Assignment2

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(dplyr)
library(rpart)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

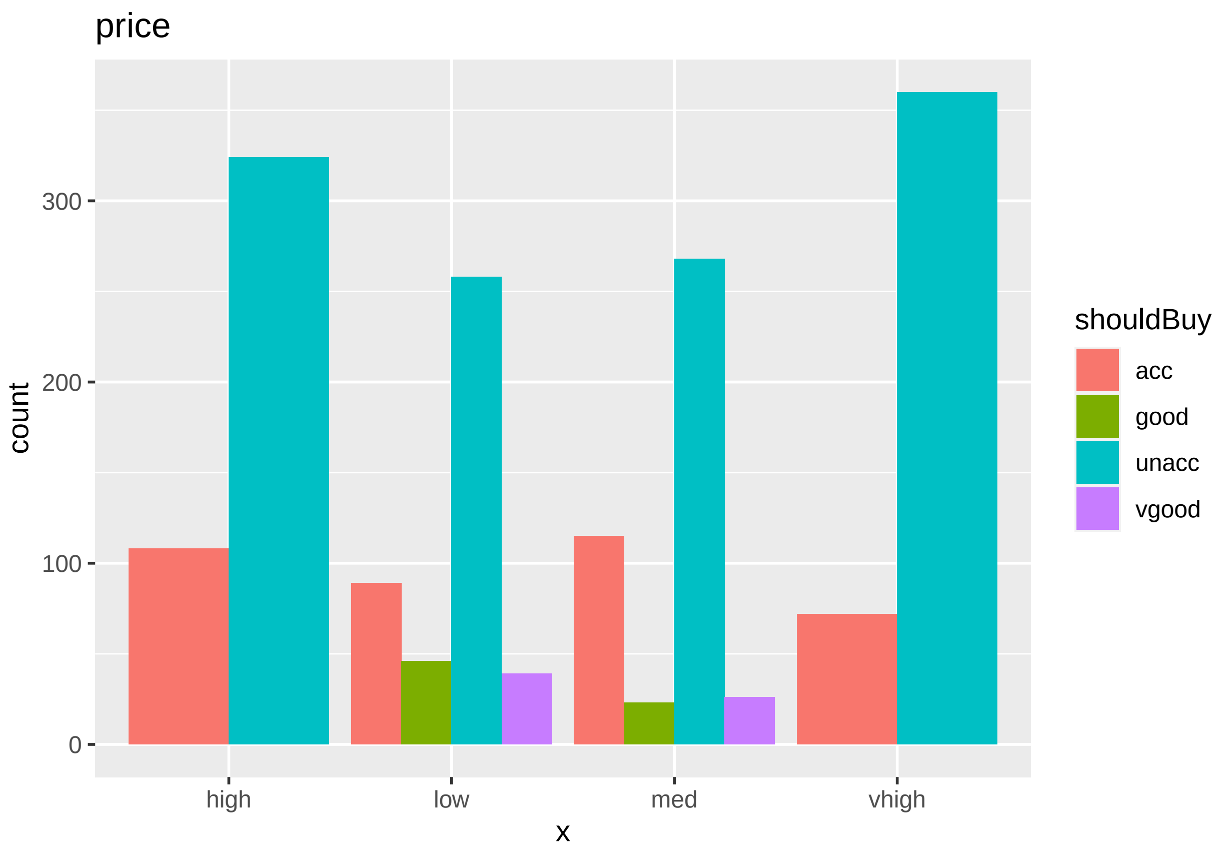
```
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

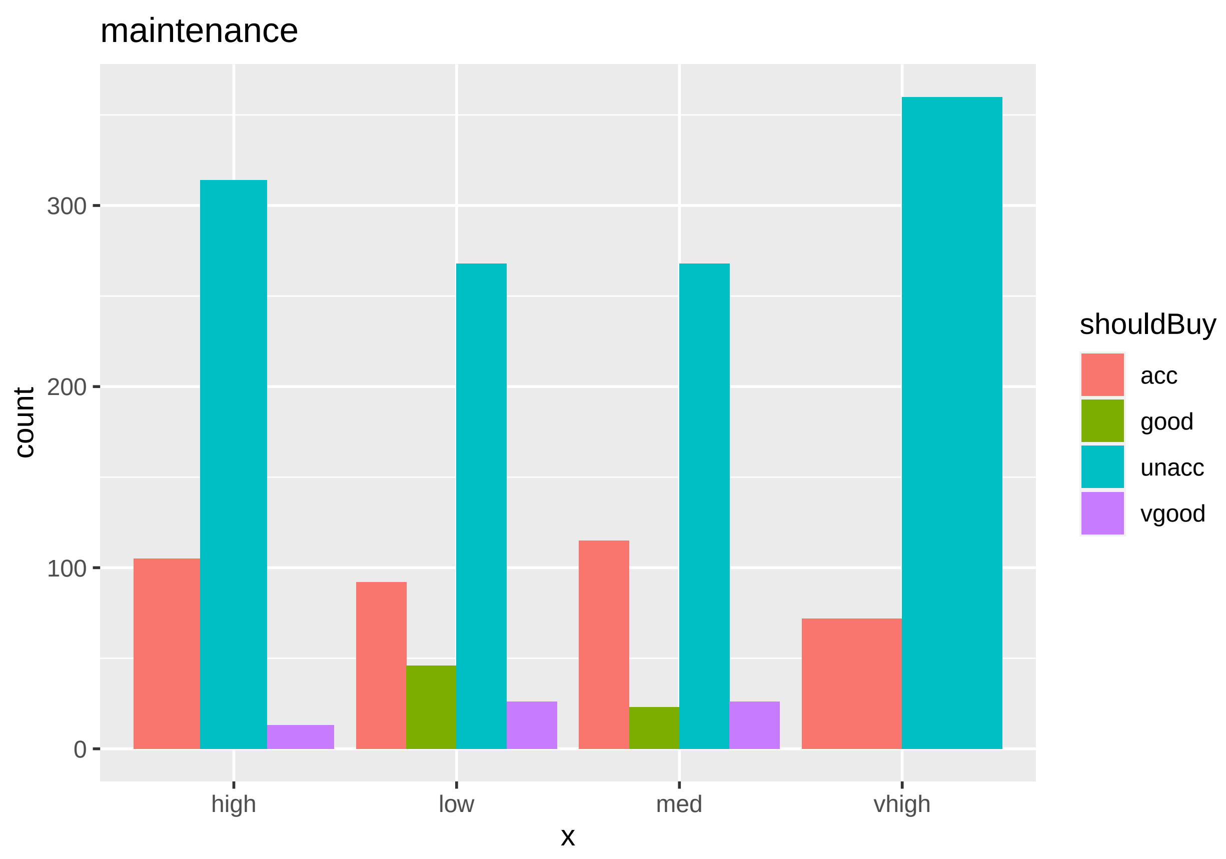
```
#DATA EXTRACTION
cardata <- read.csv(file="C:/Users/maria/Desktop/DMASSIGNMENTS/Assignment2/car.txt",header=TRUE,sep=","
```

```r
#DATA EXPLORATION

i<<-0;
lapply(cardata %>% select(price:safety),function(x,y=colnames(cardata)){
  i<<-i+1;
  ggplot(cardata)+
    geom_bar(aes(x,fill=shouldBuy),position="dodge")+
    ggtitle(y[[i]])
})
```

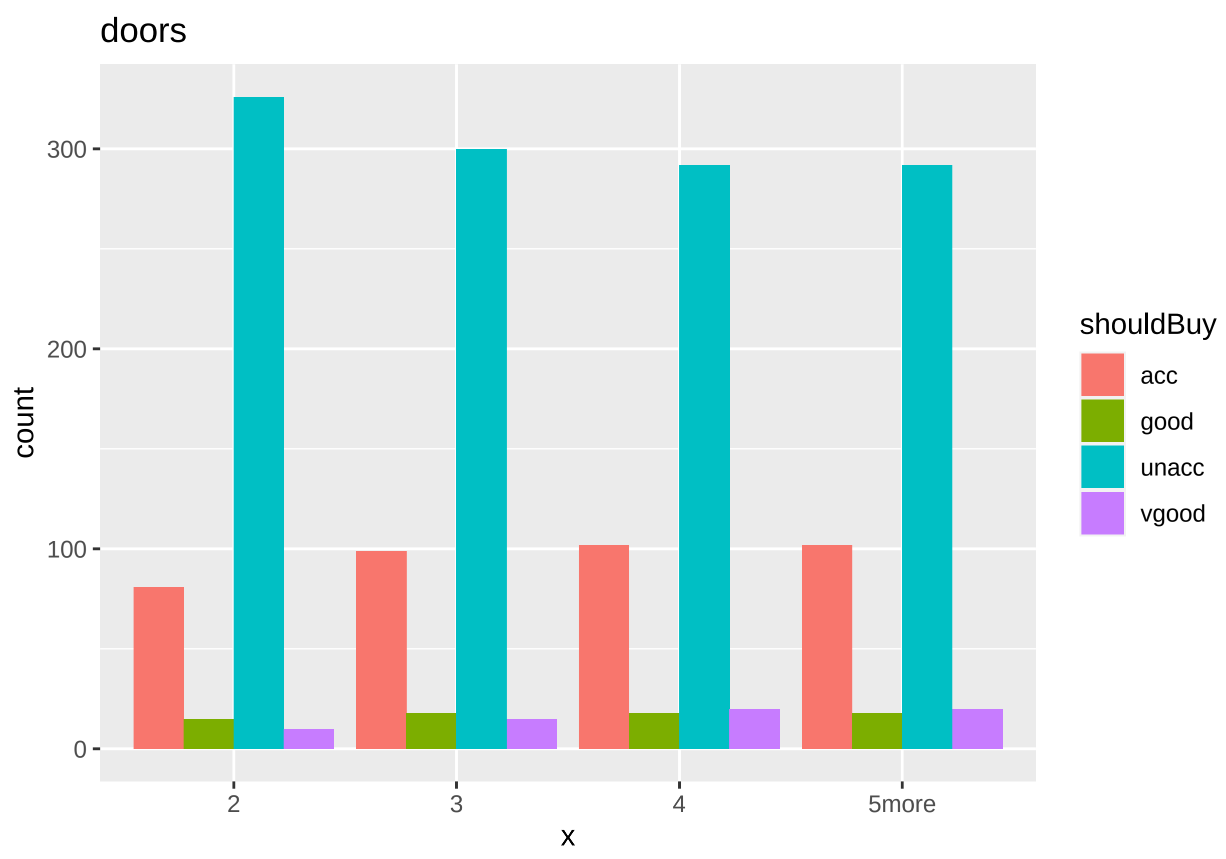
```
## $price
```



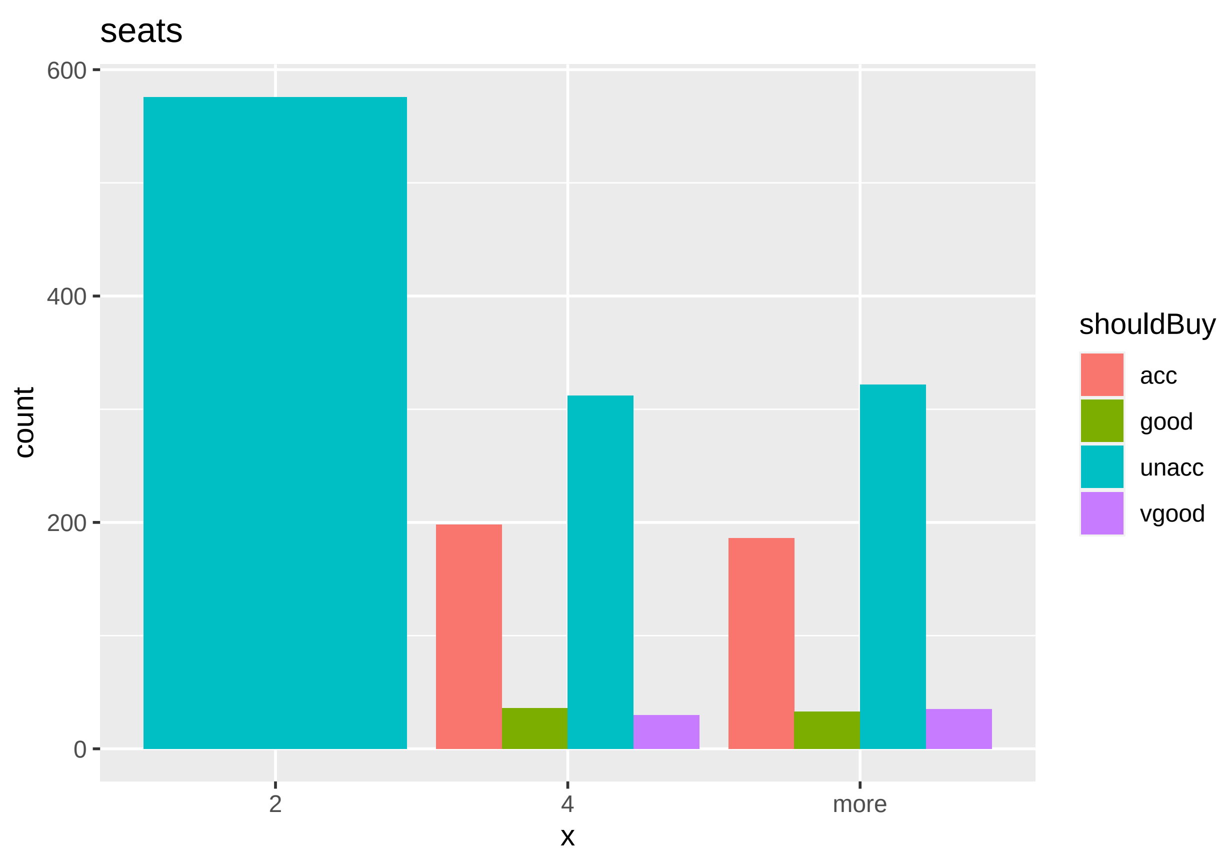
```
##
## $maintenance
```


maintenance
count
300
200
100
0
high
low
med
vhigh
x
shouldBuy
acc
good
unacc
