CAC II

20122065

KUMAR GAURAV

In [1]:
```
df <- read.csv('car_evaluation.csv')
```

In [2]:
```
df
```

A data.frame: 1728 × 7

buying.price	maintenance.cost	number.of.doors	number.of.persons	lug_boot	safety	decision
<fct>	<fct>	<fct>	<fct>	<fct>	<fct>	<fct>
vhigh	vhigh	2	2	small	low	unacc
vhigh	vhigh	2	2	small	med	unacc
vhigh	vhigh	2	2	small	high	unacc
vhigh	vhigh	2	2	med	low	unacc
vhigh	vhigh	2	2	med	med	unacc
vhigh	vhigh	2	2	med	high	unacc
vhigh	vhigh	2	2	big	low	unacc
vhigh	vhigh	2	2	big	med	unacc
vhigh	vhigh	2	2	big	high	unacc

Structure of the dataset

In [3]:
```
str(df)
```

```
'data.frame':   1728 obs. of  7 variables:
 $ buying.price     : Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4
4 4 ...
 $ maintenance.cost : Factor w/ 4 levels "high","low","med",..: 4 4 4 4 4 4 4 4
4 4 ...
 $ number.of.doors  : Factor w/ 4 levels "2","3","4","5more": 1 1 1 1 1 1 1 1 1
1 ...
 $ number.of.persons: Factor w/ 3 levels "2","4","more": 1 1 1 1 1 1 1 1 1 2
...
 $ lug_boot         : Factor w/ 3 levels "big","med","small": 3 3 3 2 2 2 1 1 1
3 ...
 $ safety           : Factor w/ 3 levels "high","low","med": 2 3 1 2 3 1 2 3 1
2 ...
 $ decision         : Factor w/ 4 levels "acc","good","unacc",..: 3 3 3 3 3 3 3
3 3 3 ...
```

Summary of the dataset

In [4]:
```
summary(df)
```

```
 buying.price maintenance.cost number.of.doors number.of.persons  lug_boot
 high :432    high :432        2    :432       2   :576          big  :576
 low  :432    low  :432        3    :432       4   :576          med  :576
 med  :432    med  :432        4    :432       more:576          small:576
 vhigh:432    vhigh:432        5more:432
  safety     decision
 high:576   acc  : 384
 low :576   good :  69
 med :576   unacc:1210
            vgood:  65
```

]:

Checking for null values

In [5]:
```
any(is.na(df))
```

FALSE

Visualizing using ggplot2

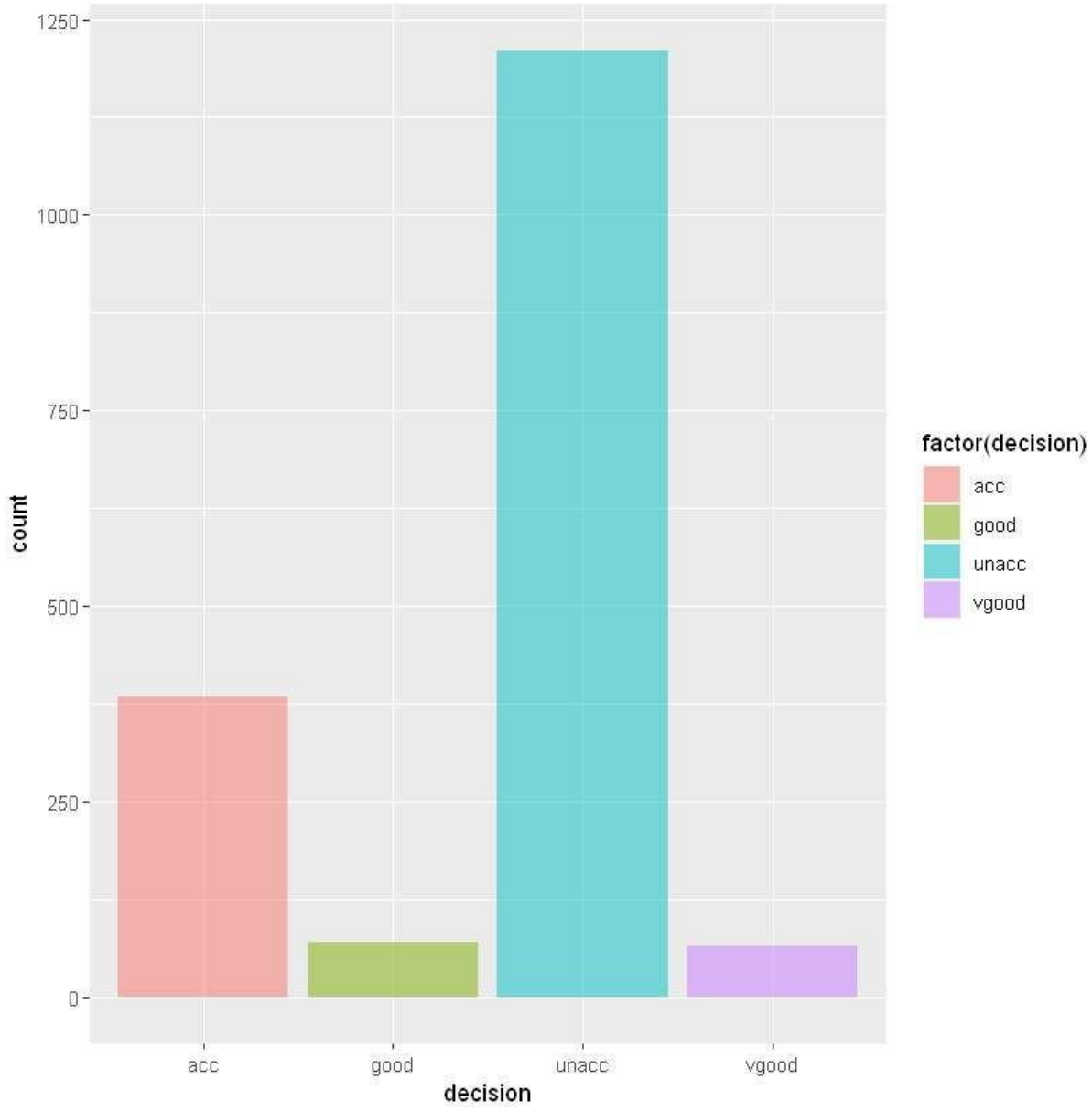
In [6]:
```
library(ggplot2)
```

```
Warning message:
"package 'ggplot2' was built under R version 3.6.3"
```

