# Practical: R and data mining

You may want to explore the website freely this week. Please click http://www.rdatamining.com/ link to open the resource. Below are R scripts/packages for data mining which are selected from this link.

**1. Decision Tree**

Building a decision tree and visualise it

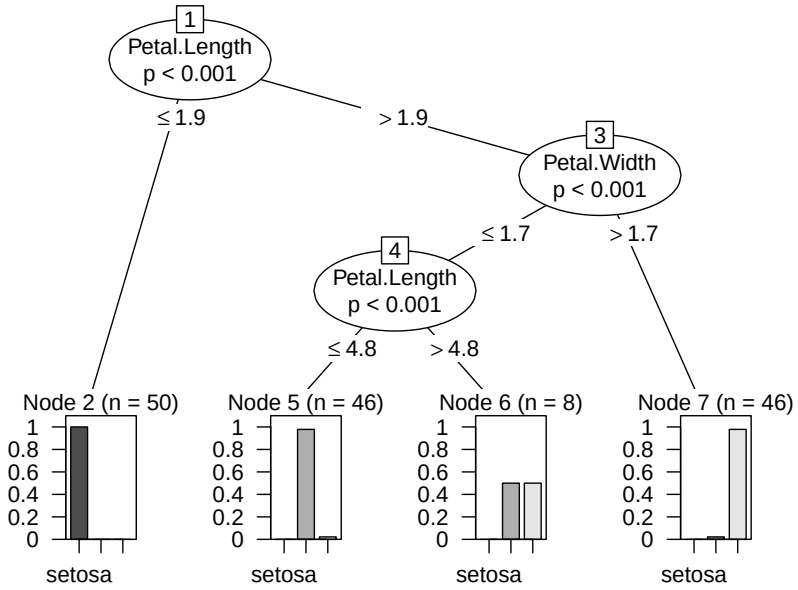
```
library("party")
str(iris)
```

```
## '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 ...
```

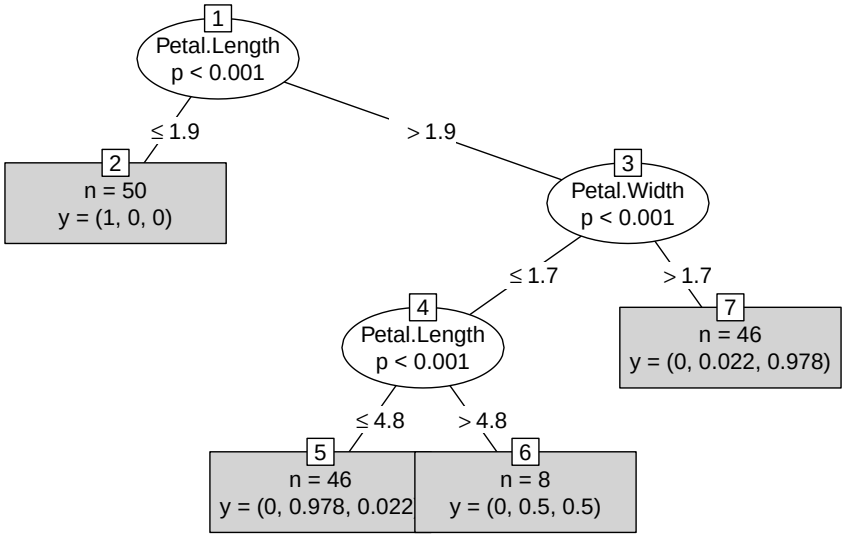
```
# Call function ctree to build a decision tree.
# The first parameter is a formula, which defines a target
# variable and a list of independent variables.

iris_ctree <-
  ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width
                    , data=iris)

print(iris_ctree)
```

```
##
##   Conditional inference tree with 4 terminal nodes
##
## Response:  Species
## Inputs:  Sepal.Length, Sepal.Width, Petal.Length, Petal.Width
## Number of observations:  150
##
## 1) Petal.Length <= 1.9; criterion = 1, statistic = 140.264
##   2)*  weights = 50
## 1) Petal.Length > 1.9
##   3) Petal.Width <= 1.7; criterion = 1, statistic = 67.894
##     4) Petal.Length <= 4.8; criterion = 0.999, statistic = 13.865
##       5)*  weights = 46
##     4) Petal.Length > 4.8
##       6)*  weights = 8
##   3) Petal.Width > 1.7
##     7)*  weights = 46
```

```
plot(iris_ctree)
```




```
plot(iris_ctree, type="simple")
```



