7 then there*

*# is a zero percent chance of high sales.*

*# If ShelveLoc is Bad or Medium, With Price >= 87 and Advertising less then > 7 then there*

*# is a 20 % percent chance of high sales.*

pred\_tree <- as.data.frame(predict(op\_tree,newdata=CD\_test))

pred\_tree["final"] <- NULL

pred\_test\_df <- predict(op\_tree,newdata=CD\_test)

mean(pred\_test\_df==CD$High) *# Accuracy = 68.75%*

## [1] 0.6875

CrossTable(CD\_test$High,pred\_test\_df)

##

##

## Cell Contents

## |-------------------------|

## | N |

## | Chi-square contribution |

## | N / Row Total |

## | N / Col Total |

## | N / Table Total |

## |-------------------------|

##

##

## Total Observations in Table: 200

##

##

## | pred\_test\_df

## CD\_test$High | No | Yes | Row Total |

## -------------|-----------|-----------|-----------|

## No | 131 | 31 | 162 |

## | 2.468 | 5.899 | |

## | 0.809 | 0.191 | 0.810 |

## | 0.929 | 0.525 | |

## | 0.655 | 0.155 | |

## -------------|-----------|-----------|-----------|

## Yes | 10 | 28 | 38 |

## | 10.523 | 25.148 | |

## | 0.263 | 0.737 | 0.190 |

## | 0.071 | 0.475 | |

## | 0.050 | 0.140 | |

## -------------|-----------|-----------|-----------|

## Column Total | 141 | 59 | 200 |

## | 0.705 | 0.295 | |

## -------------|-----------|-----------|-----------|

##

##

confusionMatrix(CD\_test$High,pred\_test\_df)

## Confusion Matrix and Statistics

##

## Reference

## Prediction No Yes

## No 131 31

## Yes 10 28

##

## Accuracy : 0.795

## 95% CI : (0.7323, 0.8487)

## No Information Rate : 0.705

## P-Value [Acc > NIR] : 0.002590

##

## Kappa : 0.4503

## Mcnemar's Test P-Value : 0.001787

##

## Sensitivity : 0.9291

## Specificity : 0.4746

## Pos Pred Value : 0.8086

## Neg Pred Value : 0.7368

## Prevalence : 0.7050

## Detection Rate : 0.6550

## Detection Prevalence : 0.8100

## Balanced Accuracy : 0.7018

##

## 'Positive' Class : No

##

*##### Using tree function*

cd\_tree\_org <- tree(High~.-Sales,data=CD)

summary(cd\_tree\_org)