'decision' column visualization

In [7

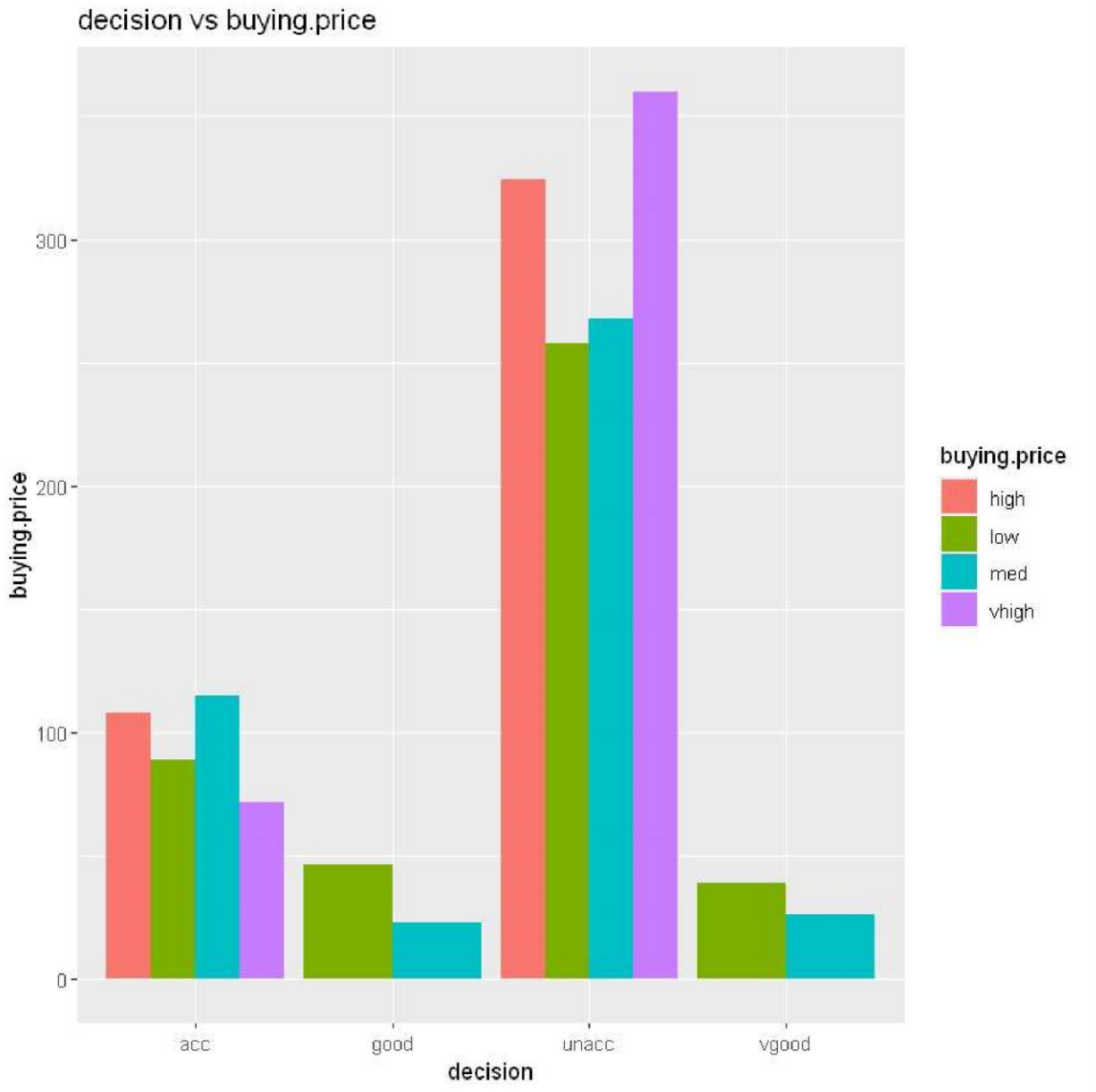
```
ggplot(df,aes(decision))+geom_bar(aes(fill=factor(decision)),alpha=0.5)
```



]:

Barplot of decision vs buying.price

In [37]:
```
ggplot(df,aes(x=decision,fill=buying.price)) + geom_bar(position="dodge")+labs(ti
```

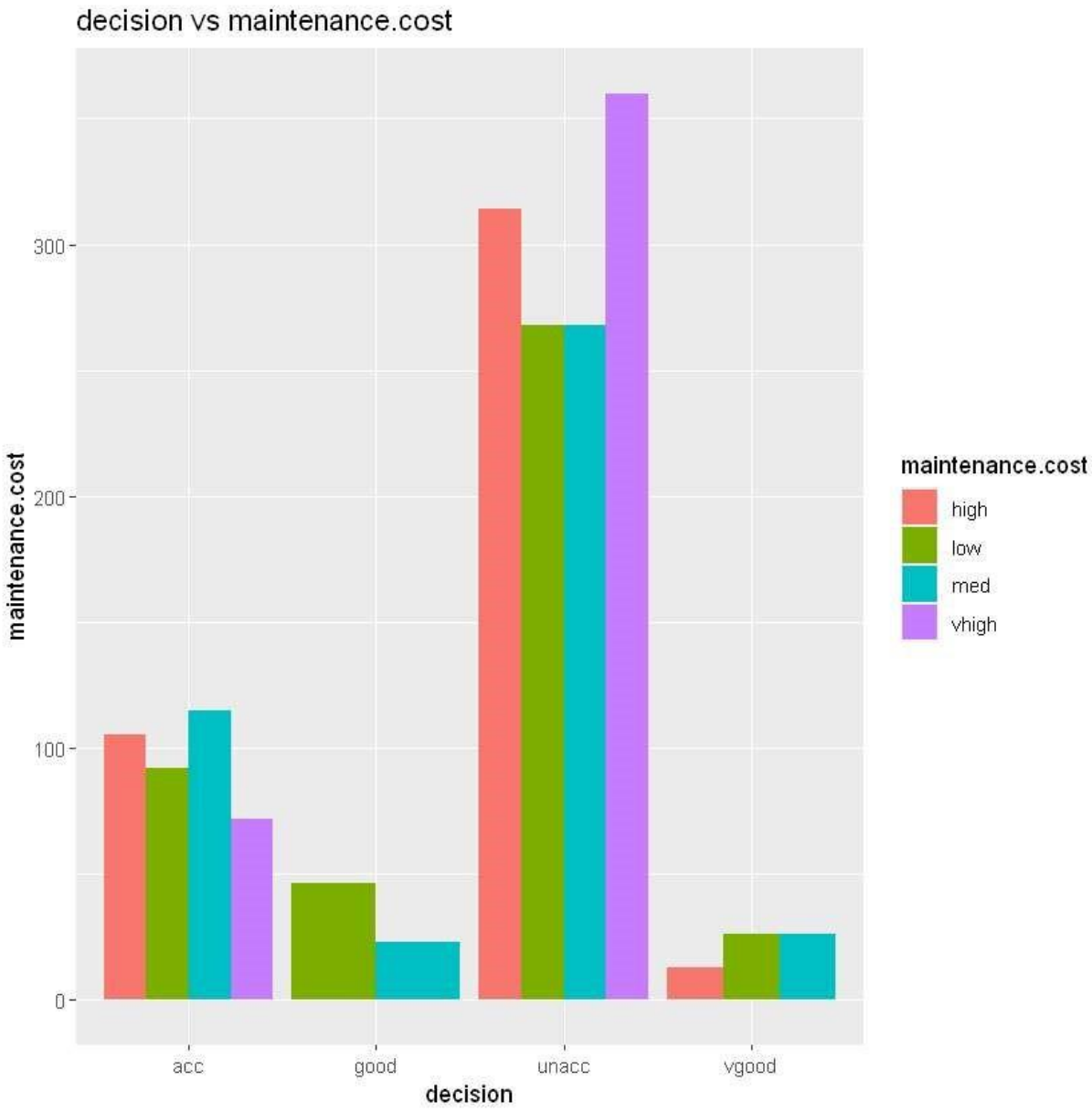


It is seen that people from unacc decision group, have the maximum no of high,low,med,vhigh buying prices.

Barplot of decision vs maintenance.cost

In [36]:
```
ggplot(df,aes(x=decision,fill=                                                    '
```

```
                                                    maintenance.cost)) + geom_bar(position= 'dodge")+lab
```

