In [122…
```
install.packages("gdata")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

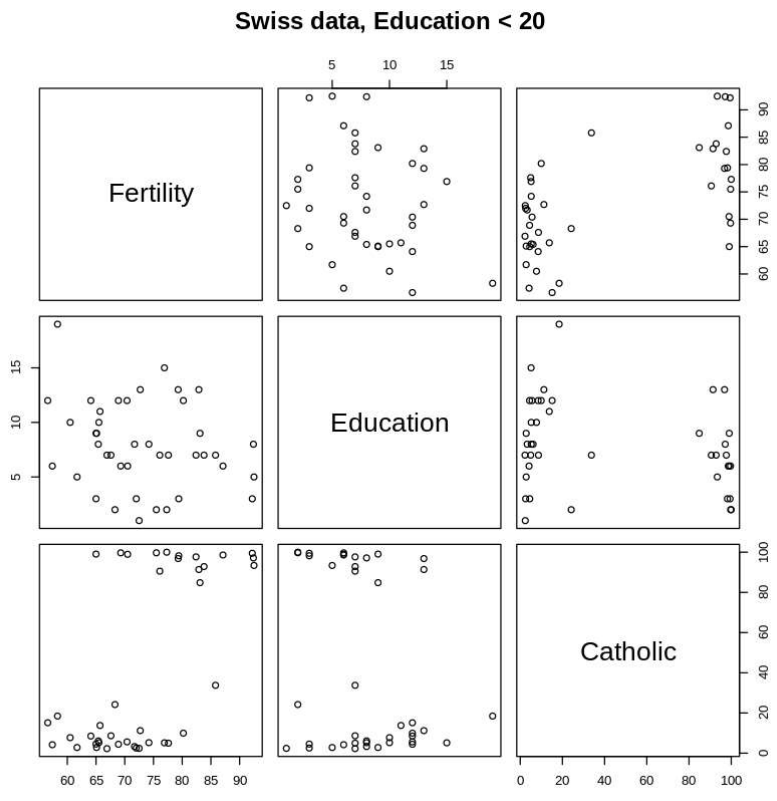
In [124…
```
install.packages("readxl")
library(readxl)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [40]:
```
data(swiss)
sclass <- kmeans(swiss[2:6], 3)
table(sclass$cluster, swiss[,1])
```

```
  35 42.8 44.7 54.3 55.7 56.6 57.4 58.3 60.5 61.7 64.1 64.4 65 65.1 65.4 65.5
1  0    0    0    0    0    0    0    0    0    0    0    0  1    0    0    0
2  0    0    0    0    0    1    1    0    1    1    1    0  1    1    1    1
3  1    1    1    1    1    0    0    1    0    0    0    1  0    0    0    0

  65.7 66.9 67.6 68.3 68.9 69.3 70.4 70.5 71.7 72 72.5 72.7 74.2 75.5 76.1
1    0    0    0    0    0    1    0    1    0  0    0    0    0    1    1
2    0    1    0    1    1    0    1    0    1  1    1    0    1    0    0
3    1    0    1    0    0    0    0    0    0  0    0    1    0    0    0

  76.9 77.3 77.6 79.3 79.4 80.2 82.4 82.9 83.1 83.8 85.8 87.1 92.2 92.4 92.5
1    0    1    0    1    1    0    1    1    1    1    0    1    1    1    1
2    1    0    1    0    0    0    0    0    0    0    0    0    0    0    0
3    0    0    0    0    0    1    0    0    0    0    1    0    0    0    0
```

In [41]:
```
data(swiss)

pairs(~ Fertility + Education + Catholic, data = swiss, subset = Education < 20, main = "Swiss data, Education < 20")
```



In [45]:
```
install.packages("kknn")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘igraph’
```

In [46]:
```
require(kknn)
data(ionosphere)
ionosphere.learn <- ionosphere[1:200,]
ionosphere.valid <- ionosphere[-c(1:200),]
fit.kknn <- kknn(class ~ ., ionosphere.learn, ionosphere.valid)
table(ionosphere.valid$class, fit.kknn$fit)
(fit.train1 <- train.kknn(class ~ ., ionosphere.learn, kmax = 15,
        kernel = c("triangular", "rectangular", "epanechnikov", "optimal"), distance = 1))
table(predict(fit.train1, ionosphere.valid), ionosphere.valid$class)
(fit.train2 <- train.kknn(class ~ ., ionosphere.learn, kmax = 15,
        kernel = c("triangular", "rectangular", "epanechnikov", "optimal"), distance = 2))
table(predict(fit.train2, ionosphere.valid), ionosphere.valid$class)
```

```
Loading required package: kknn
```

```
   b   g
b 19   8
g  2 122
```

```
Call:
train.kknn(formula = class ~ ., data = ionosphere.learn, kmax = 15,     distance = 1, kernel = c("triangular", "rectang
ular", "epanechnikov",         "optimal"))

Type of response variable: nominal
Minimal misclassification: 0.12
Best kernel: rectangular
Best k: 2

     b   g
  b 25   4
  g  2 120
Call:
train.kknn(formula = class ~ ., data = ionosphere.learn, kmax = 15,     distance = 2, kernel = c("triangular", "rectang
ular", "epanechnikov",         "optimal"))

Type of response variable: nominal
Minimal misclassification: 0.12
Best kernel: rectangular
Best k: 2

     b   g
  b 20   5
  g  7 119
```

In [50]:

```
require(kknn)
data(iris)
m <- dim(iris)[1]
val <- sample(1:m, size = round(m/3), replace = FALSE,
        prob = rep(1/m, m))
iris.learn <- iris[-val,]
iris.valid <- iris[val,]
iris.kknn <- kknn(Species~., iris.learn, iris.valid, distance = 1,
        kernel = "triangular")
summary(iris.kknn)
fit <- fitted(iris.kknn)
table(iris.valid$Species, fit)
pcol <- as.character(as.numeric(iris.valid$Species))
```

```
Call:
kknn(formula = Species ~ ., train = iris.learn, test = iris.valid,     distance = 1, kernel = "triangular")

Response: "nominal"
          fit prob.setosa prob.versicolor prob.virginica
1  versicolor           0     1.000000000     0.00000000
2  versicolor           0     1.000000000     0.00000000
3   virginica           0     0.000000000     1.00000000
4  versicolor           0     1.000000000     0.00000000
5   virginica           0     0.000000000     1.00000000
6      setosa           1     0.000000000     0.00000000
7  versicolor           0     1.000000000     0.00000000
8      setosa           1     0.000000000     0.00000000
9   virginica           0     0.000000000     1.00000000
10  virginica           0     0.000000000     1.00000000
11     setosa           1     0.000000000     0.00000000
12     setosa           1     0.000000000     0.00000000
13  virginica           0     0.301956143     0.69804386
14 versicolor           0     0.753486374     0.24651363
15  virginica           0     0.000000000     1.00000000
16     setosa           1     0.000000000     0.00000000
17  virginica           0     0.141533438     0.85846656
18 versicolor           0     1.000000000     0.00000000
19     setosa           1     0.000000000     0.00000000
20  virginica           0     0.000000000     1.00000000
21 versicolor           0     0.965302118     0.03469788
22     setosa           1     0.000000000     0.00000000
23  virginica           0     0.000000000     1.00000000
24     setosa           1     0.000000000     0.00000000
25  virginica           0     0.000000000     1.00000000
26  virginica           0     0.000000000     1.00000000
27     setosa           1     0.000000000     0.00000000
28 versicolor           0     1.000000000     0.00000000
29  virginica           0     0.001452844     0.99854716
30  virginica           0     0.354417502     0.64558250
31     setosa           1     0.000000000     0.00000000
32     setosa           1     0.000000000     0.00000000
33  virginica           0     0.016833814     0.98316619
34 versicolor           0     1.000000000     0.00000000
35 versicolor           0     0.565014394     0.43498561
36 versicolor           0     1.000000000     0.00000000
37  virginica           0     0.000000000     1.00000000
38  virginica           0     0.000000000     1.00000000
39 versicolor           0     1.000000000     0.00000000
40 versicolor           0     1.000000000     0.00000000
41     setosa           1     0.000000000     0.00000000
42 versicolor           0     1.000000000     0.00000000
43  virginica           0     0.069872520     0.93012748
44  virginica           0     0.000000000     1.00000000
45     setosa           1     0.000000000     0.00000000
46     setosa           1     0.000000000     0.00000000
47  virginica           0     0.150015038     0.84998496
48  virginica           0     0.000000000     1.00000000
49 versicolor           0     0.589432124     0.41056788
50     setosa           1     0.000000000     0.00000000
```

```
           fit
            setosa versicolor virginica
  setosa         15          0         0
  versicolor      0         14         3
  virginica       0          1        17
```

In [53]:
```
install.packages("partykit")
```

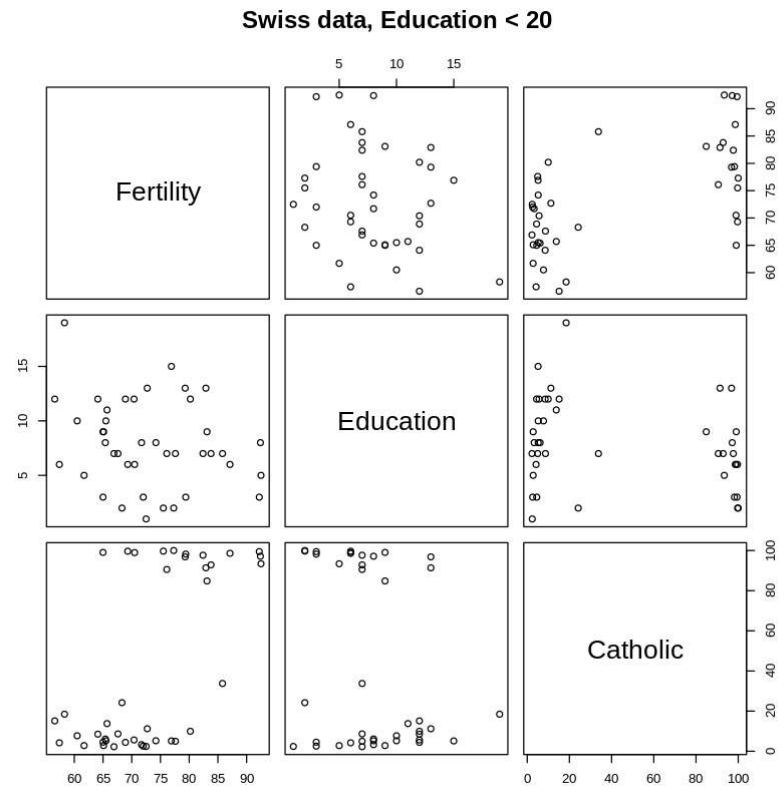
```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘libcoin’, ‘mvtnorm’, ‘Formula’, ‘inum’
```