##

## Classification tree:

## tree(formula = High ~ . - Sales, data = CD)

## Variables actually used in tree construction:

## [1] "ShelveLoc" "Price" "Advertising" "Age" "CompPrice"

## [6] "Population" "Income"

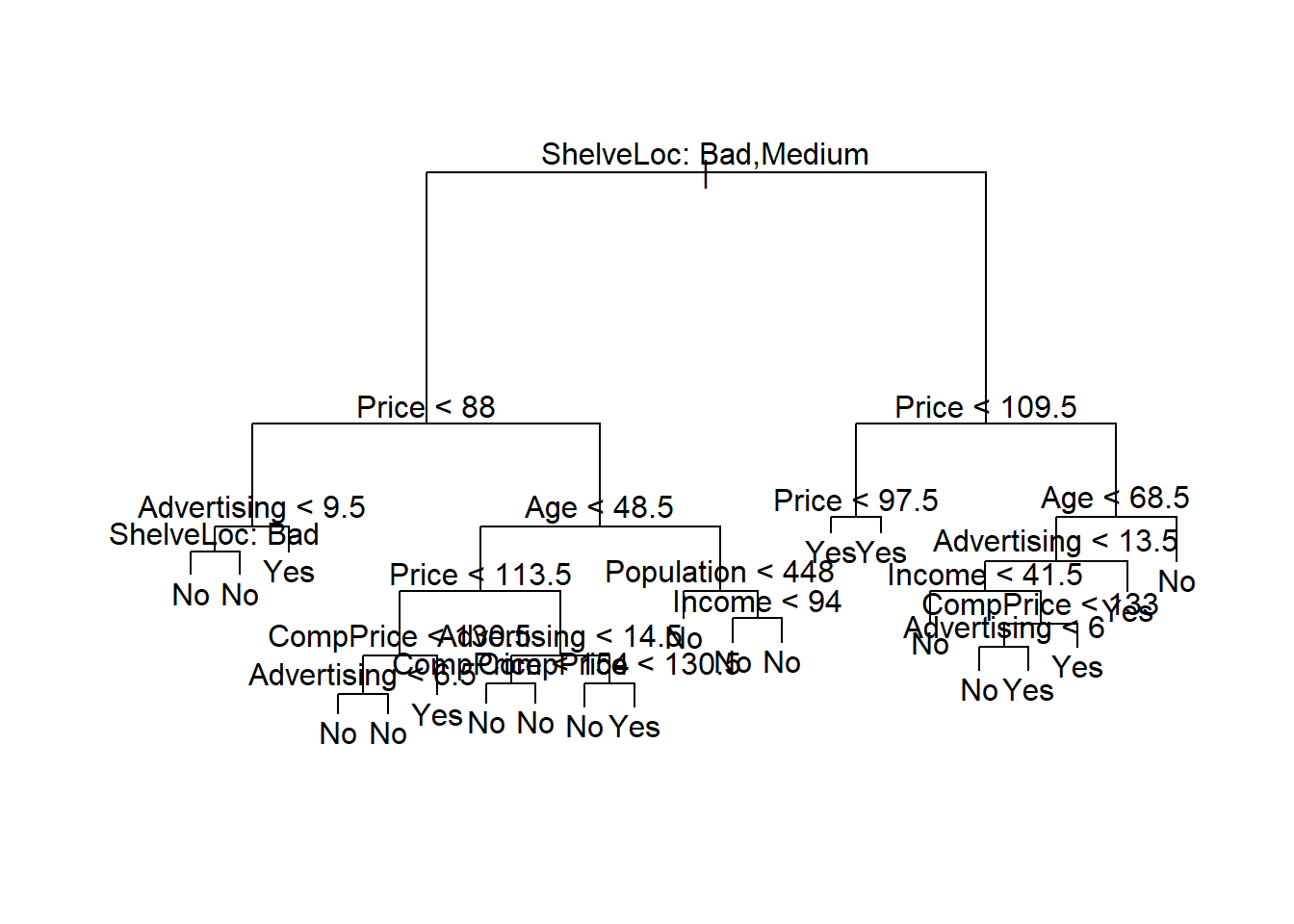
## Number of terminal nodes: 21

## Residual mean deviance: 0.297 = 112.6 / 379

## Misclassification error rate: 0.0725 = 29 / 400

plot(cd\_tree\_org)

text(cd\_tree\_org,pretty = 0)



*# Using the training data*

*##### Using tree function*

cd\_tree <- tree(High~.-Sales,data=CD\_train)

summary(cd\_tree)