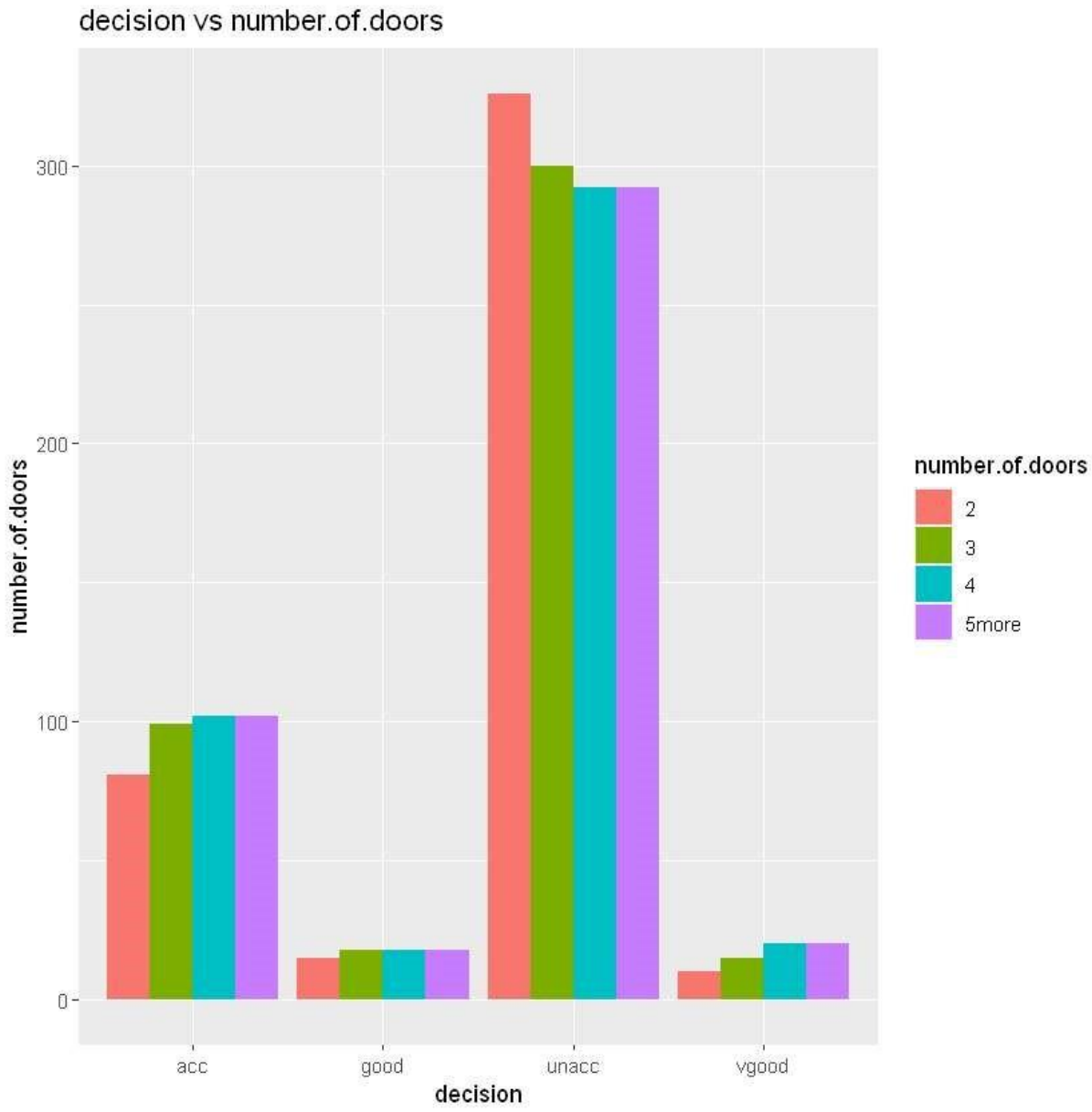


It is seen that people from unacc decision group, have the maximum no of high,low,med,vhigh maintenance costs.

Barplot of decision vs

number.of.doors

In [35]:
```
ggplot(df,aes(x=decision,fill= u                                                    '
                                                                                    n mber.of.doors)) + geom_bar(position='dodge")+labs
```



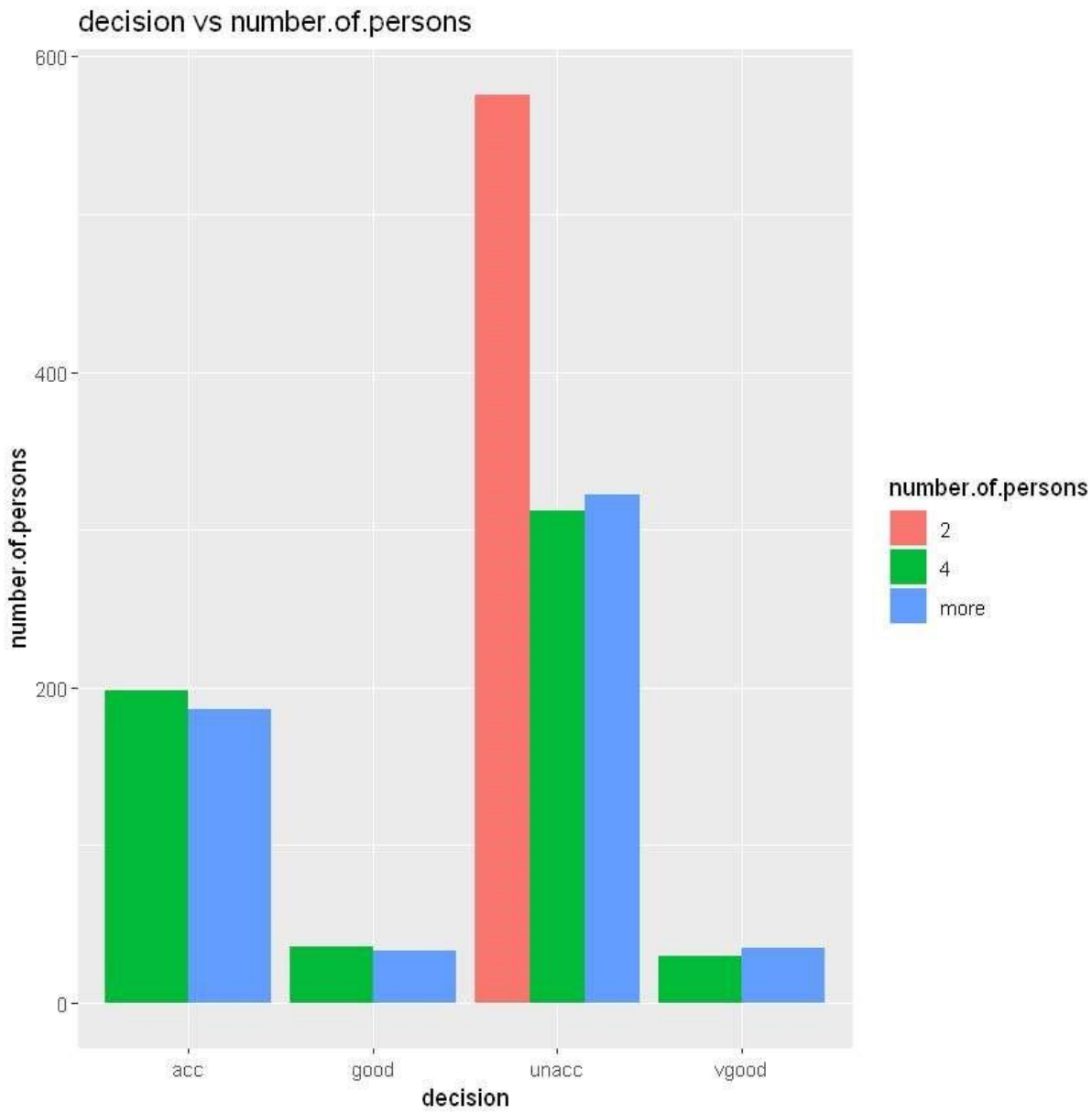
Barplot of decision vs

It is seen that people from unacc decision group, have maximum no in 2,3,4,5more car doors.

In [38]:
```
ggplot(df,aes(x=decision,fill= u                                    '
```

number.of.persons

```
    n mber.of.persons)) + geom_bar(position="dodge")+la
```



Barplot of decision vs

It is seen that people from unacc decision group, have maximum no in 2,4,more persons.