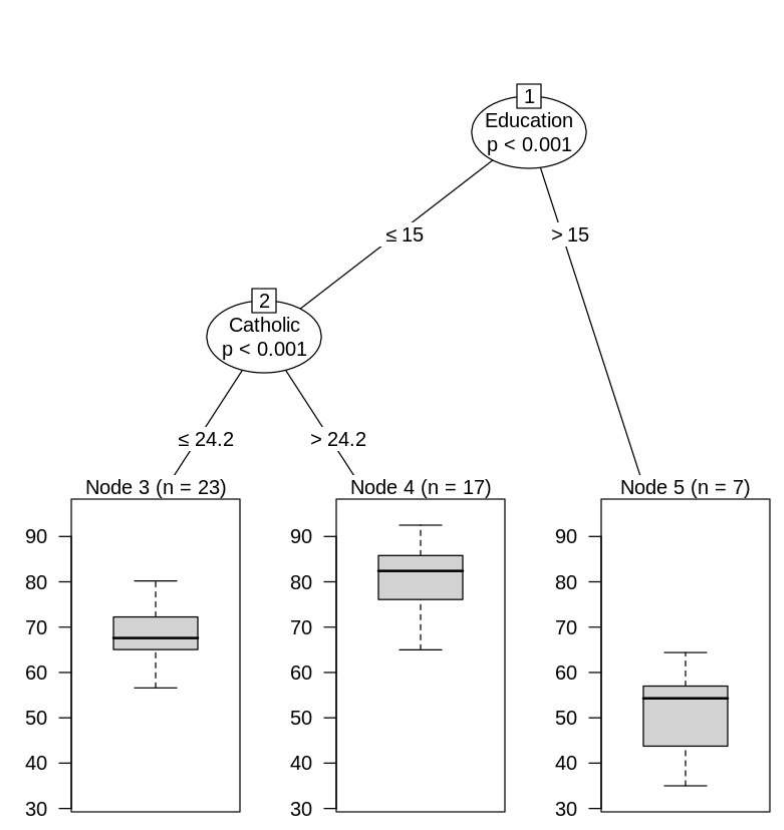
In [58]:
```
install.packages("class")
library(class)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [54]:
```
library(partykit)
pairs(~ Fertility + Education + Catholic, data = swiss, subset = Education < 20, main = "Swiss data, Education < 20")
require(party)
swiss_ctree <- ctree(Fertility ~ Agriculture + Education + Catholic, data = swiss)
plot(swiss_ctree)
```

```
Loading required package: grid

Loading required package: libcoin

Loading required package: mvtnorm

Loading required package: party

Warning message in library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE, :
“there is no package called ‘party’”
```





In [60]:
```
nyt1<-read.csv("nyt1.csv")
nyt1<-nyt1[which(nyt1$Impressions>0 & nyt1$Clicks>0 & nyt1$Age>0),]
nnyt1<-dim(nyt1)[1]            # shrink it down!
sampling.rate=0.9
```

```
num.test.set.labels=nnyt1*(1.-sampling.rate)
training <-sample(1:nnyt1,sampling.rate*nnyt1, replace=FALSE)
train<-subset(nyt1[training,],select=c(Age,Impressions))
testing<-setdiff(1:nnyt1,training)
test<-subset(nyt1[testing,],select=c(Age,Impressions))
cg<-nyt1$Gender[training]
true.labels<-nyt1$Gender[testing]
classif<-knn(train,test,cg,k=5) #
classif
attributes(.Last.value)
```

1·1·1·0·1·0·0·1·1·1·1·1·1·1·0·1·0·1·1·0·0·1·0·0·1·1·1·1·0·1·0·0·1·1·1·0·0·0·1·1·1·1·1·1·0·1·0·
1·1·0·1·0·1·1·1·1·1·1·0·1·1·1·1·1·0·0·1·1·0·0·0·0·0·1·1·1·0·0·1·0·1·1·1·0·1·1·1·0·1·1·0·1·0·1·
1·1·1·1·1·1·0·1·1·1·0·1·1·0·0·0·0·0·0·0·1·0·0·1·1·1·1·0·1·1·0·1·1·0·1·0·1·1·0·0·1·1·0·0·1·1·0·
0·0·1·1·1·1·0·0·0·1·1·1·0·0·1·0·0·1·1·1·1·1·0·0·1·1·1·1·0·1·0·0·0·1·1·1·0·1·0·1·0·1·1·1·1·1·0·1·
0·0·0·0·0·1·0·0·1·1·1·0·…·1·1·1·1·1·1·0·0·1·1·1·0·1·0·0·0·0·0·0·0·1·1·1·1·0·1·1·1·1·1·1·1·0·1·1·
1·1·0·1·1·0·0·1·0·1·0·0·1·0·1·1·0·1·1·1·0·1·1·0·0·1·0·1·0·0·0·0·0·0·1·1·0·1·1·1·1·1·1·1·1·1·0·
1·0·0·1·1·0·1·0·0·1·0·1·1·1·0·0·1·1·1·1·0·0·0·1·0·1·1·0·0·1·1·1·1·1·1·1·0·1·0·1·0·0·1·0·1·1·0·
1·0·1·1·1·1·1·0·1·1·1·0·1·0·1·1·1·0·0·1·1·1·0·0·0·0·1·1·0·0·0·0·1·1·0·0·1·0·1·0·0·1·1·0·0·1·1·
1·0·1·0·0·1·1·0·0·1·0·1·1·0·1·0·0·0·1·1·0·1·1·0·1

▶ **Levels**:

$levels '0' · '1'

$class 'factor'

In [ ]:
```
##############LAB03#########################################################################
```

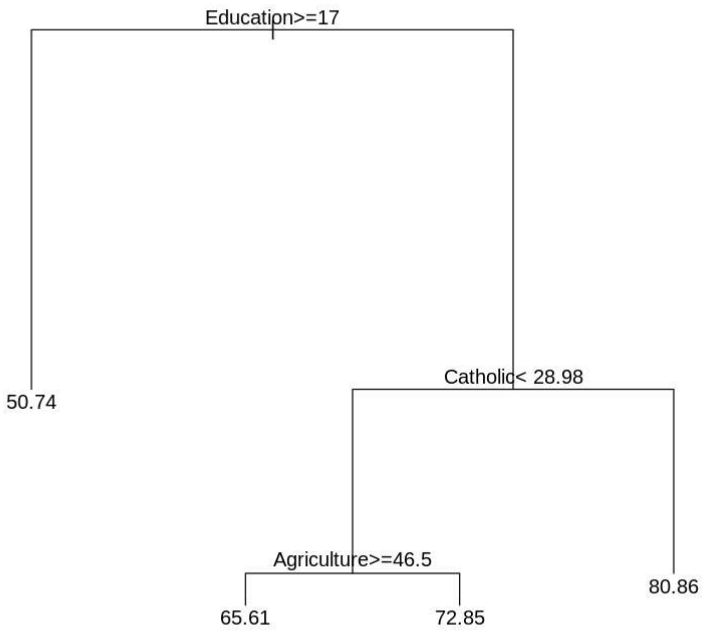
In [63]:
```
install.packages("party")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘TH.data’, ‘matrixStats’, ‘multcomp’, ‘modeltools’, ‘strucchange’, ‘coin’, ‘zoo’, ‘san
dwich’
```

In [65]:
```
install.packages("tree")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

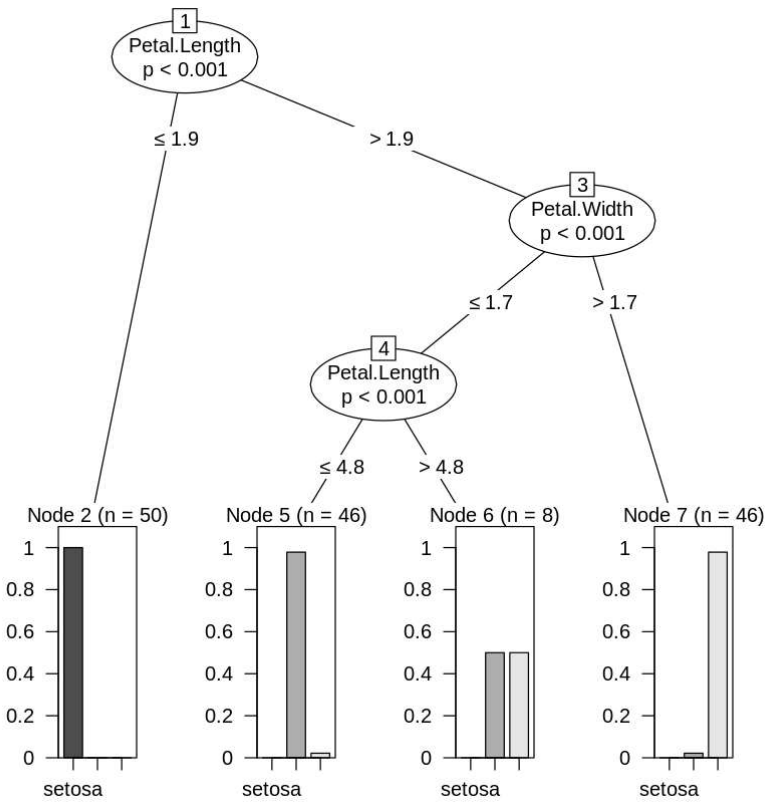
In [66]:
```
library(party)
require(rpart)
swiss_rpart <- rpart(Fertility ~ Agriculture + Education + Catholic, data = swiss)
plot(swiss_rpart) # try some different plot options
text(swiss_rpart) # try some different text options

require(party)

treeSwiss<-ctree(Species ~ ., data=iris)
plot(treeSwiss)

cforest(Species ~ ., data=iris, controls=cforest_control(mtry=2, mincriterion=0))

treeFert<-ctree(Fertility ~ Agriculture + Education + Catholic, data = swiss)

cforest(Fertility ~ Agriculture + Education + Catholic, data = swiss, controls=cforest_control(mtry=2, mincriterion=0))
# look at help info, vary parameters.

library(tree)
tr <- tree(Species ~ ., data=iris)
tr
tr$frame
plot(tr)
text(tr)
#find "prettier" ways to plot the tree
```