vgood

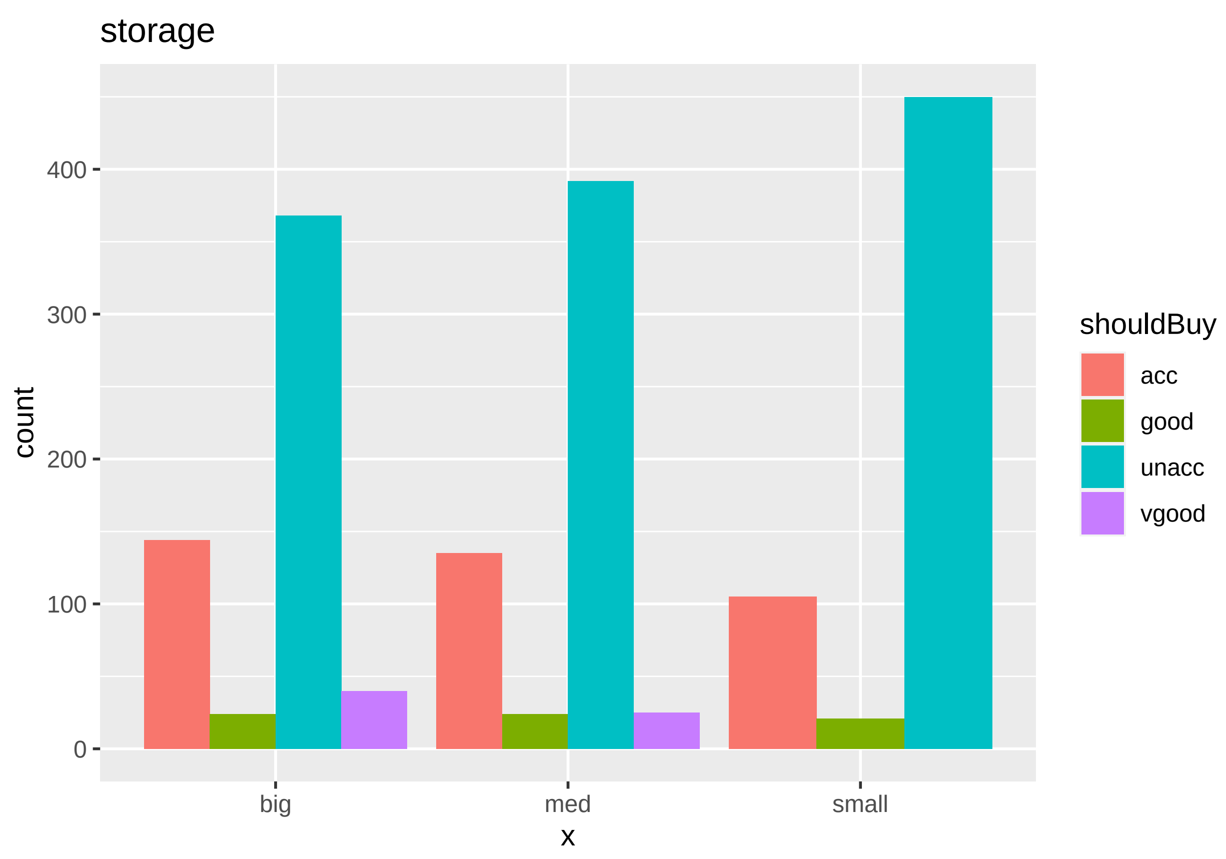
```
##
## $doors
```


doors
count
300
200
100
0
2
3
4
5more
x
shouldBuy
acc
good
unacc
vgood

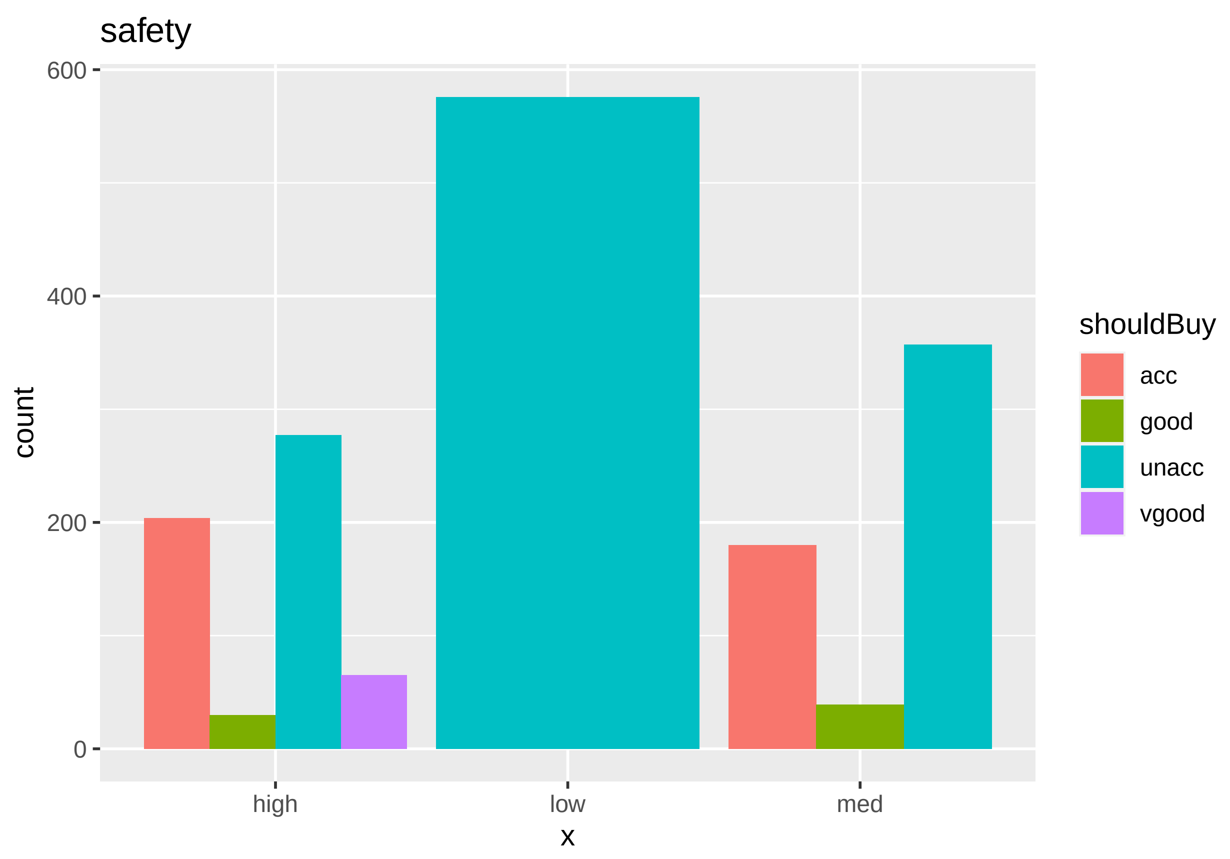
```
##
## $seats
```


seats
600
400
200
0
count
2
4
more
x
shouldBuy
acc
good
unacc
vgood

```
##
## $storage
```


storage
count
400
300
200
100
0
big
med
small
x
shouldBuy
acc
good
unacc
vgood

```
##
## $safety
```



```r
lapply(cardata,function(x){table(x)}) #no typos error
```

```
## $price
## x
##  high   low   med vhigh
##   432   432   432   432
##
## $maintenance
## x
##  high   low   med vhigh
##   432   432   432   432
##
## $doors
## x
##     2     3     4 5more
##   432   432   432   432
##
## $seats
## x
##    2    4 more
##  576  576  576
##
## $storage
## x
##   big   med small
##   576   576   576
##
## $safety
## x
## high  low  med
##  576  576  576
##
## $shouldBuy
## x
##   acc  good unacc vgood
##   384    69  1210    65
```