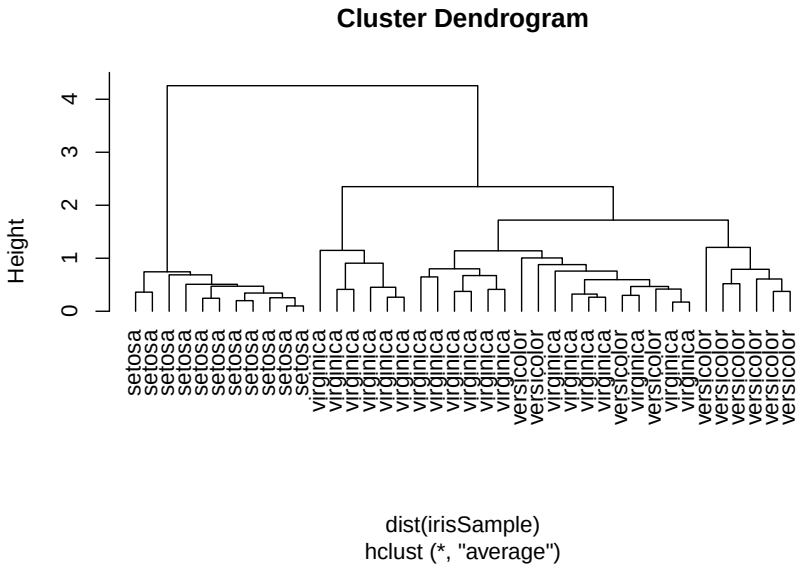

**2. Hierarchical clustering**

Draw a sample of 40 records from iris data, and remove variable Species

```
idx <- sample(1:dim(iris)[1], 40)
irisSample <- iris[idx,]
irisSample$Species <- NULL
```

Perform hierarchical clustering

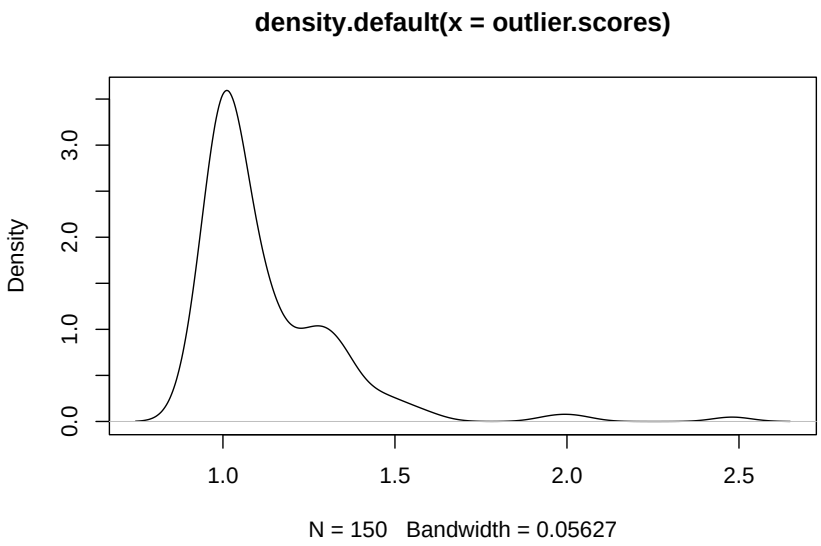
```
hc <- hclust(dist(irisSample), method="ave")
plot(hc, hang = -1, labels=iris$Species[idx])
```




**3. Outlier detection**

The below script uses the LOF (Local Outlier Factor) algorithm to detect outliers. The LOF algorithm identifies local outliers based on density. The detail of the algorithm can be seen in https://www.researchgate.net/publication/221214719_LOF_Identifying_Density-Based_Local_Outliers

```
library(DMwR2)

 # remove "Species", which is a categorical column
 iris2 <- iris[,1:4]
 outlier.scores <- lofactor(iris2, k=5)
 plot(density(outlier.scores))
```




```
# pick top 5 as outliers
outliers <- order(outlier.scores, decreasing=T)[1:5]
# who are outliers
print(outliers)
```