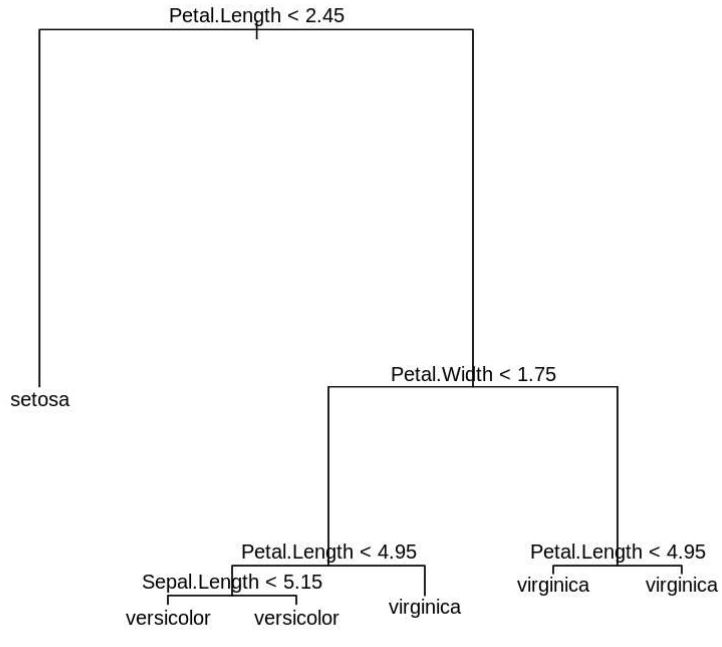



```
         Random Forest using Conditional Inference Trees

Number of trees:  500

Response:  Species
Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
Number of observations:  150
         Random Forest using Conditional Inference Trees

Number of trees:  500

Response:  Fertility
Inputs:  Agriculture, Education, Catholic
Number of observations:  47
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

 1) root 150 329.600 setosa ( 0.33333 0.33333 0.33333 )
   2) Petal.Length < 2.45 50   0.000 setosa ( 1.00000 0.00000 0.00000 ) *
   3) Petal.Length > 2.45 100 138.600 versicolor ( 0.00000 0.50000 0.50000 )
     6) Petal.Width < 1.75 54  33.320 versicolor ( 0.00000 0.90741 0.09259 )
      12) Petal.Length < 4.95 48   9.721 versicolor ( 0.00000 0.97917 0.02083 )
        24) Sepal.Length < 5.15 5   5.004 versicolor ( 0.00000 0.80000 0.20000 ) *
        25) Sepal.Length > 5.15 43   0.000 versicolor ( 0.00000 1.00000 0.00000 ) *
      13) Petal.Length > 4.95 6   7.638 virginica ( 0.00000 0.33333 0.66667 ) *
     7) Petal.Width > 1.75 46   9.635 virginica ( 0.00000 0.02174 0.97826 )
      14) Petal.Length < 4.95 6   5.407 virginica ( 0.00000 0.16667 0.83333 ) *
      15) Petal.Length > 4.95 40   0.000 virginica ( 0.00000 0.00000 1.00000 ) *
```

A data.frame: 11 × 6

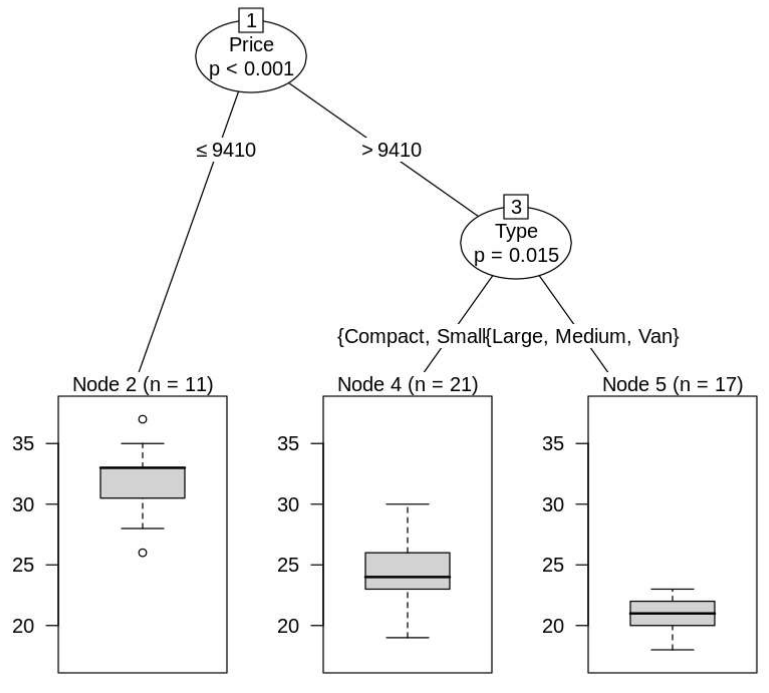
	var	n	dev	yval	splits	yprob
	<fct>	<dbl>	<dbl>	<fct>	<chr[,2]>	<dbl[,3]>
1	Petal.Length	150	329.583687	setosa	<2.45, >2.45	0.3333333, 0.33333333, 0.33333333
2	<leaf>	50	0.000000	setosa	,	1.0000000, 0.00000000, 0.00000000
3	Petal.Width	100	138.629436	versicolor	<1.75, >1.75	0.0000000, 0.50000000, 0.50000000
6	Petal.Length	54	33.317509	versicolor	<4.95, >4.95	0.0000000, 0.90740741, 0.09259259
12	Sepal.Length	48	9.721422	versicolor	<5.15, >5.15	0.0000000, 0.97916667, 0.02083333
24	<leaf>	5	5.004024	versicolor	,	0.0000000, 0.80000000, 0.20000000
25	<leaf>	43	0.000000	versicolor	,	0.0000000, 1.00000000, 0.00000000
13	<leaf>	6	7.638170	virginica	,	0.0000000, 0.33333333, 0.66666667
7	Petal.Length	46	9.635384	virginica	<4.95, >4.95	0.0000000, 0.02173913, 0.97826087
14	<leaf>	6	5.406735	virginica	,	0.0000000, 0.16666667, 0.83333333
15	<leaf>	40	0.000000	virginica	,	0.0000000, 0.00000000, 1.00000000



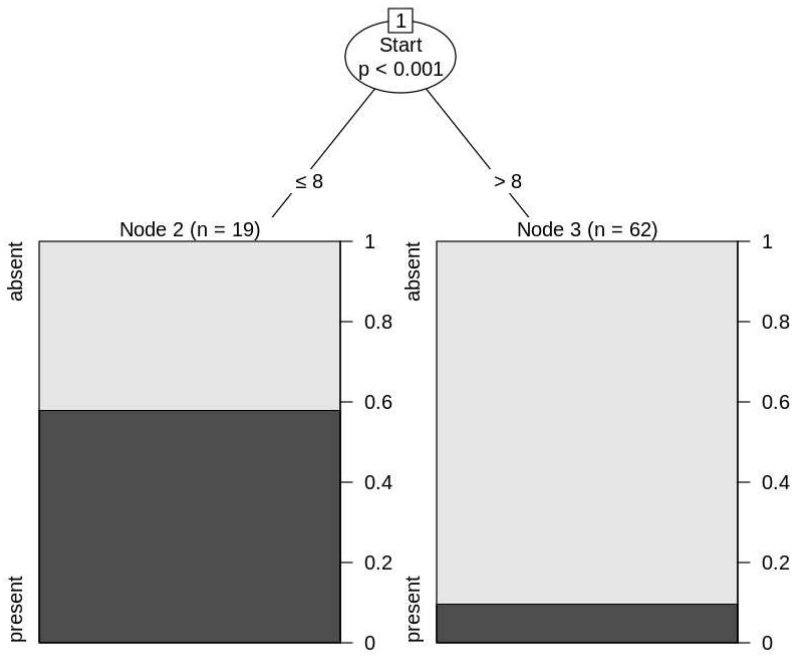


In [68]:
```
fit2M <- ctree(Mileage~Price + Country + Reliability + Type, data=na.omit(cu.summary))
summary(fit2M)
# plot tree
plot(fit2M, uniform=TRUE, main="CI Tree Tree for Mileage ")
```

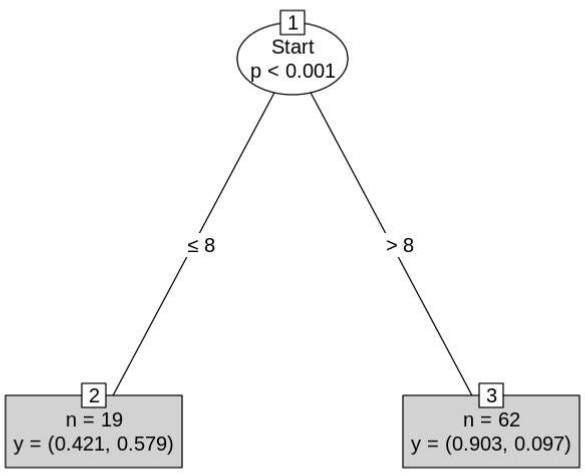
```
    Length      Class       Mode
         1 BinaryTree         S4
```



In [70]:
```
fitK <- ctree(Kyphosis ~ Age + Number + Start, data=kyphosis)
plot(fitK, main="Conditional Inference Tree for Kyphosis")
plot(fitK, main="Conditional Inference Tree for Kyphosis",type="simple")
```



Conditional Inference Tree for Kyphosis



In [71]:

```
require(kknn)
data(iris)
m <- dim(iris)[1]
val <- sample(1:m, size = round(m/3), replace = FALSE, prob = rep(1/m, m))
iris.learn <- iris[-val,]       # train
iris.valid <- iris[val,]        # test
iris.kknn <- train.kknn(Species~., iris.learn, distance = 1, kernel = c("triangular", "epanechnikov", "biweight", "triw
summary(iris.kknn)
table(predict(iris.kknn,iris.valid),iris.valid$Species)

head(iris.kknn$W)
head(iris.kknn$D)
head(iris.kknn$C)
head(iris.kknn$fitted.values)
```

```
Call:
train.kknn(formula = Species ~ ., data = iris.learn, distance = 1,     kernel = c("triangular", "epanechnikov", "biweig
ht", "triweight",         "cos", "inv", "gaussian", "rank", "optimal"))

Type of response variable: nominal
Minimal misclassification: 0.05
Best kernel: triangular
Best k: 8

             setosa versicolor virginica
  setosa         14          0         0
  versicolor      0         15         2
  virginica       0          0        19
NULL
NULL
NULL
```