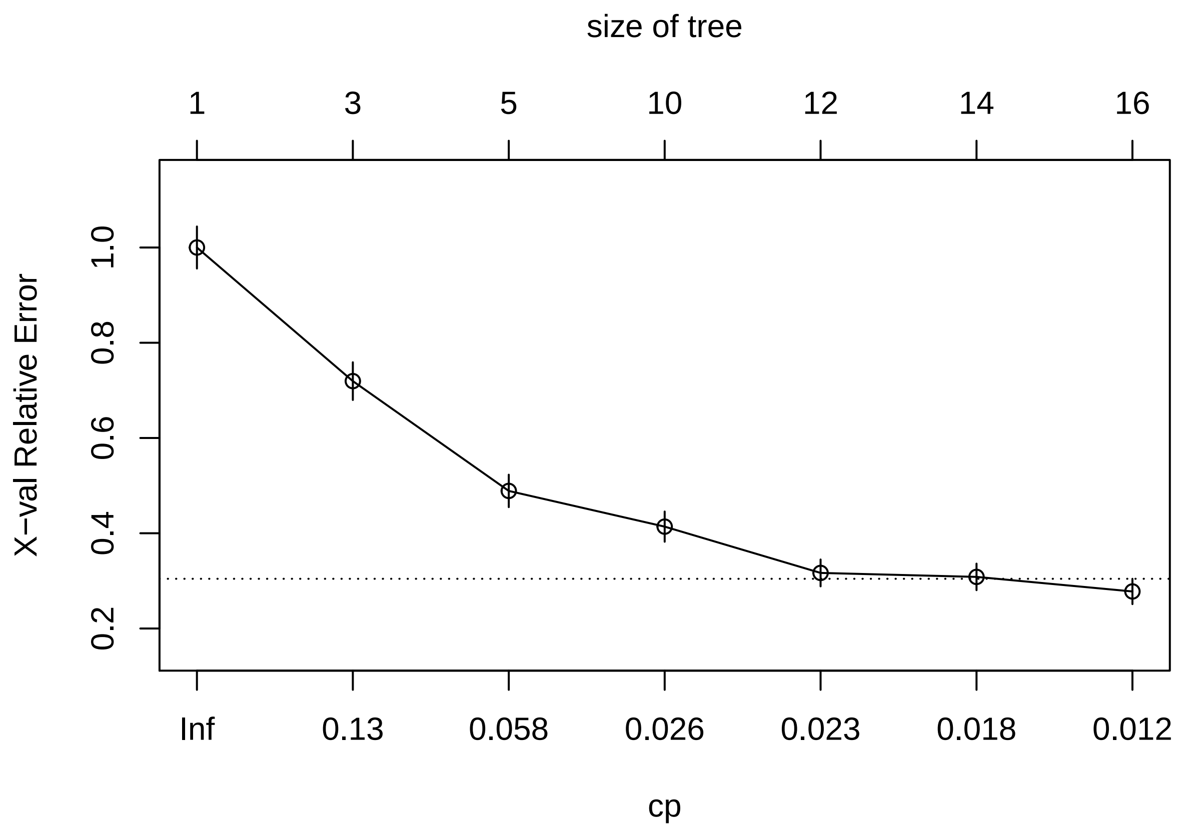
```
sum(!complete.cases(cardata)) #rows with NA
```

```
## [1] 0
```

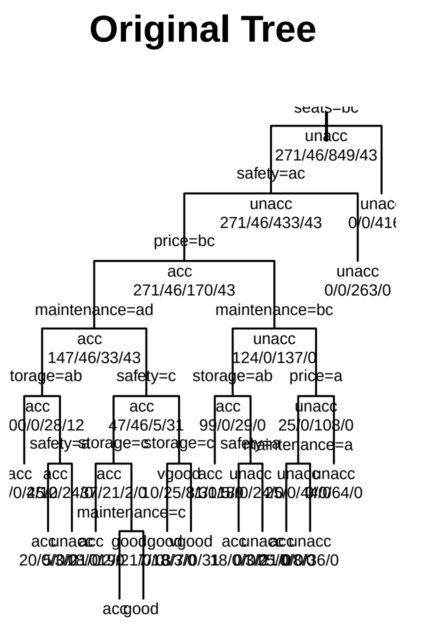
```
cardata <- cardata[complete.cases(cardata),]
data.samples <- sample(1:nrow(cardata), nrow(cardata) *0.7, replace = FALSE)
```

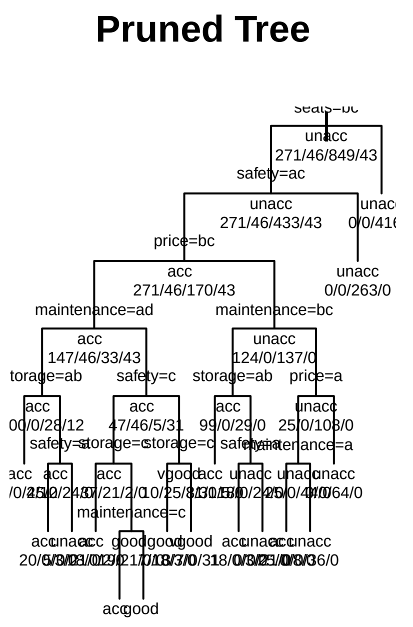
```
#SPLIT DATA INTO TRAIN AND TEST
set.seed(100)
train.data <- cardata[data.samples,]
test.data <- cardata[-data.samples,] %>% select(-shouldBuy)
```

```
#MODEL BULDING
model<-rpart(shouldBuy~.,method="class",data=train.data)

#PREDICT
pred_before_prunining<-predict(model,test.data,type="class")

plotcp(model)
```



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