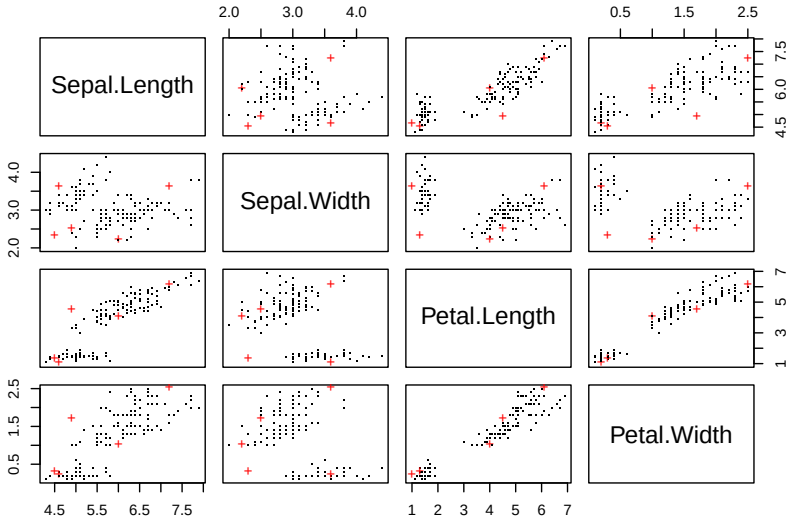
```
## [1]  42 107  23 110  63
```

```
print(iris2[outliers,])
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 42           4.5         2.3          1.3         0.3
## 107          4.9         2.5          4.5         1.7
## 23           4.6         3.6          1.0         0.2
## 110          7.2         3.6          6.1         2.5
## 63           6.0         2.2          4.0         1.0
```

show outliers with a pairs plot as below, where outliers are labeled with “+” in red

```
n <- nrow(iris2)
pch <- rep(".", n)
pch[outliers] <- "+"
col <- rep("black", n)
col[outliers] <- "red"
pairs(iris2, pch=pch, col=col)
```




**4. Associations Rules**

This section includes association rule mining, pruning redundant rules, and visualising association rules.

Association rule mining

```
# Association Rule Mining:
# Following examples use The Titanic dataset, a 4-dimensional table
# with summarized information  on the fate of passengers on the
# Titanic according to social class, sex, age and survival
# It can be found in https://www.rdatamining.com/datasets


 # get current script folder
 myPath <- dirname(rstudioapi::getSourceEditorContext()$path)

 #load dataset (assuming it is in script's folder)
 load(paste0(myPath,"/titanic.raw.rdata"))
 str(Titanic)
```

```
##  'table' num [1:4, 1:2, 1:2, 1:2] 0 0 35 0 0 0 17 0 118 154 ...
##  - attr(*, "dimnames")=List of 4
##   ..$ Class   : chr [1:4] "1st" "2nd" "3rd" "Crew"
##   ..$ Sex     : chr [1:2] "Male" "Female"
##   ..$ Age     : chr [1:2] "Child" "Adult"
##   ..$ Survived: chr [1:2] "No" "Yes"
```