1. setosa · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica
   ▶ **Levels**:
2. setosa · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica
   ▶ **Levels**:
3. setosa · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica
   ▶ **Levels**:
4. setosa · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica
   ▶ **Levels**:
5. setosa · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica
   ▶ **Levels**:
6. setosa · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · virginica · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica · virginica · virginica · virginica · virginica · virginica · versicolor · virginica
   ▶ **Levels**:

In [75]:
```
install.packages("gpairs")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘lmtest’, ‘barcode’, ‘vcd’
```

In [76]:
```
library(gpairs)

help(gpairs)
```

In [79]:
```
install.packages("mlbench")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [81]:
```
install.packages("naivebayes")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [86]:
```
install.packages("e1071")
install.packages("caTools")
install.packages("caret")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘proxy’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependency ‘bitops’


Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘listenv’, ‘parallelly’, ‘future’, ‘globals’, ‘shape’, ‘future.apply’, ‘numDeriv’, ‘pr
ogressr’, ‘SQUAREM’, ‘diagram’, ‘lava’, ‘prodlim’, ‘iterators’, ‘Rcpp’, ‘clock’, ‘gower’, ‘hardhat’, ‘ipred’, ‘timeDat
e’, ‘foreach’, ‘ModelMetrics’, ‘plyr’, ‘pROC’, ‘recipes’, ‘reshape2’
```

In [89]:
```
library(naivebayes)
```

In [95]:
```
require(mlbench)
data(HouseVotes84)
model <- naive_bayes(Class ~ ., data = HouseVotes84)
predict(model, HouseVotes84[1:10,-1])


pred <- predict(model, HouseVotes84[,-1])
table(pred, HouseVotes84$Class)

## Example of using a contingency table:
data(Titanic)
m <- naive_bayes(Survived ~ ., data = Titanic)
m
predict(m, as.data.frame(Titanic)[,1:3])

## Example with metric predictors:
data(iris)
m <- naive_bayes(Species ~ ., data = iris)
## alternatively:
m <- naive_bayes(iris[,-5], iris[,5])
m
table(predict(m, iris[,-5]), iris[,5])
```

republican · republican · republican · democrat · democrat · democrat · republican · republican · republican · democrat

▶ **Levels**:

```
pred         democrat republican
  democrat        238         13
  republican       29        155
```

```
================================= Naive Bayes ==================================

 Call:
naive_bayes.formula(formula = Survived ~ ., data = Titanic)

--------------------------------------------------------------------------------

Laplace smoothing: 0

--------------------------------------------------------------------------------

 A priori probabilities:

 No Yes
0.5 0.5

--------------------------------------------------------------------------------

 Tables:

--------------------------------------------------------------------------------
 ::: Class (Categorical)
--------------------------------------------------------------------------------

Class    No  Yes
  1st  0.25 0.25
  2nd  0.25 0.25
  3rd  0.25 0.25
  Crew 0.25 0.25

--------------------------------------------------------------------------------
 ::: Sex (Bernoulli)
--------------------------------------------------------------------------------

Sex       No Yes
  Male   0.5 0.5
  Female 0.5 0.5

--------------------------------------------------------------------------------
 ::: Age (Bernoulli)
--------------------------------------------------------------------------------

Age      No Yes
  Child 0.5 0.5
  Adult 0.5 0.5

--------------------------------------------------------------------------------
 ::: Freq (Gaussian)
--------------------------------------------------------------------------------

Freq          No       Yes
  mean  93.12500  44.43750
  sd   183.88071  56.09274

--------------------------------------------------------------------------------
```

```
Warning message:
“predict.naive_bayes(): only 3 feature(s) out of 4 defined in the naive_bayes object "m" are used for prediction.
”
```

No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No

▶ **Levels**:

```
================================== Naive Bayes ==================================

 Call:
naive_bayes.default(x = iris[, -5], y = iris[, 5])

---------------------------------------------------------------------------------

Laplace smoothing: 0

---------------------------------------------------------------------------------

 A priori probabilities:

    setosa versicolor  virginica
 0.3333333  0.3333333  0.3333333

---------------------------------------------------------------------------------

 Tables:

---------------------------------------------------------------------------------
 ::: Sepal.Length (Gaussian)
---------------------------------------------------------------------------------

Sepal.Length    setosa versicolor virginica
        mean 5.0060000  5.9360000 6.5880000
        sd   0.3524897  0.5161711 0.6358796

---------------------------------------------------------------------------------
 ::: Sepal.Width (Gaussian)
---------------------------------------------------------------------------------

Sepal.Width    setosa versicolor virginica
       mean 3.4280000  2.7700000 2.9740000
       sd   0.3790644  0.3137983 0.3224966

---------------------------------------------------------------------------------
 ::: Petal.Length (Gaussian)
---------------------------------------------------------------------------------

Petal.Length    setosa versicolor virginica
        mean 1.4620000  4.2600000 5.5520000
        sd   0.1736640  0.4699110 0.5518947

---------------------------------------------------------------------------------
 ::: Petal.Width (Gaussian)
---------------------------------------------------------------------------------

Petal.Width    setosa versicolor virginica
       mean 0.2460000  1.3260000 2.0260000
       sd   0.1053856  0.1977527 0.2746501

---------------------------------------------------------------------------------

             setosa versicolor virginica
  setosa         50          0         0
  versicolor      0         47         3
  virginica       0          3        47
```

In [96]:
```
data(Titanic)
mdl <- naive_bayes(Survived ~ ., data = Titanic)
mdl
```

```
================================== Naive Bayes ==================================

 Call:
naive_bayes.formula(formula = Survived ~ ., data = Titanic)

---------------------------------------------------------------------------------

Laplace smoothing: 0

---------------------------------------------------------------------------------

 A priori probabilities:

 No Yes
0.5 0.5

---------------------------------------------------------------------------------

 Tables:

---------------------------------------------------------------------------------
 ::: Class (Categorical)
---------------------------------------------------------------------------------

Class    No  Yes
  1st  0.25 0.25
  2nd  0.25 0.25
  3rd  0.25 0.25
  Crew 0.25 0.25

---------------------------------------------------------------------------------
 ::: Sex (Bernoulli)
---------------------------------------------------------------------------------

Sex       No Yes
  Male   0.5 0.5
  Female 0.5 0.5

---------------------------------------------------------------------------------
 ::: Age (Bernoulli)
---------------------------------------------------------------------------------

Age      No Yes
  Child 0.5 0.5
  Adult 0.5 0.5

---------------------------------------------------------------------------------
 ::: Freq (Gaussian)
---------------------------------------------------------------------------------

Freq          No       Yes
  mean  93.12500  44.43750
  sd   183.88071  56.09274

---------------------------------------------------------------------------------
```

In [99]:
```
install.packages("klaR")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘R.methodsS3’, ‘R.oo’, ‘R.utils’, ‘httpuv’, ‘xtable’, ‘sourcetools’, ‘later’, ‘promise
s’, ‘R.cache’, ‘shiny’, ‘miniUI’, ‘styler’, ‘classInt’, ‘labelled’, ‘combinat’, ‘questionr’
```

In [104…
```
require(mlbench)
data(HouseVotes84)
library(klaR)
model <- naive_bayes(Class ~ ., data = HouseVotes84)
predict(model, HouseVotes84[1:10,-1])

pred <- predict(model, HouseVotes84[,-1])
table(pred, HouseVotes84$Class)
```

republican · republican · republican · democrat · democrat · democrat · republican · republican · republican · democrat

▶ **Levels**:

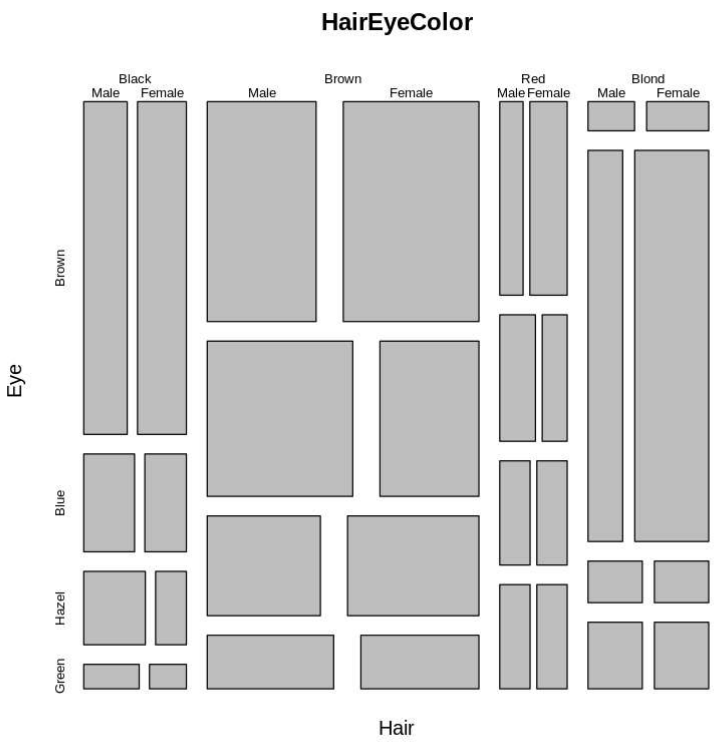
```
pred         democrat republican
  democrat        238         13
  republican       29        155
```

In [106…
```
install.packages('ElemStatLearn')
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Warning message:
“package ‘ElemStatLearn’ is not available for this version of R

A version of this package for your version of R might be available elsewhere,
see the ideas at
https://cran.r-project.org/doc/manuals/r-patched/R-admin.html#Installing-packages”
```

In [111…
```
data(HairEyeColor)
mosaicplot(HairEyeColor)
margin.table(HairEyeColor,3)
margin.table(HairEyeColor,c(1,3))
```

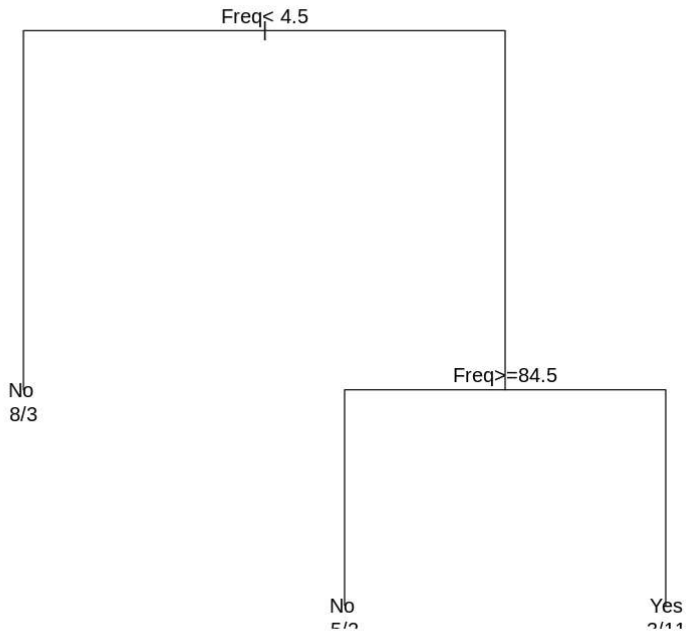
```
Sex
  Male Female
   279    313
       Sex
Hair    Male Female
  Black   56     52
  Brown  143    143
  Red     34     37
  Blond   46     81
```