plot(model, uniform = TRUE, margin = 0, main = "Original Tree")
text(model, use.n = TRUE, all = TRUE, cex = 0.5)

fit.pruned <- prune(model, cp = model$cptable[which.min(model$cptable[,"xerror"]), "CP"])
pred_after_prunining<-predict(fit.pruned,test.data,type="class")

plot(fit.pruned, uniform = TRUE, margin = 0, main = "Pruned Tree")
text(fit.pruned, use.n = TRUE, all = TRUE, cex = 0.5)
```





```
table(pred_before_prunining,cardata[-data.samples,]$shouldBuy) #confusion matrix
```

```
##
## pred_before_prunining acc good unacc vgood
##                 acc   111    9    20     1
##                 good    0   12     1     0
##                 unacc   2    0   340     0
##                 vgood   0    2     0    21
```

```
table(pred_after_prunining,cardata[-data.samples,]$shouldBuy)
```

```
##
## pred_after_prunining acc good unacc vgood
##                acc   111    9    20     1
##                good    0   12     1     0
##                unacc   2    0   340     0
##                vgood   0    2     0    21
```

```
#ACCURACY, SPECIFICITY, SENSITIVITY, ROC

confusionMatrix(pred_before_prunining,factor(cardata[-data.samples,]$shouldBuy))
```

```
## Confusion Matrix and Statistics
##
```

```
##           Reference
## Prediction acc good unacc vgood
##      acc   111    9    20     1
##      good    0   12     1     0
##      unacc   2    0   340     0
##      vgood   0    2     0    21
##
## Overall Statistics
##
##                Accuracy : 0.9326
##                  95% CI : (0.9075, 0.9526)
##     No Information Rate : 0.6956
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8594
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: acc Class: good Class: unacc Class: vgood