use APRIORI algorithm for association rule mining [Agrawal and Srikant, 1994]. package arules [Hahsler et al., 2014] implements it in ***apriori()*** function

```
library(arules)
 # find association rules with default settings
 rules <- apriori(titanic.raw)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5     0.1      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 220
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[10 item(s), 2201 transaction(s)] done [0.00s].
## sorting and recoding items ... [9 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [27 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
 inspect(rules)
```

```
##      lhs                                   rhs           support   confidence
## [1]  {}                                 => {Age=Adult}   0.9504771 0.9504771
## [2]  {Class=2nd}                        => {Age=Adult}   0.1185825 0.9157895
## [3]  {Class=1st}                        => {Age=Adult}   0.1449341 0.9815385
## [4]  {Sex=Female}                       => {Age=Adult}   0.1930940 0.9042553
## [5]  {Class=3rd}                        => {Age=Adult}   0.2848705 0.8881020
## [6]  {Survived=Yes}                     => {Age=Adult}   0.2971377 0.9198312
## [7]  {Class=Crew}                       => {Sex=Male}    0.3916402 0.9740113
## [8]  {Class=Crew}                       => {Age=Adult}   0.4020900 1.0000000
## [9]  {Survived=No}                      => {Sex=Male}    0.6197183 0.9154362
## [10] {Survived=No}                      => {Age=Adult}   0.6533394 0.9651007
## [11] {Sex=Male}                         => {Age=Adult}   0.7573830 0.9630272
## [12] {Sex=Female, Survived=Yes}         => {Age=Adult}   0.1435711 0.9186047
## [13] {Class=3rd, Sex=Male}              => {Survived=No} 0.1917310 0.8274510
## [14] {Class=3rd, Survived=No}           => {Age=Adult}   0.2162653 0.9015152
## [15] {Class=3rd, Sex=Male}              => {Age=Adult}   0.2099046 0.9058824
## [16] {Sex=Male, Survived=Yes}           => {Age=Adult}   0.1535666 0.9209809
## [17] {Class=Crew, Survived=No}          => {Sex=Male}    0.3044071 0.9955423
## [18] {Class=Crew, Survived=No}          => {Age=Adult}   0.3057701 1.0000000
## [19] {Class=Crew, Sex=Male}             => {Age=Adult}   0.3916402 1.0000000
## [20] {Class=Crew, Age=Adult}            => {Sex=Male}    0.3916402 0.9740113
## [21] {Sex=Male, Survived=No}            => {Age=Adult}   0.6038164 0.9743402
## [22] {Age=Adult, Survived=No}           => {Sex=Male}    0.6038164 0.9242003
## [23] {Class=3rd, Sex=Male, Survived=No} => {Age=Adult}   0.1758292 0.9170616
```

```
## [24] {Class=3rd, Age=Adult, Survived=No}  => {Sex=Male}    0.1758292 0.8130252
## [25] {Class=3rd, Sex=Male, Age=Adult}     => {Survived=No} 0.1758292 0.8376623
## [26] {Class=Crew, Sex=Male, Survived=No}  => {Age=Adult}   0.3044071 1.0000000
## [27] {Class=Crew, Age=Adult, Survived=No} => {Sex=Male}    0.3044071 0.9955423
##      coverage  lift      count
## [1]  1.0000000 1.0000000 2092
## [2]  0.1294866 0.9635051  261
## [3]  0.1476602 1.0326798  319
## [4]  0.2135393 0.9513700  425
## [5]  0.3207633 0.9343750  627
## [6]  0.3230350 0.9677574  654
## [7]  0.4020900 1.2384742  862
## [8]  0.4020900 1.0521033  885
## [9]  0.6769650 1.1639949 1364
## [10] 0.6769650 1.0153856 1438
## [11] 0.7864607 1.0132040 1667
## [12] 0.1562926 0.9664669  316
## [13] 0.2317129 1.2222950  422
## [14] 0.2398910 0.9484870  476
## [15] 0.2317129 0.9530818  462
## [16] 0.1667424 0.9689670  338
## [17] 0.3057701 1.2658514  670
## [18] 0.3057701 1.0521033  673
## [19] 0.3916402 1.0521033  862
## [20] 0.4020900 1.2384742  862
## [21] 0.6197183 1.0251065 1329
## [22] 0.6533394 1.1751385 1329
## [23] 0.1917310 0.9648435  387
## [24] 0.2162653 1.0337773  387
## [25] 0.2099046 1.2373791  387
## [26] 0.3044071 1.0521033  670
## [27] 0.3057701 1.2658514  670
```

```
## use code below if above code does not work
arules::inspect(rules)
```

```