In [112…
```
titanic_df <- as.data.frame(Titanic)
```

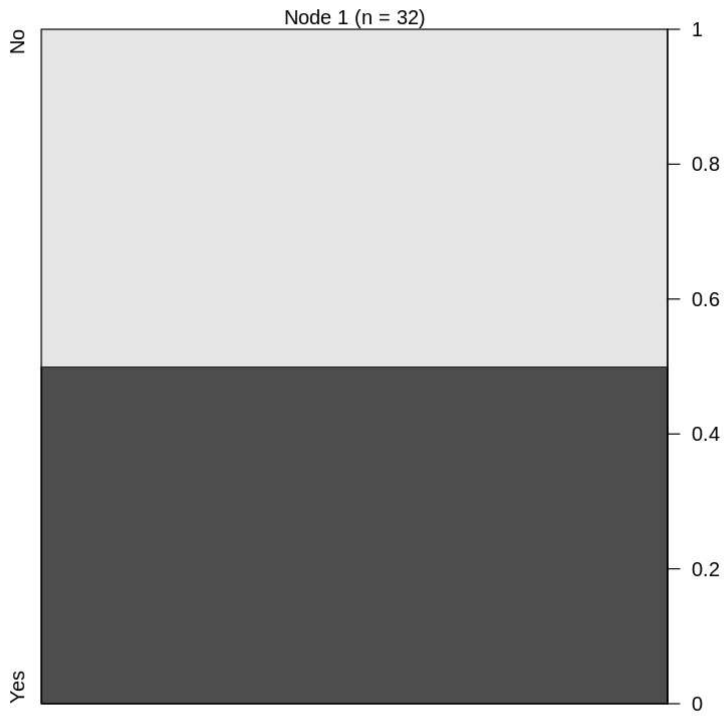
In [113…
```
install.packages("rpart")
library(rpart)

fit_rpart <- rpart(Survived ~ ., data = titanic_df, method = "class")

# Plot the decision tree
plot(fit_rpart)
text(fit_rpart, use.n = TRUE)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```



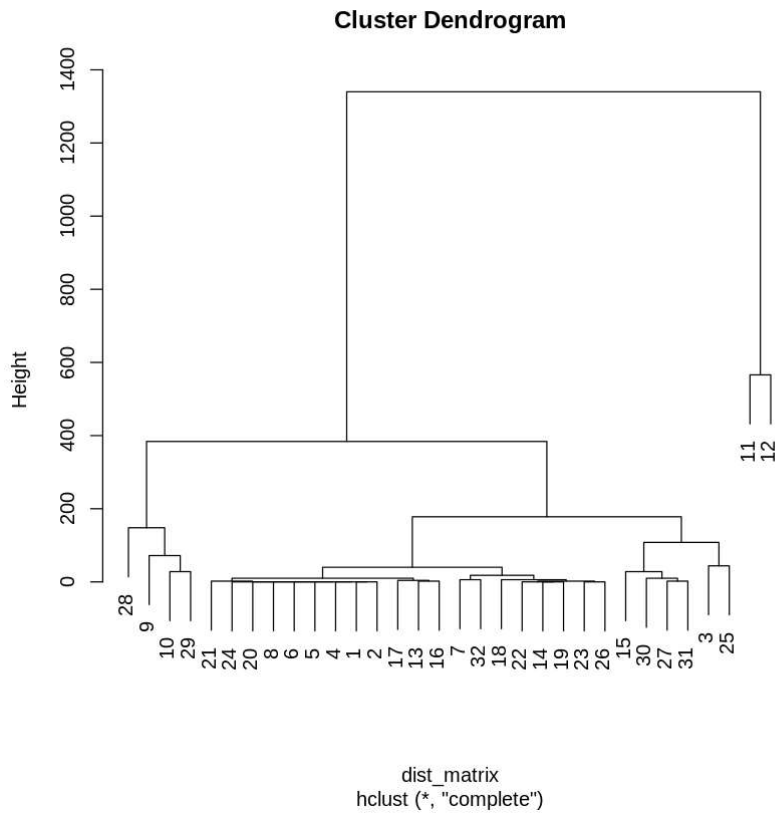
In [114…
```
install.packages("partykit")
library(partykit)


fit_ctree <- ctree(Survived ~ ., data = titanic_df)


plot(fit_ctree)
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```



In [115…
```
data_matrix <- as.matrix(titanic_df[, -which(names(titanic_df) == "Survived")])


dist_matrix <- dist(data_matrix)

fit_hclust <- hclust(dist_matrix)


plot(fit_hclust)
```

```
Warning message in dist(data_matrix):
“NAs introduced by coercion”
```



In [ ]:
```

```

In [7]:
```
install.packages("gdata")
install.packages("readxl")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [27]:
```
install.packages("xlsx")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)

also installing the dependencies ‘rJava’, ‘xlsxjars’


Warning message in install.packages("xlsx"):
“installation of package ‘rJava’ had non-zero exit status”
Warning message in install.packages("xlsx"):
“installation of package ‘xlsxjars’ had non-zero exit status”
Warning message in install.packages("xlsx"):
“installation of package ‘xlsx’ had non-zero exit status”
```

In [117…
```
install.packages("ISLR")
```

```
Installing package into ‘/usr/local/lib/R/site-library’
(as ‘lib’ is unspecified)
```

In [118…

```r
# KNN example using ISLR package (Textbook)
library(ISLR) # you need to install the ISLR package first
# Caravan dataset is about the insurance
library(class)
head(Caravan)
str(Caravan)
# Purchase: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
# "Yes" or "No" indicates wheater people purchased the insurance policy or not.
dim(Caravan) # it has 5822 rows (observations) and 86 features/columns
summary(Caravan)
summary(Caravan$Purchase)
# Yes: 348 people purchased the insurance and No:5474 people did not purchase the insurance
# check for any NA and missing values
any(is.na(Caravan)) # FALSE, that means no NA values in this dataset
# Now we want to check the scales of the variables becuase the variable with a large scale
# will have a larger effect on distances between observations when using the KNN Algorithm
# Checking the Variances on features/Colums using the var() function in R
var(Caravan[,1]) # Variance of the 1st column is 165.0378
var(Caravan[,2]) # Variance of the 2nd column is 0.1647
var(Caravan[,3]) # Variance of the 3rd column is 0.6238
# You can see that the variances are different for each of the column variables, 1st one is 165.03 and 2nd one is 0.16
# and there is a huge difference, because of that we want to standardize the variables.
# We will do that for all the columns except the "Purchase" variable which we are going predict.
purchase <- Caravan[,86] # you can write the same as
# purchase <-Caravan[,'Purchase'] with the column name, we use the column number 86 for the simplcity.
purchase
# Now we want to Standardize the columns except the 86th column
StandardizedCaravan <- scale(Caravan[,-86]) # when we use -86 it will not include the 86th column.
var(StandardizedCaravan[,1])
var(StandardizedCaravan[,2])
var(StandardizedCaravan[,3])

# test set
test_index <- 1:1000
test_data <- StandardizedCaravan[test_index,]
test_purchase <- purchase[test_index]

# train set
train_data <- StandardizedCaravan[-test_index,]
train_purchase <- purchase[-test_index]

# set seed
set.seed(101)
predicted_purchase <- knn(train_data,test_data,train_purchase, k = 10)
head(predicted_purchase)

missClassError <- mean(test_purchase != predicted_purchase)
print(missClassError)

# Choosing the K value
# we can write a for-loop
predicted_purchase <- NULL
error_rate <- NULL

for (i in 1:20) {
  set.seed(101)
  predicted_purchase <- knn(train_data, test_data, train_purchase, k =i)
  error_rate[i] <- mean(test_purchase != predicted_purchase)
}

print(error_rate)

# Plot the K value on a graph
library(ggplot2)
k_values <- 1:20

error_df <- data.frame(error_rate, k_values)
print(error_df)
ggplot(error_df,aes(k_values,error_rate)) + geom_point() + geom_line(lty='dotted', color='blue')
```

A data.frame: 6 × 86

	MOSTYPE	MAANTHUI	MGEMOMV	MGEMLEEF	MOSHOOFD	MGODRK	MGODPR	MGODOV	MGODGE	MRELGE	…	APERSONG	AGEZ
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	…	<dbl>	<
1	33	1	3	2	8	0	5	1	3	7	…	0	
2	37	1	2	2	8	1	4	1	4	6	…	0	
3	37	1	2	2	8	0	4	2	4	3	…	0	
4	9	1	3	3	3	2	3	2	4	5	…	0	
5	40	1	4	2	10	1	4	1	4	7	…	0	
6	23	1	2	1	5	0	5	0	5	0	…	0	