##

## Classification tree:

## tree(formula = High ~ . - Sales, data = CD\_train)

## Variables actually used in tree construction:

## [1] "ShelveLoc" "Price" "Advertising" "Age" "CompPrice"

## [6] "Income"

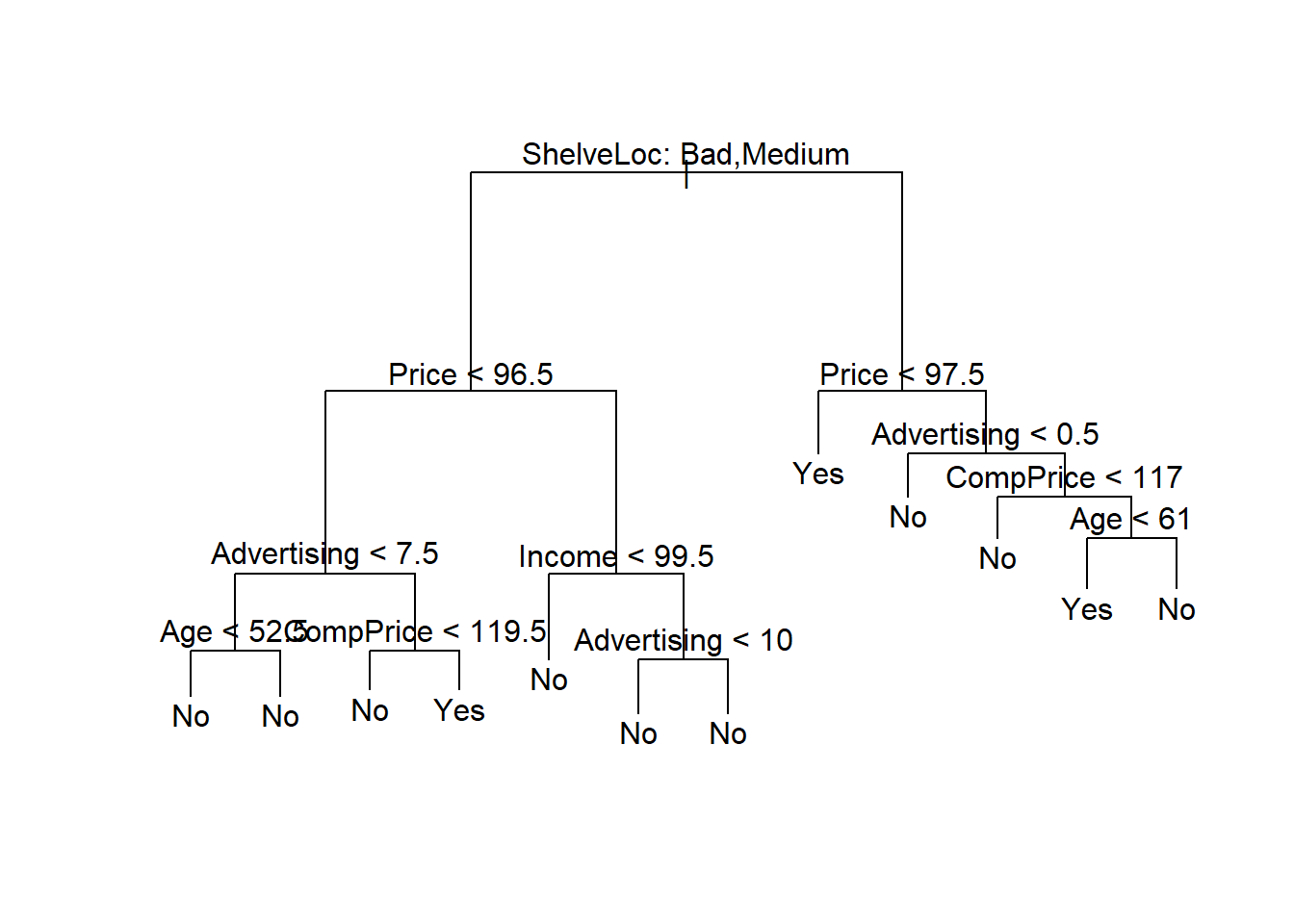
## Number of terminal nodes: 12

## Residual mean deviance: 0.2927 = 55.02 / 188

## Misclassification error rate: 0.08 = 16 / 200

plot(cd\_tree)

text(cd\_tree,pretty = 0)



*### Evaluate the Model*

*# Predicting the test data using the model*

pred\_tree <- as.data.frame(predict(cd\_tree,newdata=CD\_test))

pred\_tree["final"] <- NULL

pred\_test\_df <- predict(cd\_tree,newdata=CD\_test)

pred\_tree$final <- colnames(pred\_test\_df)[apply(pred\_test\_df,1,which.max)]

pred\_tree$final <- as.factor(pred\_tree$final)

summary(pred\_tree$final)

## No Yes

## 172 28

summary(CD\_test$High)

## No Yes

## 162 38

mean(pred\_tree$final==CD$High) *# Accuracy = 77.25*

## [1] 0.7725

CrossTable(CD\_test$High,pred\_tree$final)

##

##

## Cell Contents

## |-------------------------|

## | N |

## | Chi-square contribution |

## | N / Row Total |

## | N / Col Total |

## | N / Table Total |

## |-------------------------|

##

##

## Total Observations in Table: 200

##

##

## | pred\_tree$final

## CD\_test$High | No | Yes | Row Total |

## -------------|-----------|-----------|-----------|

## No | 153 | 9 | 162 |

## | 1.343 | 8.251 | |

## | 0.944 | 0.056 | 0.810 |

## | 0.890 | 0.321 | |

## | 0.765 | 0.045 | |

## -------------|-----------|-----------|-----------|

## Yes | 19 | 19 | 38 |

## | 5.727 | 35.177 | |

## | 0.500 | 0.500 | 0.190 |

## | 0.110 | 0.679 | |

## | 0.095 | 0.095 | |

## -------------|-----------|-----------|-----------|

## Column Total | 172 | 28 | 200 |

## | 0.860 | 0.140 | |

## -------------|-----------|-----------|-----------|

##

##

confusionMatrix(CD\_test$High,pred\_tree$final)

## Confusion Matrix and Statistics

##

## Reference

## Prediction No Yes

## No 153 9

## Yes 19 19

##

## Accuracy : 0.86

## 95% CI : (0.8041, 0.9049)

## No Information Rate : 0.86

## P-Value [Acc > NIR] : 0.55018

##

## Kappa : 0.4942

## Mcnemar's Test P-Value : 0.08897

##

## Sensitivity : 0.8895

## Specificity : 0.6786

## Pos Pred Value : 0.9444

## Neg Pred Value : 0.5000

## Prevalence : 0.8600

## Detection Rate : 0.7650

## Detection Prevalence : 0.8100

## Balanced Accuracy : 0.7841

##

## 'Positive' Class : No

##