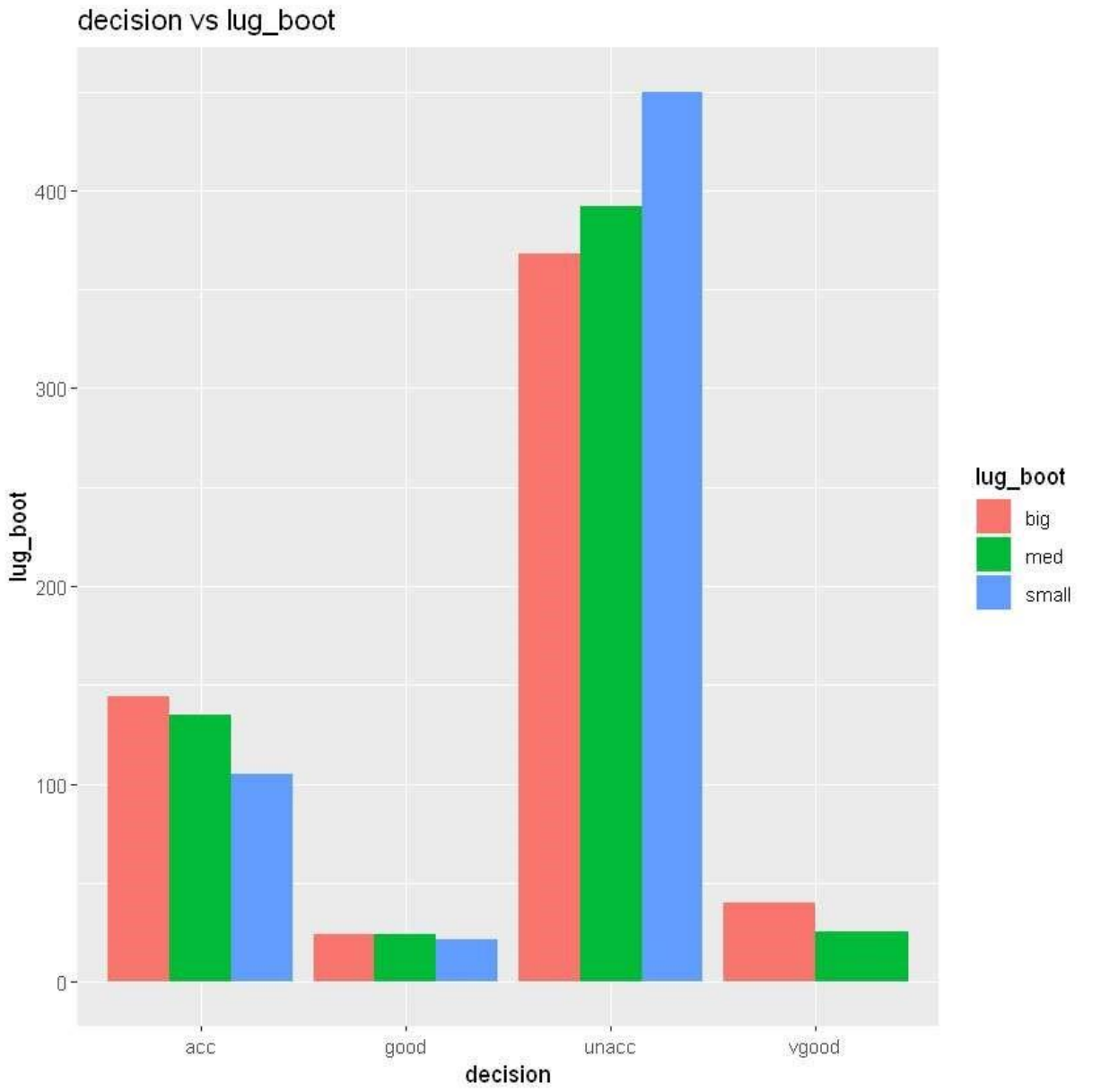
In [3]:
```
ggplot(df,aes(x=decision,fill= u                                    '
```

lug_boot

Barplot of decision vs

In [39]:
```
ggplot(df,aes(x=decision,fill= u
```

```
l g_boot)) + geom_bar(position="dodge")+labs(title=
```

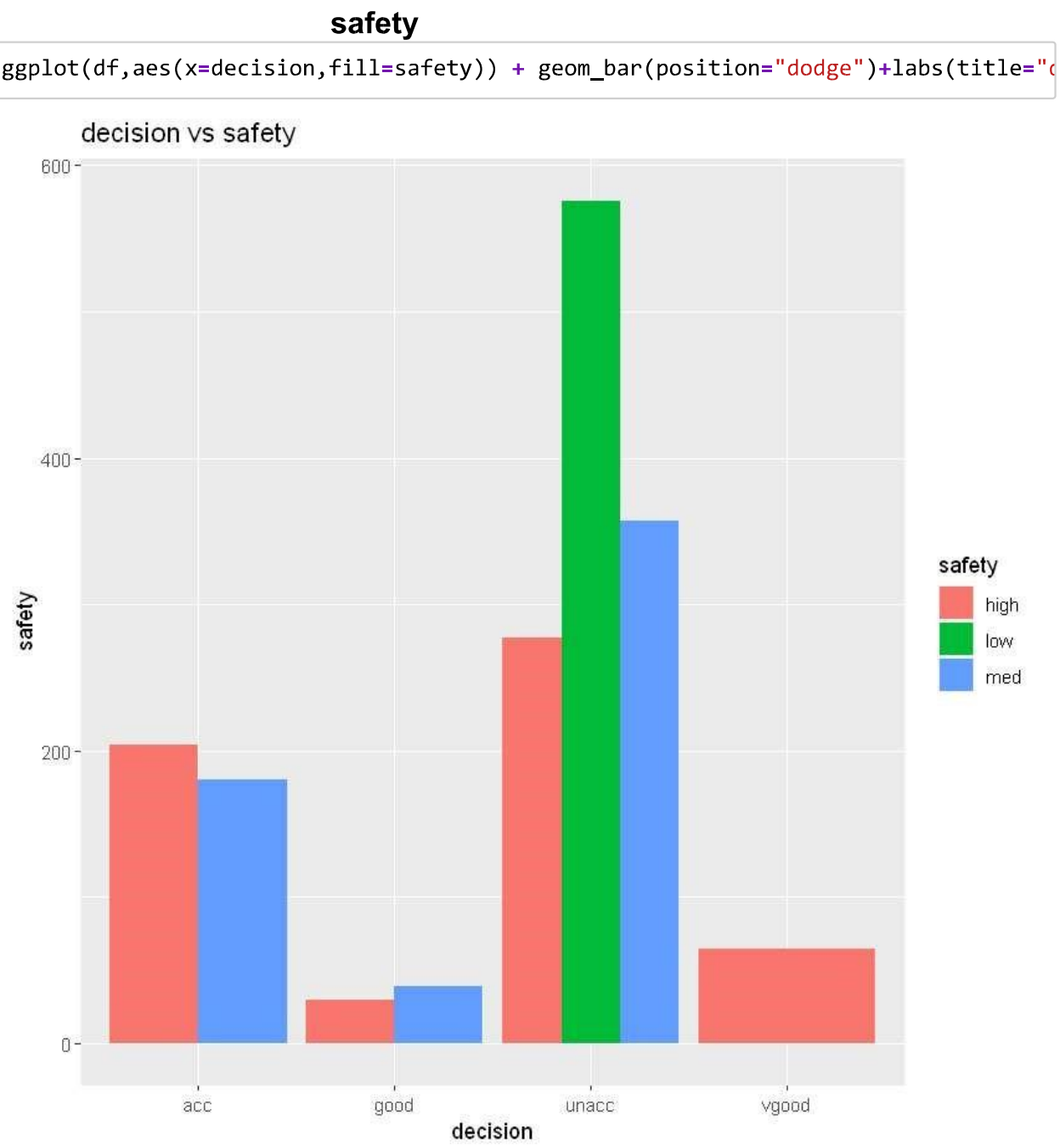


Barplot of decision vs

It is seen that people from unacc decision group, have maximum no of big,med,small lug boot.

In [ ]:

```
40  ggplot(df,aes(x=decision,fill=safety)) + geom_bar(position="dodge")+labs(title="
```



In [ ]:

It is seen that people from unacc decision group, have maximum no of high,low,med safety.