```
'data.frame':   5822 obs. of  86 variables:
 $ MOSTYPE : num  33 37 37 9 40 23 39 33 33 11 ...
 $ MAANTHUI: num  1 1 1 1 1 1 2 1 1 2 ...
 $ MGEMOMV : num  3 2 2 3 4 2 3 2 2 3 ...
 $ MGEMLEEF: num  2 2 2 3 2 1 2 3 4 3 ...
 $ MOSHOOFD: num  8 8 8 3 10 5 9 8 8 3 ...
 $ MGODRK  : num  0 1 0 2 1 0 2 0 0 3 ...
 $ MGODPR  : num  5 4 4 3 4 5 2 7 1 5 ...
 $ MGODOV  : num  1 1 2 2 1 0 0 0 3 0 ...
 $ MGODGE  : num  3 4 4 4 4 5 5 2 6 2 ...
 $ MRELGE  : num  7 6 3 5 7 0 7 7 6 7 ...
 $ MRELSA  : num  0 2 2 2 1 6 2 2 0 0 ...
 $ MRELOV  : num  2 2 4 2 2 3 0 0 3 2 ...
 $ MFALLEEN: num  1 0 4 2 2 3 0 0 3 2 ...
 $ MFGEKIND: num  2 4 4 3 4 5 3 5 3 2 ...
 $ MFWEKIND: num  6 5 2 4 4 2 6 4 3 6 ...
 $ MOPLHOOG: num  1 0 0 3 5 0 0 0 0 0 ...
 $ MOPLMIDD: num  2 5 5 4 4 5 4 3 1 4 ...
 $ MOPLLAAG: num  7 4 4 2 0 4 5 6 8 5 ...
 $ MBERHOOG: num  1 0 0 4 0 2 0 2 1 2 ...
 $ MBERZELF: num  0 0 0 0 5 0 0 0 1 0 ...
 $ MBERBOER: num  1 0 0 0 4 0 0 0 0 0 ...
 $ MBERMIDD: num  2 5 7 3 0 4 4 2 1 3 ...
 $ MBERARBG: num  5 0 0 1 0 2 1 5 8 3 ...
 $ MBERARBO: num  2 4 2 2 0 2 5 2 1 3 ...
 $ MSKA    : num  1 0 0 3 9 2 0 2 1 1 ...
 $ MSKB1   : num  1 2 5 2 0 2 1 1 1 2 ...
 $ MSKB2   : num  2 3 0 1 0 2 4 2 0 1 ...
 $ MSKC    : num  6 5 4 4 0 4 5 5 8 4 ...
 $ MSKD    : num  1 0 0 0 0 2 0 2 1 2 ...
 $ MHHUUR  : num  1 2 7 5 4 9 6 0 9 0 ...
 $ MHKOOP  : num  8 7 2 4 5 0 3 9 0 9 ...
 $ MAUT1   : num  8 7 7 9 6 5 8 4 5 6 ...
 $ MAUT2   : num  0 1 0 0 2 3 0 4 2 1 ...
 $ MAUT0   : num  1 2 2 0 1 3 1 2 3 2 ...
 $ MZFONDS : num  8 6 9 7 5 9 9 6 7 6 ...
 $ MZPART  : num  1 3 0 2 4 0 0 3 2 3 ...
 $ MINKM30 : num  0 2 4 1 0 5 4 2 7 2 ...
 $ MINK3045: num  4 0 5 5 0 2 3 5 2 3 ...
 $ MINK4575: num  5 5 0 3 9 3 3 3 1 3 ...
 $ MINK7512: num  0 2 0 0 0 0 0 0 0 1 ...
 $ MINK123M: num  0 0 0 0 0 0 0 0 0 0 ...
 $ MINKGEM : num  4 5 3 4 6 3 3 3 2 4 ...
 $ MKOOPKLA: num  3 4 4 4 3 3 5 3 3 7 ...
 $ PWAPART : num  0 2 2 0 0 0 0 0 0 2 ...
 $ PWABEDR : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PWALAND : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PPERSAUT: num  6 0 6 6 0 6 6 0 5 0 ...
 $ PBESAUT : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PMOTSCO : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PVRAAUT : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PAANHANG: num  0 0 0 0 0 0 0 0 0 0 ...
 $ PTRACTOR: num  0 0 0 0 0 0 0 0 0 0 ...
 $ PWERKT  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PBROM   : num  0 0 0 0 0 0 0 3 0 0 ...
 $ PLEVEN  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PPERSONG: num  0 0 0 0 0 0 0 0 0 0 ...
 $ PGEZONG : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PWAOREG : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PBRAND  : num  5 2 2 2 6 0 0 0 0 3 ...
 $ PZEILPL : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PPLEZIER: num  0 0 0 0 0 0 0 0 0 0 ...
 $ PFIETS  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PINBOED : num  0 0 0 0 0 0 0 0 0 0 ...
 $ PBYSTAND: num  0 0 0 0 0 0 0 0 0 0 ...
 $ AWAPART : num  0 2 1 0 0 0 0 0 0 1 ...
 $ AWABEDR : num  0 0 0 0 0 0 0 0 0 0 ...
 $ AWALAND : num  0 0 0 0 0 0 0 0 0 0 ...
 $ APERSAUT: num  1 0 1 1 0 1 1 0 1 0 ...
 $ ABESAUT : num  0 0 0 0 0 0 0 0 0 0 ...
 $ AMOTSCO : num  0 0 0 0 0 0 0 0 0 0 ...
 $ AVRAAUT : num  0 0 0 0 0 0 0 0 0 0 ...
 $ AAANHANG: num  0 0 0 0 0 0 0 0 0 0 ...
 $ ATRACTOR: num  0 0 0 0 0 0 0 0 0 0 ...
 $ AWERKT  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ABROM   : num  0 0 0 0 0 0 0 1 0 0 ...
 $ ALEVEN  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ APERSONG: num  0 0 0 0 0 0 0 0 0 0 ...
 $ AGEZONG : num  0 0 0 0 0 0 0 0 0 0 ...
 $ AWAOREG : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ABRAND  : num  1 1 1 1 1 0 0 0 0 1 ...
 $ AZEILPL : num  0 0 0 0 0 0 0 0 0 0 ...
 $ APLEZIER: num  0 0 0 0 0 0 0 0 0 0 ...
 $ AFIETS  : num  0 0 0 0 0 0 0 0 0 0 ...
 $ AINBOED : num  0 0 0 0 0 0 0 0 0 0 ...
 $ ABYSTAND: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Purchase: Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 1 1 1 1 ...
5822·86
```

```
   MOSTYPE         MAANTHUI        MGEMOMV         MGEMLEEF
Min.   : 1.00   Min.   : 1.000   Min.   :1.000   Min.   :1.000
1st Qu.:10.00   1st Qu.: 1.000   1st Qu.:2.000   1st Qu.:2.000
Median :30.00   Median : 1.000   Median :3.000   Median :3.000
Mean   :24.25   Mean   : 1.111   Mean   :2.679   Mean   :2.991
3rd Qu.:35.00   3rd Qu.: 1.000   3rd Qu.:3.000   3rd Qu.:3.000
Max.   :41.00   Max.   :10.000   Max.   :5.000   Max.   :6.000
   MOSHOOFD         MGODRK          MGODPR          MGODOV
Min.   : 1.000   Min.   :0.0000   Min.   :0.000   Min.   :0.00
1st Qu.: 3.000   1st Qu.:0.0000   1st Qu.:4.000   1st Qu.:0.00
Median : 7.000   Median :0.0000   Median :5.000   Median :1.00
Mean   : 5.774   Mean   :0.6965   Mean   :4.627   Mean   :1.07
3rd Qu.: 8.000   3rd Qu.:1.0000   3rd Qu.:6.000   3rd Qu.:2.00
Max.   :10.000   Max.   :9.0000   Max.   :9.000   Max.   :5.00
    MGODGE          MRELGE          MRELSA          MRELOV
Min.   :0.000   Min.   :0.000   Min.   :0.0000   Min.   :0.00
1st Qu.:2.000   1st Qu.:5.000   1st Qu.:0.0000   1st Qu.:1.00
Median :3.000   Median :6.000   Median :1.0000   Median :2.00
Mean   :3.259   Mean   :6.183   Mean   :0.8835   Mean   :2.29
3rd Qu.:4.000   3rd Qu.:7.000   3rd Qu.:1.0000   3rd Qu.:3.00
Max.   :9.000   Max.   :9.000   Max.   :7.0000   Max.   :9.00
   MFALLEEN        MFGEKIND        MFWEKIND       MOPLHOOG        MOPLMIDD
Min.   :0.000   Min.   :0.00   Min.   :0.0   Min.   :0.000   Min.   :0.000
1st Qu.:0.000   1st Qu.:2.00   1st Qu.:3.0   1st Qu.:0.000   1st Qu.:2.000
Median :2.000   Median :3.00   Median :4.0   Median :1.000   Median :3.000
Mean   :1.888   Mean   :3.23   Mean   :4.3   Mean   :1.461   Mean   :3.351
3rd Qu.:3.000   3rd Qu.:4.00   3rd Qu.:6.0   3rd Qu.:2.000   3rd Qu.:4.000
Max.   :9.000   Max.   :9.00   Max.   :9.0   Max.   :9.000   Max.   :9.000
   MOPLLAAG        MBERHOOG        MBERZELF        MBERBOER
Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.0000
1st Qu.:3.000   1st Qu.:0.000   1st Qu.:0.000   1st Qu.:0.0000
Median :5.000   Median :2.000   Median :0.000   Median :0.0000
Mean   :4.572   Mean   :1.895   Mean   :0.398   Mean   :0.5223
3rd Qu.:6.000   3rd Qu.:3.000   3rd Qu.:1.000   3rd Qu.:1.0000
Max.   :9.000   Max.   :9.000   Max.   :5.000   Max.   :9.0000
   MBERMIDD        MBERARBG        MBERARBO          MSKA           MSKB1
Min.   :0.000   Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :0.000
1st Qu.:2.000   1st Qu.:1.00   1st Qu.:1.000   1st Qu.:0.000   1st Qu.:1.000
Median :3.000   Median :2.00   Median :2.000   Median :1.000   Median :2.000
Mean   :2.899   Mean   :2.22   Mean   :2.306   Mean   :1.621   Mean   :1.607
3rd Qu.:4.000   3rd Qu.:3.00   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:2.000
Max.   :9.000   Max.   :9.00   Max.   :9.000   Max.   :9.000   Max.   :9.000
    MSKB2           MSKC            MSKD          MHHUUR
Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
1st Qu.:1.000   1st Qu.:2.000   1st Qu.:0.000   1st Qu.:2.000
Median :2.000   Median :4.000   Median :1.000   Median :4.000
Mean   :2.203   Mean   :3.759   Mean   :1.067   Mean   :4.237
3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.:2.000   3rd Qu.:7.000
Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
    MHKOOP          MAUT1           MAUT2           MAUT0           MZFONDS
Min.   :0.000   Min.   :0.00   Min.   :0.000   Min.   :0.000   Min.   :0.000
1st Qu.:2.000   1st Qu.:5.00   1st Qu.:0.000   1st Qu.:1.000   1st Qu.:5.000
Median :5.000   Median :6.00   Median :1.000   Median :2.000   Median :7.000
Mean   :4.772   Mean   :6.04   Mean   :1.316   Mean   :1.959   Mean   :6.277
3rd Qu.:7.000   3rd Qu.:7.00   3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.:8.000
Max.   :9.000   Max.   :9.00   Max.   :7.000   Max.   :9.000   Max.   :9.000
    MZPART          MINKM30         MINK3045        MINK4575
Min.   :0.000   Min.   :0.000   Min.   :0.000   Min.   :0.000
1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000
Median :2.000   Median :2.000   Median :4.000   Median :3.000
Mean   :2.729   Mean   :2.574   Mean   :3.536   Mean   :2.731
3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:5.000   3rd Qu.:4.000
Max.   :9.000   Max.   :9.000   Max.   :9.000   Max.   :9.000
   MINK7512        MINK123M          MINKGEM         MKOOPKLA
Min.   :0.0000   Min.   :0.0000   Min.   :0.000   Min.   :1.000
1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:3.000
Median :0.0000   Median :0.0000   Median :4.000   Median :4.000
Mean   :0.7961   Mean   :0.2027   Mean   :3.784   Mean   :4.236
3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:4.000   3rd Qu.:6.000
Max.   :9.0000   Max.   :9.0000   Max.   :9.000   Max.   :8.000
   PWAPART          PWABEDR           PWALAND           PPERSAUT
Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00
1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00
Median :0.0000   Median :0.00000   Median :0.00000   Median :5.00
Mean   :0.7712   Mean   :0.04002   Mean   :0.07162   Mean   :2.97
3rd Qu.:2.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:6.00
Max.   :3.0000   Max.   :6.00000   Max.   :4.00000   Max.   :8.00
   PBESAUT           PMOTSCO          PVRAAUT           PAANHANG
Min.   :0.00000   Min.   :0.0000   Min.   :0.000000   Min.   :0.00000
1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.000000   1st Qu.:0.00000
Median :0.00000   Median :0.0000   Median :0.000000   Median :0.00000
Mean   :0.04827   Mean   :0.1754   Mean   :0.009447   Mean   :0.02096
3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.000000   3rd Qu.:0.00000
Max.   :7.00000   Max.   :7.0000   Max.   :9.000000   Max.   :5.00000
   PTRACTOR           PWERKT           PBROM           PLEVEN
Min.   :0.00000   Min.   :0.00000   Min.   :0.000   Min.   :0.0000
1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:0.0000
Median :0.00000   Median :0.00000   Median :0.000   Median :0.0000
Mean   :0.09258   Mean   :0.01305   Mean   :0.215   Mean   :0.1948
3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000   3rd Qu.:0.0000
Max.   :6.00000   Max.   :6.00000   Max.   :6.000   Max.   :9.0000
   PPERSONG          PGEZONG           PWAOREG           PBRAND
Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000
1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000
Median :0.00000   Median :0.00000   Median :0.00000   Median :2.000
```