## Sensitivity              0.9823     0.52174       0.9418      0.95455
## Specificity              0.9261     0.99798       0.9873      0.99598
## Pos Pred Value           0.7872     0.92308       0.9942      0.91304
## Neg Pred Value           0.9947     0.97826       0.8814      0.99798
## Prevalence               0.2177     0.04432       0.6956      0.04239
## Detection Rate           0.2139     0.02312       0.6551      0.04046
## Detection Prevalence     0.2717     0.02505       0.6590      0.04432
## Balanced Accuracy        0.9542     0.75986       0.9646      0.97526
```

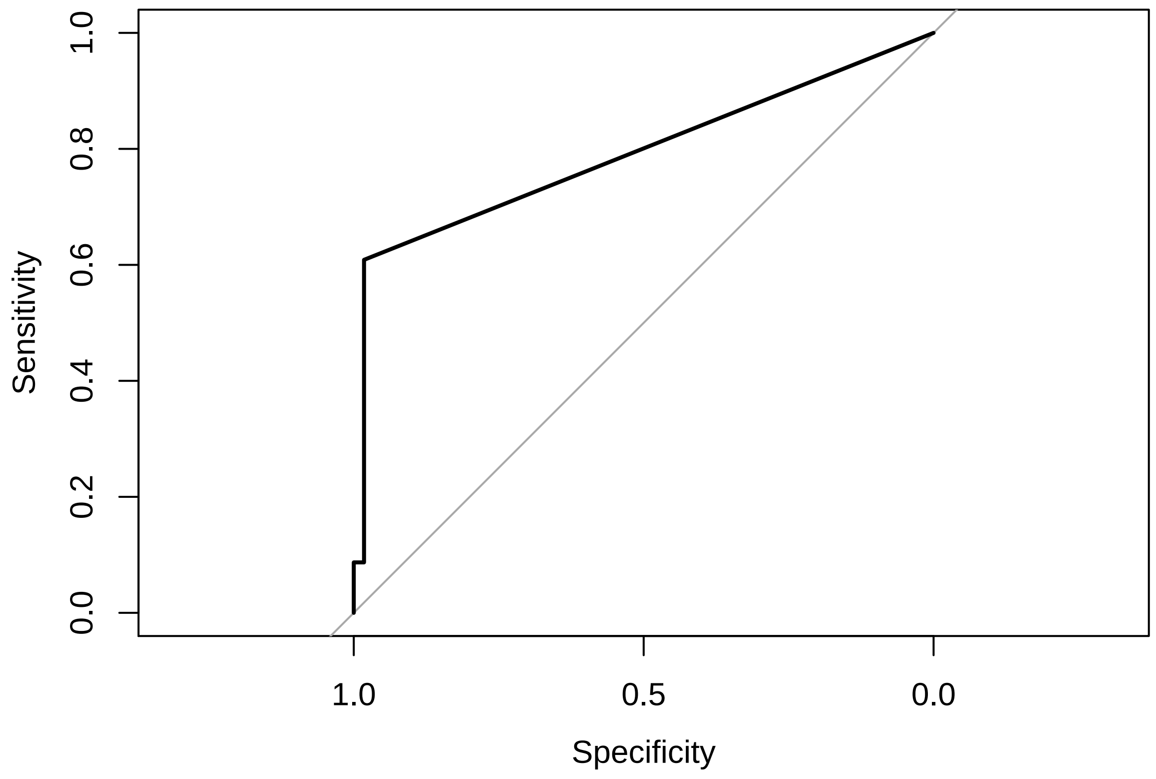
```
confusionMatrix(pred_after_prunining,factor(cardata[-data.samples,]$shouldBuy))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction acc good unacc vgood