Splitting the dataset into test and train

In [10]:
```
library(caTools)
```

```
Warning message:
"package 'caTools' was built under R version 3.6.3"
```

11
```
set.seed(101)
sample<-sample.split(df$decision,SplitRatio = 0.7)
train=subset(df,sample==TRUE)
test=subset(df,sample==FALSE)
print(dim(train));print(dim(test))
```

```
[1] 1210    7
[1] 518   7
```

Decision Tree

In [19]:
```
library(rpart)
```

```
Warning message:
"package 'rpart' was built under R version 3.6.3"
```

In [ ]:

```
tree = rpart(decision~.,method="class",data=train)
summary(tree)
```

```
Call:
rpart(formula = decision ~ ., data = train, method = "class")
  n= 1210

           CP nsplit rel error    xerror       xstd
1  0.12396694      0 1.0000000 1.0000000 0.04391326
2  0.11019284      2 0.7520661 0.9173554 0.04279791
3  0.03994490      4 0.5316804 0.5316804 0.03508648
4  0.03856749      6 0.4517906 0.5096419 0.03448645
5  0.03719008      7 0.4132231 0.4710744 0.03338155
6  0.02203857      9 0.3388430 0.3471074 0.02926851
7  0.02066116     11 0.2947658 0.3553719 0.02957389
8  0.01928375     13 0.2534435 0.3553719 0.02957389
9  0.01377410     15 0.2148760 0.2644628 0.02589876
10 0.01101928     16 0.2011019 0.2424242 0.02488505
11 0.01000000     17 0.1900826 0.2396694 0.02475427

Variable importance
           safety number.of.persons  maintenance.cost          lug_boot
               30                28                18                12
     buying.price   number.of.doors
               11                 1

Node number 1: 1210 observations,    complexity param=0.1239669
  predicted class=unacc  expected loss=0.3  P(node) =1
    class counts:   269    48   847    46
   probabilities: 0.222 0.040 0.700 0.038
  left son=2 (806 obs) right son=3 (404 obs)
  Primary splits:
      safety            splits as  LRL,  improve=86.391530, (0 missing)
      number.of.persons splits as  RLL,  improve=82.595730, (0 missing)
      buying.price      splits as  RLLR, improve=15.115330, (0 missing)
      maintenance.cost  splits as  RLLR, improve= 7.633465, (0 missing)
      lug_boot          splits as  LLR,  improve= 6.596195, (0 missing)

Node number 2: 806 observations,    complexity param=0.1239669
  predicted class=unacc  expected loss=0.4503722  P(node) =0.6661157
    class counts:   269    48   443    46
   probabilities: 0.334 0.060 0.550 0.057
  left son=4 (542 obs) right son=5 (264 obs)
  Primary splits:
      number.of.persons splits as  RLL,  improve=126.03170, (0 missing)
      buying.price      splits as  RLLR, improve= 23.12407, (0 missing)
      maintenance.cost  splits as  RLLR, improve= 11.31667, (0 missing)
      lug_boot          splits as  LLR,  improve= 10.59030, (0 missing)
      safety            splits as  R-L,  improve=  7.02684, (0 missing)

Node number 3: 404 observations
  predicted class=unacc  expected loss=0  P(node) =0.3338843
    class counts:     0     0   404     0
   probabilities: 0.000 0.000 1.000 0.000

Node number 4: 542 observations,    complexity param=0.1101928
```

```
              l s
  predicted c a s=acc    expected loss=0.50369  P(node) =0.4479339
    class counts:   269    48   179    46
   probabilities: 0.496 0.089 0.330 0.085
  left son=8 (272 obs) right son=9 (270 obs)
  Primary splits:
      buying.price     splits as  RLLR, improve=33.670420, (0 missing)
      maintenance.cost splits as  RLLR, improve=18.979220, (0 missing)
      lug_boot         splits as  LLR,  improve=15.613500, (0 missing)
      safety           splits as  R-L,  improve=10.691700, (0 missing)
      number.of.doors  splits as  RRLL, improve= 4.436457, (0 missing)
  Surrogate splits:
      maintenance.cost splits as  RRLL, agree=0.526, adj=0.048, (0 split)
      lug_boot         splits as  LRL,  agree=0.526, adj=0.048, (0 split)
      safety           splits as  L-R,  agree=0.522, adj=0.041, (0 split)
      number.of.doors  splits as  RRLL, agree=0.515, adj=0.026, (0 split)

Node number 5: 264 observations
  predicted class=unacc  expected loss=0  P(node) =0.2181818
    class counts:     0     0   264     0
   probabilities: 0.000 0.000 1.000 0.000

Node number 8: 272 observations,    complexity param=0.0399449
  predicted class=acc    expected loss=0.4632353  P(node) =0.2247934
    class counts:   146    48    32    46
   probabilities: 0.537 0.176 0.118 0.169
  left son=16 (140 obs) right son=17 (132 obs)
  Primary splits:
      maintenance.cost splits as  LRRL, improve=22.940590, (0 missing)
      safety           splits as  R-L,  improve= 9.913895, (0 missing)
      lug_boot         splits as  RRL,  improve= 8.910401, (0 missing)
      buying.price     splits as  -RL-, improve= 4.735297, (0 missing)
      number.of.doors  splits as  RLLL, improve= 1.737302, (0 missing)
  Surrogate splits:
      safety          splits as  L-R,  agree=0.533, adj=0.038, (0 split)
      buying.price    splits as  -RL-, agree=0.518, adj=0.008, (0 split)
      number.of.doors splits as  RLLL, agree=0.518, adj=0.008, (0 split)
      lug_boot        splits as  LRL,  agree=0.518, adj=0.008, (0 split)

Node number 9: 270 observations,    complexity param=0.1101928
  predicted class=unacc  expected loss=0.4555556  P(node) =0.2231405
    class counts:   123     0   147     0
   probabilities: 0.456 0.000 0.544 0.000
  left son=18 (136 obs) right son=19 (134 obs)
  Primary splits:
      maintenance.cost splits as  RLLR, improve=34.343340, (0 missing)
      lug_boot         splits as  LLR,  improve=11.792880, (0 missing)
      safety           splits as  L-R,  improve= 8.274663, (0 missing)
      number.of.doors  splits as  RRLL, improve= 4.115751, (0 missing)
      buying.price     splits as  L--R, improve= 2.330567, (0 missing)
  Surrogate splits:
      buying.price      splits as  R--L, agree=0.533, adj=0.060, (0 split)
      number.of.doors   splits as  RRLL, agree=0.526, adj=0.045, (0 split)
      lug_boot          splits as  RLL,  agree=0.519, adj=0.030, (0 split)
      safety            splits as  R-L,  agree=0.519, adj=0.030, (0 split)
      number.of.persons splits as  -LR,  agree=0.507, adj=0.007, (0 split)

Node number 16: 140 observations,    complexity param=0.01928375
```

```
              l s
  predicted c a s=acc    expected loss=0.2714286  P(node) =0.1157025
    class counts:   102     0    26    12
   probabilities: 0.729 0.000 0.186 0.086
  left son=32 (91 obs) right son=33 (49 obs)
  Primary splits:
      lug_boot         splits as  LLR,  improve=8.1551020, (0 missing)
      safety           splits as  L-R,  improve=5.5313130, (0 missing)
      buying.price     splits as  -RL-, improve=1.6215900, (0 missing)
      maintenance.cost splits as  R--L, improve=1.4690940, (0 missing)
      number.of.doors  splits as  RLLL, improve=0.8585381, (0 missing)

Node number 17: 132 observations,    complexity param=0.0399449
  predicted class=good   expected loss=0.6363636  P(node) =0.1090909
    class counts:    44    48     6    34
   probabilities: 0.333 0.364 0.045 0.258
  left son=34 (68 obs) right son=35 (64 obs)
  Primary splits:
      safety           splits as  R-L,  improve=17.589290, (0 missing)
      lug_boot         splits as  RRL,  improve= 7.478744, (0 missing)
      buying.price     splits as  -RL-, improve= 4.500098, (0 missing)
      maintenance.cost splits as  -RL-, improve= 3.059983, (0 missing)
      number.of.doors  splits as  LRRR, improve= 1.567778, (0 missing)
  Surrogate splits:
      maintenance.cost splits as  -RL-, agree=0.538, adj=0.047, (0 split)
      number.of.doors  splits as  LLRL, agree=0.523, adj=0.016, (0 split)
      lug_boot         splits as  LLR,  agree=0.523, adj=0.016, (0 split)

Node number 18: 136 observations,    complexity param=0.03719008
  predicted class=acc    expected loss=0.2941176  P(node) =0.1123967
    class counts:    96     0    40     0
   probabilities: 0.706 0.000 0.294 0.000
  left son=36 (88 obs) right son=37 (48 obs)
  Primary splits:
      lug_boot          splits as  LLR,  improve=16.24332000, (0 missing)
      safety            splits as  L-R,  improve=13.76091000, (0 missing)
      number.of.doors   splits as  RRLL, improve= 3.13033300, (0 missing)
      buying.price      splits as  R--L, improve= 0.18462490, (0 missing)
      number.of.persons splits as  -RL,  improve= 0.09853546, (0 missing)

Node number 19: 134 observations,    complexity param=0.02203857
  predicted class=unacc  expected loss=0.2014925  P(node) =0.1107438
    class counts:    27     0   107     0
   probabilities: 0.201 0.000 0.799 0.000
  left son=38 (70 obs) right son=39 (64 obs)
  Primary splits:
      buying.price     splits as  L--R, improve=9.9479740, (0 missing)
      maintenance.cost splits as  L--R, improve=8.5594030, (0 missing)
      lug_boot         splits as  LLR,  improve=1.0113220, (0 missing)
      number.of.doors  splits as  RRLL, improve=0.4416576, (0 missing)
      safety           splits as  L-R,  improve=0.4358023, (0 missing)
  Surrogate splits:
      lug_boot        splits as  RLL,  agree=0.537, adj=0.031, (0 split)
      number.of.doors splits as  LRLL, agree=0.530, adj=0.016, (0 split)

Node number 32: 91 observations
  predicted class=acc    expected loss=0.1758242  P(node) =0.07520661
    class counts:    75     0     4    12
```

```
              l s
   probabilitie : 0.824 0.000 0.044 0.132

Node number 33: 49 observations,    complexity param=0.01928375
  predicted class=acc    expected loss=0.4489796  P(node) =0.04049587
    class counts:    27     0    22     0
   probabilities: 0.551 0.000 0.449 0.000
  left son=66 (25 obs) right son=67 (24 obs)
  Primary splits:
      safety            splits as  L-R,  improve=11.0482300, (0 missing)
      number.of.doors   splits as  RLLL, improve= 1.5833590, (0 missing)
      number.of.persons splits as  -LR,  improve= 1.2582310, (0 missing)
      buying.price      splits as  -LR-, improve= 0.8870384, (0 missing)
      maintenance.cost  splits as  L--R, improve= 0.5148980, (0 missing)
  Surrogate splits:
      maintenance.cost  splits as  R--L, agree=0.551, adj=0.083, (0 split)
      number.of.doors   splits as  LRLL, agree=0.531, adj=0.042, (0 split)
      number.of.persons splits as  -LR,  agree=0.531, adj=0.042, (0 split)

Node number 34: 68 observations,    complexity param=0.02066116
  predicted class=acc    expected loss=0.4264706  P(node) =0.05619835
    class counts:    39    27     2     0
   probabilities: 0.574 0.397 0.029 0.000
  left son=68 (21 obs) right son=69 (47 obs)
  Primary splits:
      lug_boot          splits as  RRL,  improve=8.255170, (0 missing)
      buying.price      splits as  -RL-, improve=4.364629, (0 missing)
      maintenance.cost  splits as  -RL-, improve=1.524913, (0 missing)
      number.of.persons splits as  -LR,  improve=1.147059, (0 missing)
      number.of.doors   splits as  LRRR, improve=1.124052, (0 missing)

Node number 35: 64 observations,    complexity param=0.03856749
  predicted class=vgood  expected loss=0.46875  P(node) =0.05289256
    class counts:     5    21     4    34
   probabilities: 0.078 0.328 0.062 0.531
  left son=70 (22 obs) right son=71 (42 obs)
  Primary splits:
      lug_boot          splits as  RRL,  improve=13.4841700, (0 missing)
      number.of.doors   splits as  LRRR, improve= 2.1634460, (0 missing)
      maintenance.cost  splits as  -RL-, improve= 1.2187500, (0 missing)
      buying.price      splits as  -RL-, improve= 0.7431958, (0 missing)
      number.of.persons splits as  -LR,  improve= 0.4727212, (0 missing)

Node number 36: 88 observations
  predicted class=acc    expected loss=0.1136364  P(node) =0.07272727
    class counts:    78     0    10     0
   probabilities: 0.886 0.000 0.114 0.000

Node number 37: 48 observations,    complexity param=0.03719008
  predicted class=unacc  expected loss=0.375  P(node) =0.03966942
    class counts:    18     0    30     0
   probabilities: 0.375 0.000 0.625 0.000
  left son=74 (21 obs) right son=75 (27 obs)
  Primary splits:
      safety            splits as  L-R,  improve=17.3571400, (0 missing)
      number.of.doors   splits as  RRLR, improve= 0.8292383, (0 missing)
      maintenance.cost  splits as  -LR-, improve= 0.3857143, (0 missing)
      number.of.persons splits as  -LR,  improve= 0.1666667, (0 missing)
```

```
      buying.price      splits as  R--L, improve= 0.1296296, (0 missing)
  Surrogate splits:
      number.of.doors splits as  LRLR, agree=0.604, adj=0.095, (0 split)

Node number 38: 70 observations,    complexity param=0.02203857
  predicted class=unacc  expected loss=0.3857143  P(node) =0.05785124
    class counts:    27     0    43     0
   probabilities: 0.386 0.000 0.614 0.000
  left son=76 (38 obs) right son=77 (32 obs)
  Primary splits:
      maintenance.cost  splits as  L--R, improve=17.5398500, (0 missing)
      lug_boot          splits as  LLR,  improve= 3.0945050, (0 missing)
      safety            splits as  L-R,  improve= 1.2271210, (0 missing)
      number.of.doors   splits as  RLLL, improve= 0.7833997, (0 missing)
      number.of.persons splits as  -LR,  improve= 0.4067227, (0 missing)
  Surrogate splits:
      number.of.doors splits as  RLLL, agree=0.557, adj=0.031, (0 split)

Node number 39: 64 observations
  predicted class=unacc  expected loss=0  P(node) =0.05289256
    class counts:     0     0    64     0
   probabilities: 0.000 0.000 1.000 0.000

Node number 66: 25 observations
  predicted class=acc    expected loss=0.12  P(node) =0.02066116
    class counts:    22     0     3     0
   probabilities: 0.880 0.000 0.120 0.000

Node number 67: 24 observations
  predicted class=unacc  expected loss=0.2083333  P(node) =0.01983471
    class counts:     5     0    19     0
   probabilities: 0.208 0.000 0.792 0.000

Node number 68: 21 observations
  predicted class=acc    expected loss=0.0952381  P(node) =0.01735537
    class counts:    19     0     2     0
   probabilities: 0.905 0.000 0.095 0.000

Node number 69: 47 observations,    complexity param=0.02066116
  predicted class=good   expected loss=0.4255319  P(node) =0.03884298
    class counts:    20    27     0     0
   probabilities: 0.426 0.574 0.000 0.000
  left son=138 (22 obs) right son=139 (25 obs)
  Primary splits:
      buying.price      splits as  -RL-, improve=5.4332690, (0 missing)
      maintenance.cost  splits as  -RL-, improve=2.9486480, (0 missing)
      lug_boot          splits as  RL-,  improve=2.2623600, (0 missing)
      number.of.doors   splits as  LRRR, improve=1.8739610, (0 missing)
      number.of.persons splits as  -LR,  improve=0.1948406, (0 missing)

Node number 70: 22 observations
  predicted class=good   expected loss=0.3636364  P(node) =0.01818182
    class counts:     4    14     4     0
   probabilities: 0.182 0.636 0.182 0.000

Node number 71: 42 observations
```

```
    class counts:     1     7     0    34
   probabilities: 0.024 0.167 0.000 0.810

Node number 74: 21 observations
  predicted class=acc    expected loss=0.1428571  P(node) =0.01735537
    class counts:    18     0     3     0
   probabilities: 0.857 0.000 0.143 0.000

Node number 75: 27 observations
  predicted class=unacc  expected loss=0  P(node) =0.02231405
    class counts:     0     0    27     0
   probabilities: 0.000 0.000 1.000 0.000

Node number 76: 38 observations,    complexity param=0.01101928
  predicted class=acc    expected loss=0.2894737  P(node) =0.03140496
    class counts:    27     0    11     0
   probabilities: 0.711 0.000 0.289 0.000
  left son=152 (24 obs) right son=153 (14 obs)
  Primary splits:
      lug_boot          splits as  LLR,  improve=5.5363410, (0 missing)
      safety            splits as  L-R,  improve=3.2730360, (0 missing)
      number.of.doors   splits as  RLLL, improve=0.5664448, (0 missing)
      number.of.persons splits as  -LR,  improve=0.1315789, (0 missing)

Node number 77: 32 observations
  predicted class=unacc  expected loss=0  P(node) =0.02644628
    class counts:     0     0    32     0
   probabilities: 0.000 0.000 1.000 0.000

Node number 138: 22 observations,    complexity param=0.0137741
  predicted class=acc    expected loss=0.3181818  P(node) =0.01818182
    class counts:    15     7     0     0
   probabilities: 0.682 0.318 0.000 0.000
  left son=276 (13 obs) right son=277 (9 obs)
  Primary splits:
      maintenance.cost  splits as  -RL-, improve=6.43434300, (0 missing)
      number.of.doors   splits as  LRRL, improve=0.24545450, (0 missing)
      lug_boot          splits as  RL-,  improve=0.09090909, (0 missing)
      number.of.persons splits as  -LR,  improve=0.01212121, (0 missing)

Node number 139: 25 observations
  predicted class=good   expected loss=0.2  P(node) =0.02066116
    class counts:     5    20     0     0
   probabilities: 0.200 0.800 0.000 0.000

Node number 152: 24 observations
  predicted class=acc    expected loss=0.08333333  P(node) =0.01983471
    class counts:    22     0     2     0
   probabilities: 0.917 0.000 0.083 0.000

Node number 153: 14 observations
  predicted class=unacc  expected loss=0.3571429  P(node) =0.01157025
    class counts:     5     0     9     0
   probabilities: 0.357 0.000 0.643 0.000

Node number 276: 13 observations
```