```
 Mean   :0.01374   Mean   :0.01529   Mean   :0.02353   Mean   :1.828  
 3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:4.000  
 Max.   :6.00000   Max.   :3.00000   Max.   :7.00000   Max.   :8.000  
    PZEILPL             PPLEZIER           PFIETS            PINBOED       
 Min.   :0.0000000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000  
 1st Qu.:0.0000000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000  
 Median :0.0000000   Median :0.00000   Median :0.00000   Median :0.00000  
 Mean   :0.0008588   Mean   :0.01889   Mean   :0.02525   Mean   :0.01563  
 3rd Qu.:0.0000000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000  
 Max.   :3.0000000   Max.   :6.00000   Max.   :1.00000   Max.   :6.00000  
    PBYSTAND          AWAPART         AWABEDR           AWALAND       
 Min.   :0.00000   Min.   :0.000   Min.   :0.00000   Min.   :0.00000  
 1st Qu.:0.00000   1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.00000  
 Median :0.00000   Median :0.000   Median :0.00000   Median :0.00000  
 Mean   :0.04758   Mean   :0.403   Mean   :0.01477   Mean   :0.02061  
 3rd Qu.:0.00000   3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.00000  
 Max.   :5.00000   Max.   :2.000   Max.   :5.00000   Max.   :1.00000  
    APERSAUT         ABESAUT            AMOTSCO            AVRAAUT        
 Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.000000  
 1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000  
 Median :1.0000   Median :0.00000   Median :0.00000   Median :0.000000  
 Mean   :0.5622   Mean   :0.01048   Mean   :0.04105   Mean   :0.002233  
 3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000000  
 Max.   :7.0000   Max.   :4.00000   Max.   :8.00000   Max.   :3.000000  
    AAANHANG          ATRACTOR           AWERKT              ABROM        
 Min.   :0.00000   Min.   :0.00000   Min.   :0.000000   Min.   :0.00000  
 1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.00000  
 Median :0.00000   Median :0.00000   Median :0.000000   Median :0.00000  
 Mean   :0.01254   Mean   :0.03367   Mean   :0.006183   Mean   :0.07042  
 3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.00000  
 Max.   :3.00000   Max.   :4.00000   Max.   :6.000000   Max.   :2.00000  
     ALEVEN           APERSONG            AGEZONG             AWAOREG         
 Min.   :0.00000   Min.   :0.000000   Min.   :0.000000   Min.   :0.000000  
 1st Qu.:0.00000   1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:0.000000  
 Median :0.00000   Median :0.000000   Median :0.000000   Median :0.000000  
 Mean   :0.07661   Mean   :0.005325   Mean   :0.006527   Mean   :0.004638  
 3rd Qu.:0.00000   3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:0.000000  
 Max.   :8.00000   Max.   :1.000000   Max.   :1.000000   Max.   :2.000000  
     ABRAND          AZEILPL              APLEZIER             AFIETS       
 Min.   :0.0000   Min.   :0.0000000   Min.   :0.000000   Min.   :0.00000  
 1st Qu.:0.0000   1st Qu.:0.0000000   1st Qu.:0.000000   1st Qu.:0.00000  
 Median :1.0000   Median :0.0000000   Median :0.000000   Median :0.00000  
 Mean   :0.5701   Mean   :0.0005153   Mean   :0.006012   Mean   :0.03178  
 3rd Qu.:1.0000   3rd Qu.:0.0000000   3rd Qu.:0.000000   3rd Qu.:0.00000  
 Max.   :7.0000   Max.   :1.0000000   Max.   :2.000000   Max.   :3.00000  
    AINBOED            ABYSTAND        Purchase  
 Min.   :0.000000   Min.   :0.00000   No :5474  
 1st Qu.:0.000000   1st Qu.:0.00000   Yes: 348  
 Median :0.000000   Median :0.00000             
 Mean   :0.007901   Mean   :0.01426             
 3rd Qu.:0.000000   3rd Qu.:0.00000             
 Max.   :2.000000   Max.   :2.00000             
```

No: 5474 **Yes:** 348

FALSE

165.037847395189

0.164707781931954

0.623838598103468

No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · ⋯ · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · Yes · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · No · Yes · No · No

▶ **Levels**:

1

1

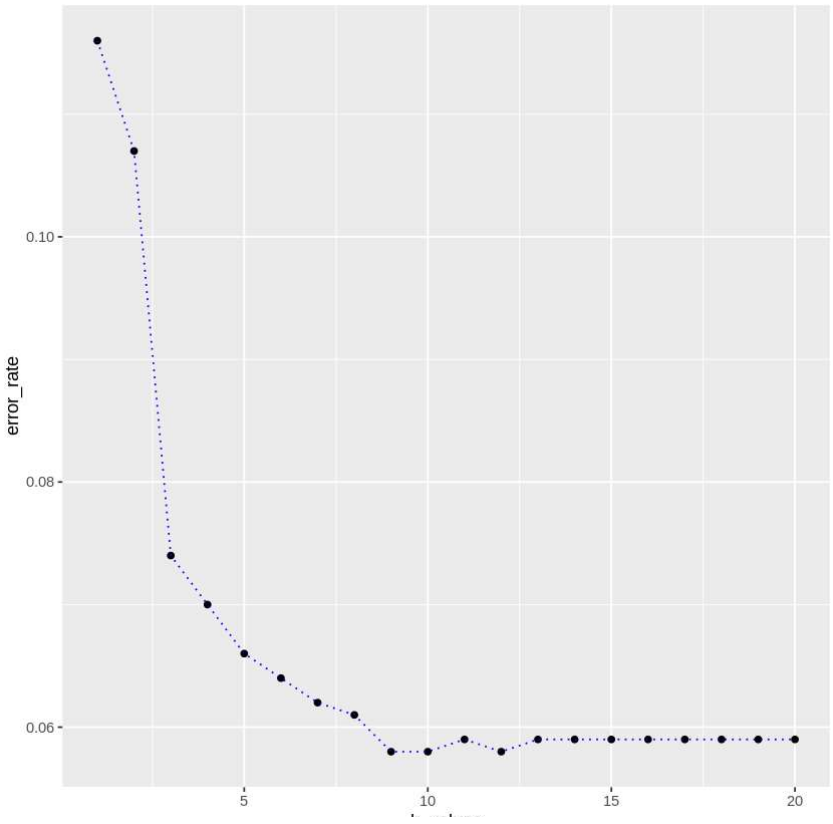
1

No · No · No · No · No · No

▶ **Levels**:

```
[1] 0.058
 [1] 0.116 0.107 0.074 0.070 0.066 0.064 0.062 0.061 0.058 0.058 0.059 0.058
[13] 0.059 0.059 0.059 0.059 0.059 0.059 0.059 0.059
```

	error_rate	k_values
1	0.116	1
2	0.107	2
3	0.074	3
4	0.070	4
5	0.066	5
6	0.064	6
7	0.062	7
8	0.061	8
9	0.058	9
10	0.058	10
11	0.059	11
12	0.058	12
13	0.059	13
14	0.059	14
15	0.059	15
16	0.059	16
17	0.059	17
18	0.059	18
19	0.059	19
20	0.059	20



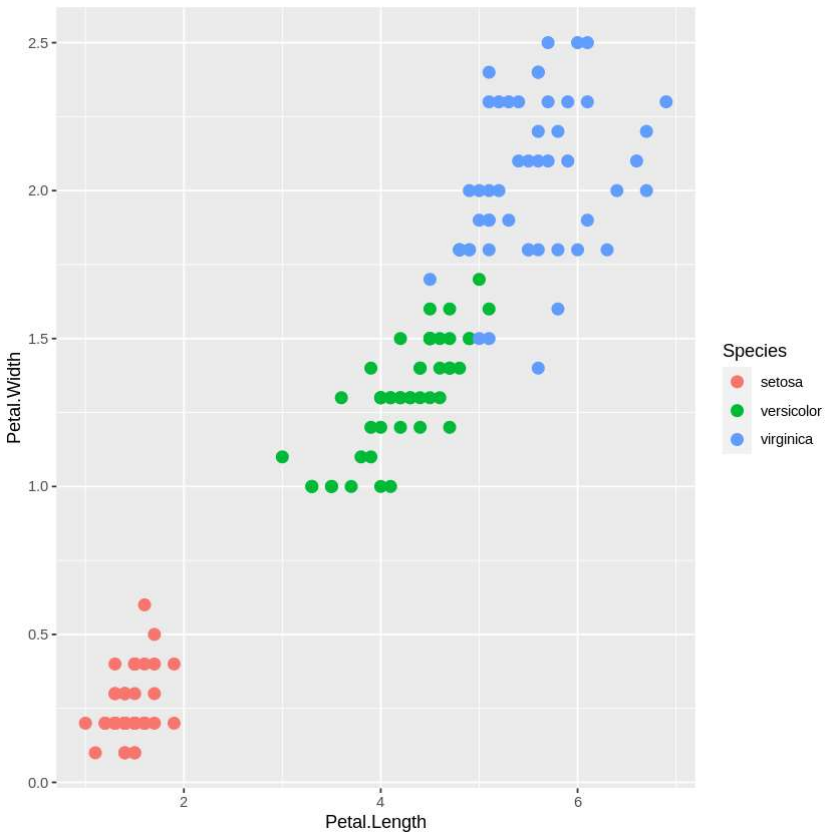
In [119…