##      acc   111    9    20     1
##      good    0   12     1     0
##      unacc   2    0   340     0
##      vgood   0    2     0    21
##
## Overall Statistics
##
##                Accuracy : 0.9326
##                  95% CI : (0.9075, 0.9526)
##     No Information Rate : 0.6956
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8594
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
```

```
##                      Class: acc Class: good Class: unacc Class: vgood
## Sensitivity              0.9823     0.52174       0.9418      0.95455
## Specificity              0.9261     0.99798       0.9873      0.99598
## Pos Pred Value           0.7872     0.92308       0.9942      0.91304
## Neg Pred Value           0.9947     0.97826       0.8814      0.99798
## Prevalence               0.2177     0.04432       0.6956      0.04239
## Detection Rate           0.2139     0.02312       0.6551      0.04046
## Detection Prevalence     0.2717     0.02505       0.6590      0.04432
## Balanced Accuracy        0.9542     0.75986       0.9646      0.97526
```

```
par(mfrow = c(1, 1))
plot(roc(response=cardata[-data.samples,]$shouldBuy,predictor=factor(pred_before_prunining,ordered = TRU
```

```
## Warning in roc.default(response = cardata[-data.samples, ]$shouldBuy, predictor
## = factor(pred_before_prunining, : 'response' has more than two levels. Consider
## setting 'levels' explicitly or using 'multiclass.roc' instead

## Setting levels: control = acc, case = good

## Warning in value[[3L]](cond): Ordered predictor converted to numeric vector.
## Threshold values will not correspond to values in predictor.

## Setting direction: controls < cases
```


Sensitivity
Specificity