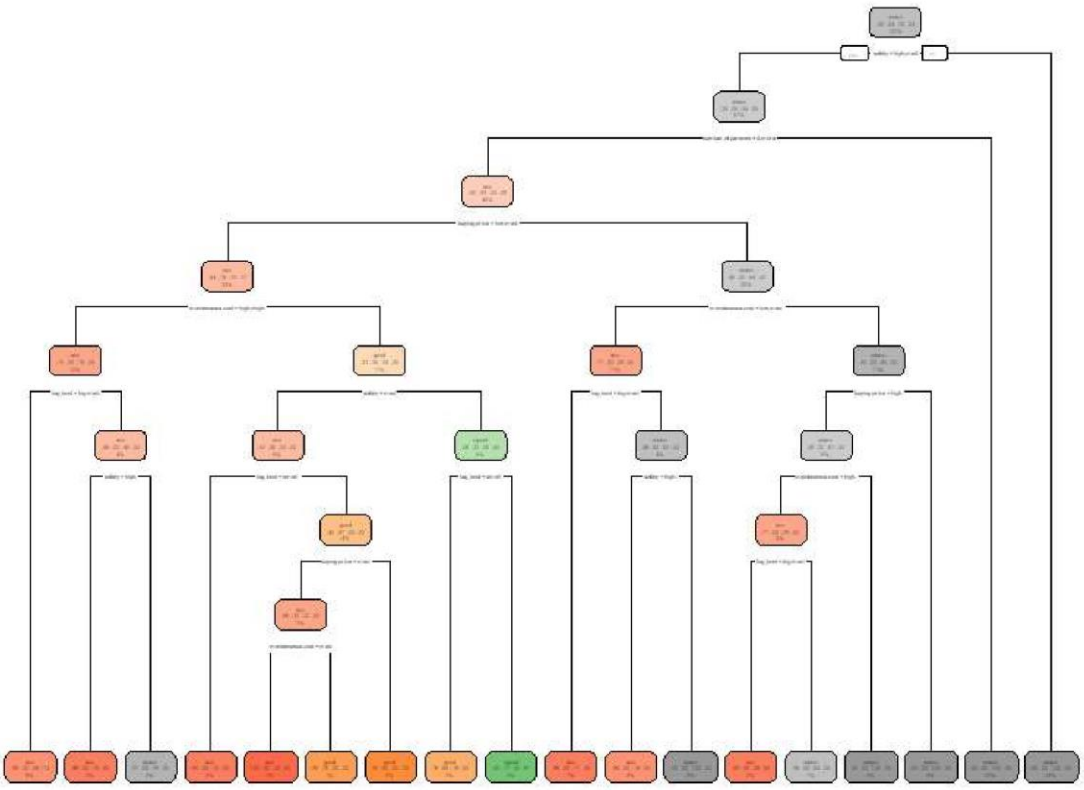
```
    class counts:    13     0     0     0
   probabilities: 1.000 0.000 0.000 0.000

Node number 277: 9 observations
  predicted class=good   expected loss=0.2222222  P(node) =0.007438017
    class counts:     2     7     0     0
   probabilities: 0.222 0.778 0.000 0.000
```

In [21]:
```
library(rpart.plot)
```

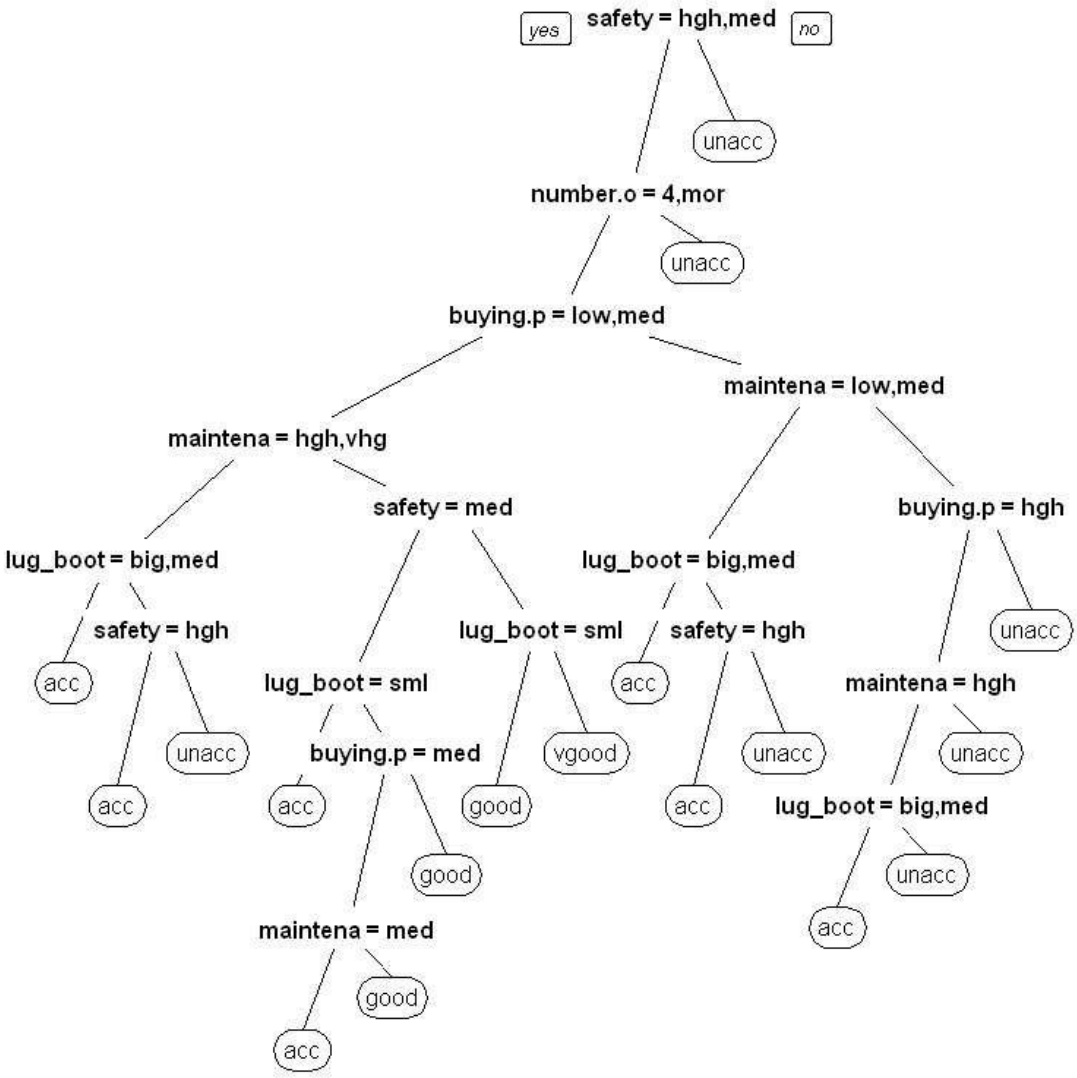
```
Warning message:
"package 'rpart.plot' was built under R version 3.6.3"
```