```r
# ISLR Package (ISRL 7th Eidtion Textbook )
# KMeans example with Iris dataset
# KMeans is an Unsupervised technique
library(ISLR)
head(iris)
str(iris)
# visually inspecting the three factors: Species clusters with a plot
plot1 <- ggplot(iris, aes(Petal.Length,Petal.Width, color=Species))
print(plot1 + geom_point(size=3))

# set seed for the random numbers
set.seed(101)
# Read the documentation for KMeans in R-Studio
help("kmeans")

irisClusters <- kmeans(iris[,1:4], 3, nstart = 20) # nstart is the number of random start
print(irisClusters)
table(irisClusters$cluster,iris$Species)
# plotting the clusters
library(cluster)
help("clusplot")
clusplot(iris,irisClusters$cluster, color = TRUE, shade = TRUE, labels = 0, lines = 0)
```

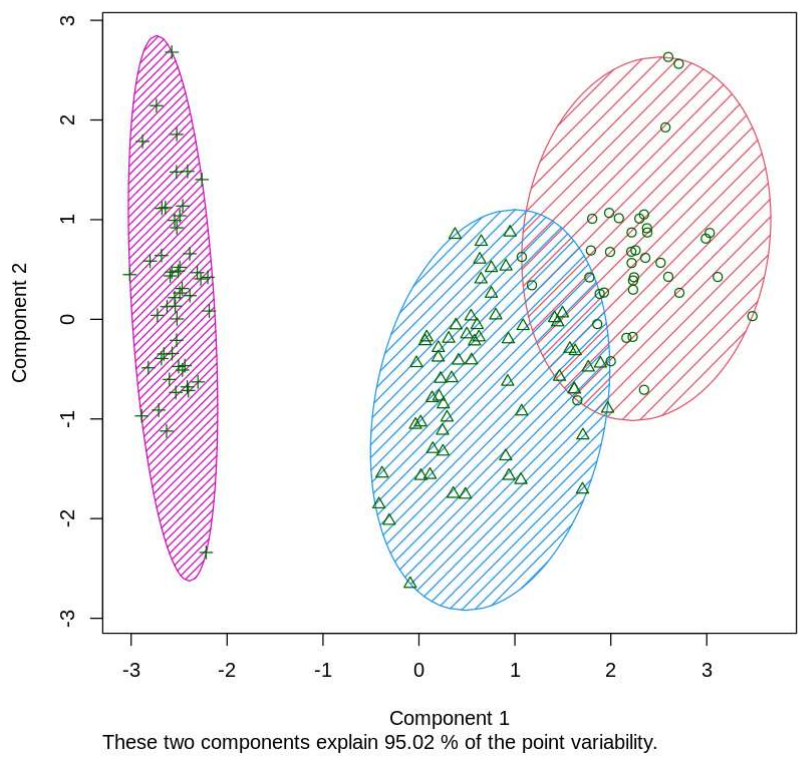
A data.frame: 6 × 5

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
	<dbl>	<dbl>	<dbl>	<dbl>	<fct>
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

```
'data.frame':   150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
[112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1
[149] 1 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"

    setosa versicolor virginica
1       0          2        36
2       0         48        14
3      50          0         0
```



CLUSPLOT(iris)



These two components explain 95.02 % of the point variability.

In [ ]:

In [24]:
```
# Outlier Examples
# Cars dataset is built in Rstudio.
# you need to load the cars dataset first.
cars1<- cars[1:30,] # first 30 rows of the original cars dataset.
head(cars1)
summary(cars1)
```

```
# Now we will introduce some additional data points that are outliers.
cars_outliers <- data.frame ( speed =c(19,19,20,20,20), dist=c (190,186,210,220,218))# introduced the outliears
head(cars_outliers)
cars2 <- rbind(cars1, cars_outliers)
help(par) # Set or Query Graphical Parameters, read the RStudio documentation for "par" function.

plot(cars2$speed, cars2$dist, xlim=c(0, 28), ylim=c(0, 230), main="With Outliers", xlab="speed", ylab="dist",
pch="*", col="red", cex=2)
abline(lm(dist~ speed, data=cars2), col="blue", Iwd=3, Ity=2)
# Plot of original data without outliers. Note the change in slope (angle) of best fit line.
plot(cars1$speed, cars1$ dist, xlim=c(0,28), ylim=c (0,230), main="Outliers removed In A much better fit!",
xlab="speed", ylab="dist", pch="*", col="red", cex=2)
abline(lm(dist~ speed, data=cars1 ), col="blue", Iwd=3, Ity=2)
```

A data.frame: 6 × 2

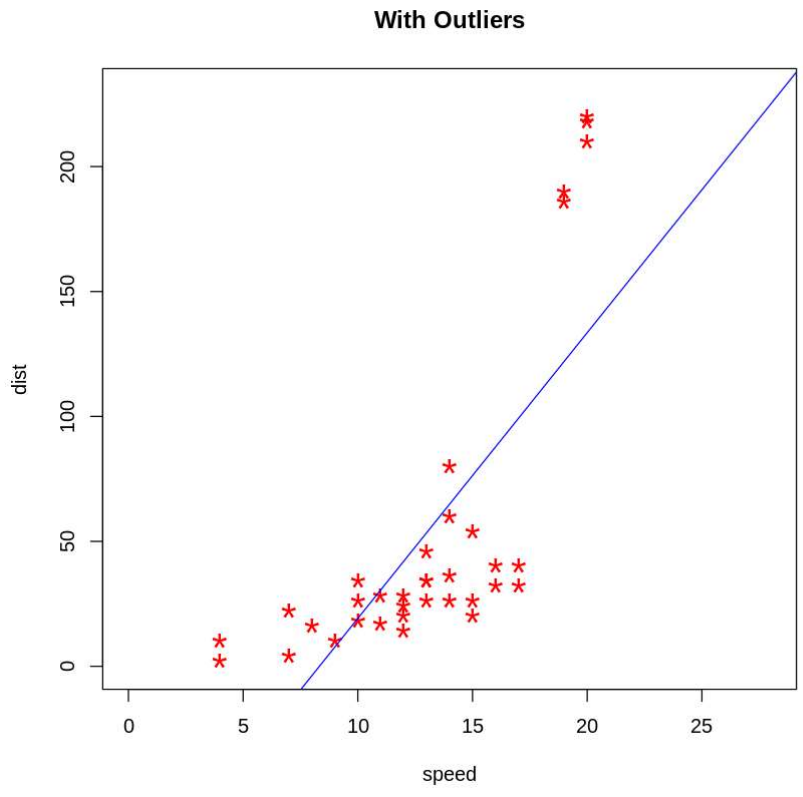
	speed	dist
	<dbl>	<dbl>
1	4	2
2	4	10
3	7	4
4	7	22
5	8	16
6	9	10

```
     speed           dist
 Min.   : 4.00   Min.   : 2.00
 1st Qu.:10.00   1st Qu.:18.50
 Median :12.50   Median :26.00
 Mean   :11.93   Mean   :28.63
 3rd Qu.:14.00   3rd Qu.:34.00
 Max.   :17.00   Max.   :80.00
```

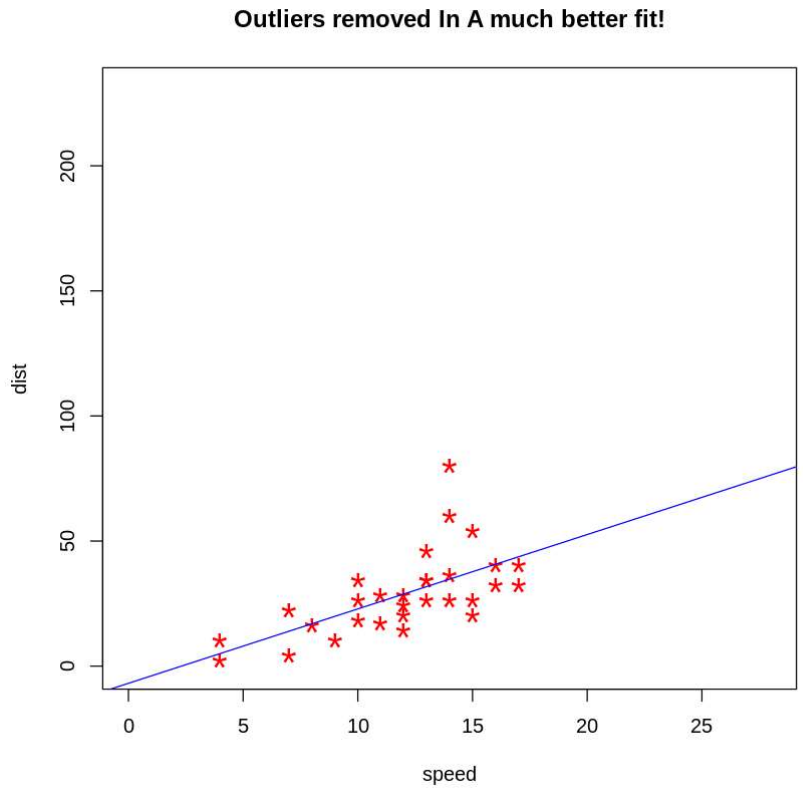
A data.frame: 5 × 2

	speed	dist
	<dbl>	<dbl>
1	19	190
2	19	186
3	20	210
4	20	220
5	20	218

```
Warning message in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
“"Iwd" is not a graphical parameter”
Warning message in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
“"Ity" is not a graphical parameter”
```




```
Warning message in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
“"Iwd" is not a graphical parameter”
Warning message in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
“"Ity" is not a graphical parameter”
```


Outliers removed In A much better fit!
dist
speed
0
50
100
150
200
0
5
10
15
20
25