In [22]:
```
rpart.plot(tree, box.palette='auto')
```



In [ ]:

```
23   prp(tree)
```



In [1]:

In [25]:
```
predict_unseen <-predict(tree, test, type = 'class')
```

In [28]:
```
table_mat <- table(test$decision, predict_unseen)
table_mat
```

```
       predict_unseen
        acc good unacc vgood
  acc   104    5     4     2
  good    0   19     0     2
  unacc  12    0   351     0
  vgood   1    0     0    18
```

Confusion Matrix for the decision tree

In [30]:
```
library(caret)
```

```
Warning message:
"package 'caret' was built under R version 3.6.3"Loading required package: latt
ice
Warning message:
"package 'lattice' was built under R version 3.6.3"
```

In [31]:
```
confusionMatrix(table_mat)
```

```
Confusion Matrix and Statistics

       predict_unseen
        acc good unacc vgood
  acc   104    5     4     2
  good    0   19     0     2
  unacc  12    0   351     0
  vgood   1    0     0    18

Overall Statistics

               Accuracy : 0.9498
                 95% CI : (0.9273, 0.967)
    No Information Rate : 0.6853
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.8923

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: acc Class: good Class: unacc Class: vgood
Sensitivity              0.8889     0.79167       0.9887      0.81818
Specificity              0.9726     0.99595       0.9264      0.99798
Pos Pred Value           0.9043     0.90476       0.9669      0.94737
Neg Pred Value           0.9677     0.98994       0.9742      0.99198
Prevalence               0.2259     0.04633       0.6853      0.04247
Detection Rate           0.2008     0.03668       0.6776      0.03475
Detection Prevalence     0.2220     0.04054       0.7008      0.03668
Balanced Accuracy        0.9307     0.89381       0.9576      0.90808
```

The accuracy is 94.98% obtained from decision tree.

In [1]:

Random Forest

```
2  library(randomForest)
```

```
Warning message:
"package 'randomForest' was built under R version 3.6.3"randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

The following object is masked from 'package:ggplot2':

    margin
```

In [13]:

```
model1 <- randomForest(decision ~ ., data = train,importance = TRUE)
model1
```

```
Call:
 randomForest(formula = decision ~ ., data = train, importance = TRUE)
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 3.47%
Confusion matrix:
      acc good unacc vgood class.error
acc   260    4     4     1  0.03345725
good    9   36     0     3  0.25000000
unacc  15    0   832     0  0.01770956
vgood   5    1     0    40  0.13043478
```

Prediction

In [14]:
```
# Predicting on train set
predTrain <- predict(model1, train, type = "class")
# Checking classification accuracy
table(predTrain, train$decision)
```

```
predTrain acc good unacc vgood
    acc   269    0     1     0
    good    0   48     0     0
    unacc   0    0   846     0
    vgood   0    0     0    46
```

Validation

In [1]:

```
predValid <- predict(model1, test, type = "class")
# Checking classification accuracy
table(predValid,test$decision)
```

```
predValid acc good unacc vgood
    acc   113    0     6     1
    good    0   21     0     1
    unacc   1    0   357     0
    vgood   1    0     0    17
```

While testing on the train dataset, there is 1 misclassification of data, which is given by the diagonal matrix. in test dataset, there are 10 sets which are misclassified.

Accuracy

In [16]:
```
mean(predValid==test$decision)
```

0.980694980694981

The accuracy of the model is 98%

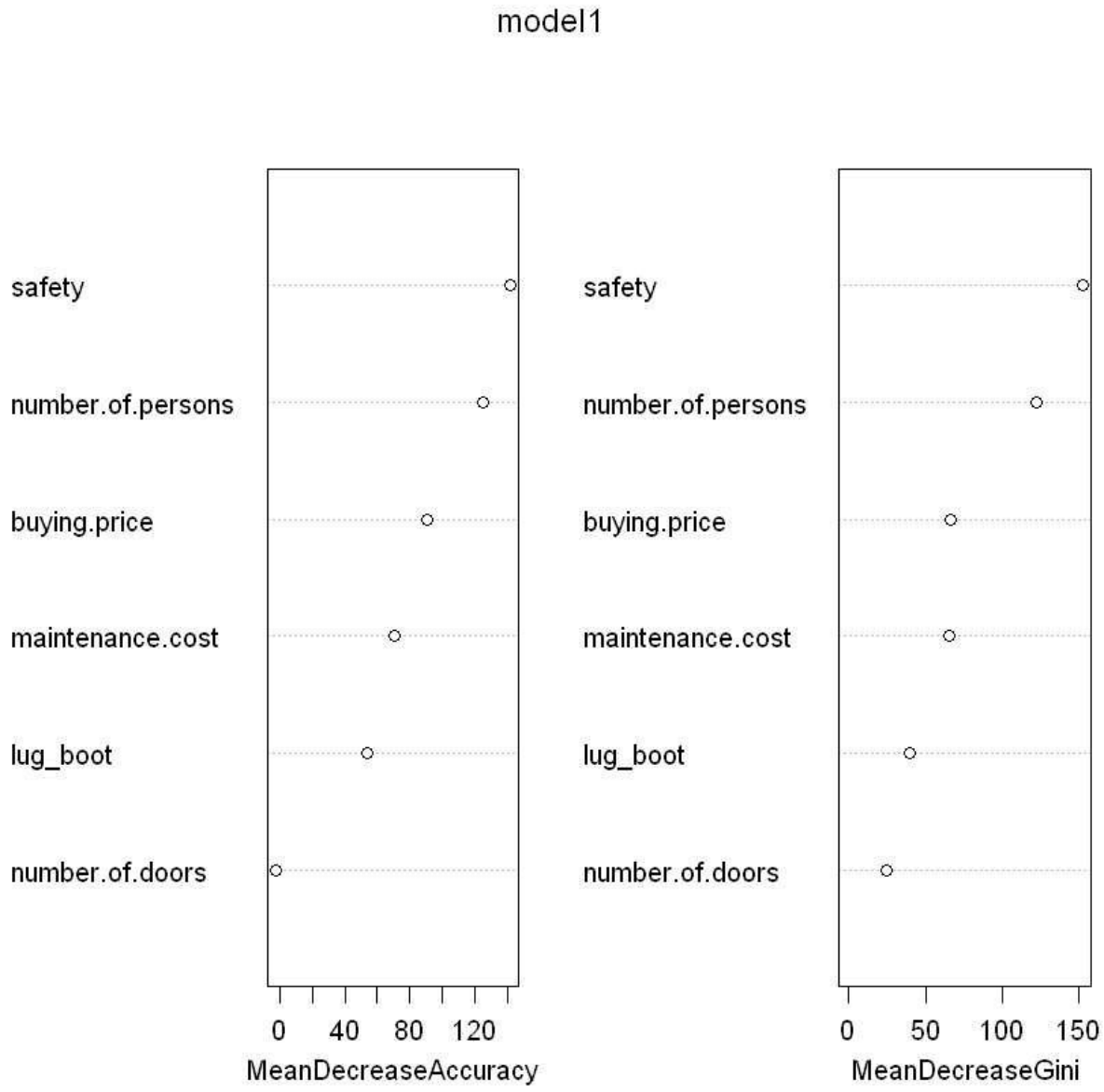
In [17]:
```
#to check important variables
importance(model1)
```

A matrix: 6 × 6 of type dbl

	acc	good	unacc	vgood	MeanDecreaseAccuracy	MeanDecr
buying.price	75.388010	39.538338	59.14934	40.5882097	90.375965	
maintenance.cost	60.966230	38.375714	43.83564	21.2414720	70.801847	
number.of.doors	-4.037215	-3.756878	2.11321	0.5777709	-2.195898	
number.of.persons	92.264022	25.977075	114.93793	28.5122320	124.922191	1
lug_boot	31.611493	23.438396	37.46292	32.0900208	53.485749	
safety	99.783340	42.657037	130.79992	51.6093307	141.639356	1

In [1]:

```
8  varImpPlot(model1)
```



The higher the value of the Mean Decrease Accuracy or Mean Decrease Gini score, the higher the importance of the variable in the model.