##      lhs                                    rhs           support   confidence
## [1]  {}                                  => {Age=Adult}   0.9504771 0.9504771
## [2]  {Class=2nd}                         => {Age=Adult}   0.1185825 0.9157895
## [3]  {Class=1st}                         => {Age=Adult}   0.1449341 0.9815385
## [4]  {Sex=Female}                        => {Age=Adult}   0.1930940 0.9042553
## [5]  {Class=3rd}                         => {Age=Adult}   0.2848705 0.8881020
## [6]  {Survived=Yes}                      => {Age=Adult}   0.2971377 0.9198312
## [7]  {Class=Crew}                        => {Sex=Male}    0.3916402 0.9740113
## [8]  {Class=Crew}                        => {Age=Adult}   0.4020900 1.0000000
## [9]  {Survived=No}                       => {Sex=Male}    0.6197183 0.9154362
## [10] {Survived=No}                       => {Age=Adult}   0.6533394 0.9651007
## [11] {Sex=Male}                          => {Age=Adult}   0.7573830 0.9630272
## [12] {Sex=Female, Survived=Yes}          => {Age=Adult}   0.1435711 0.9186047
## [13] {Class=3rd, Sex=Male}               => {Survived=No} 0.1917310 0.8274510
## [14] {Class=3rd, Survived=No}            => {Age=Adult}   0.2162653 0.9015152
## [15] {Class=3rd, Sex=Male}               => {Age=Adult}   0.2099046 0.9058824
## [16] {Sex=Male, Survived=Yes}            => {Age=Adult}   0.1535666 0.9209809
## [17] {Class=Crew, Survived=No}           => {Sex=Male}    0.3044071 0.9955423
```

```
## [18] {Class=Crew, Survived=No}            => {Age=Adult}   0.3057701 1.0000000
## [19] {Class=Crew, Sex=Male}               => {Age=Adult}   0.3916402 1.0000000
## [20] {Class=Crew, Age=Adult}              => {Sex=Male}    0.3916402 0.9740113
## [21] {Sex=Male, Survived=No}              => {Age=Adult}   0.6038164 0.9743402
## [22] {Age=Adult, Survived=No}             => {Sex=Male}    0.6038164 0.9242003
## [23] {Class=3rd, Sex=Male, Survived=No}   => {Age=Adult}   0.1758292 0.9170616
## [24] {Class=3rd, Age=Adult, Survived=No}  => {Sex=Male}    0.1758292 0.8130252
## [25] {Class=3rd, Sex=Male, Age=Adult}     => {Survived=No} 0.1758292 0.8376623
## [26] {Class=Crew, Sex=Male, Survived=No}  => {Age=Adult}   0.3044071 1.0000000
## [27] {Class=Crew, Age=Adult, Survived=No} => {Sex=Male}    0.3044071 0.9955423
##      coverage  lift      count
## [1]  1.0000000 1.0000000 2092
## [2]  0.1294866 0.9635051  261
## [3]  0.1476602 1.0326798  319
## [4]  0.2135393 0.9513700  425
## [5]  0.3207633 0.9343750  627
## [6]  0.3230350 0.9677574  654
## [7]  0.4020900 1.2384742  862
## [8]  0.4020900 1.0521033  885
## [9]  0.6769650 1.1639949 1364
## [10] 0.6769650 1.0153856 1438
## [11] 0.7864607 1.0132040 1667
## [12] 0.1562926 0.9664669  316
## [13] 0.2317129 1.2222950  422
## [14] 0.2398910 0.9484870  476
## [15] 0.2317129 0.9530818  462
## [16] 0.1667424 0.9689670  338
## [17] 0.3057701 1.2658514  670
## [18] 0.3057701 1.0521033  673
## [19] 0.3916402 1.0521033  862
## [20] 0.4020900 1.2384742  862
## [21] 0.6197183 1.0251065 1329
## [22] 0.6533394 1.1751385 1329
## [23] 0.1917310 0.9648435  387
## [24] 0.2162653 1.0337773  387
## [25] 0.2099046 1.2373791  387
## [26] 0.3044071 1.0521033  670
## [27] 0.3057701 1.2658514  670
```

```
# rules with rhs (right-hand side) containing "Survived" only
rules <- apriori(titanic.raw, control = list(verbose=F)
                 ,parameter = list(minlen=2, supp=0.005, conf=0.8)
                 ,appearance = list(rhs=c("Survived=No", "Survived=Yes")
                                    ,default="lhs"))

rules.sorted <- sort(rules, by="lift")
inspect(rules.sorted)
```

```
##      lhs                                  rhs            support
## [1]  {Class=2nd, Age=Child}              => {Survived=Yes} 0.010904134
## [2]  {Class=2nd, Sex=Female, Age=Child}  => {Survived=Yes} 0.005906406
## [3]  {Class=1st, Sex=Female}             => {Survived=Yes} 0.064061790
## [4]  {Class=1st, Sex=Female, Age=Adult}  => {Survived=Yes} 0.063607451
## [5]  {Class=2nd, Sex=Female}             => {Survived=Yes} 0.042253521
```

```
## [6]  {Class=Crew, Sex=Female}            => {Survived=Yes} 0.009086779
## [7]  {Class=Crew, Sex=Female, Age=Adult} => {Survived=Yes} 0.009086779
## [8]  {Class=2nd, Sex=Female, Age=Adult}  => {Survived=Yes} 0.036347115
## [9]  {Class=2nd, Sex=Male, Age=Adult}    => {Survived=No}  0.069968196
## [10] {Class=2nd, Sex=Male}               => {Survived=No}  0.069968196
## [11] {Class=3rd, Sex=Male, Age=Adult}    => {Survived=No}  0.175829169
## [12] {Class=3rd, Sex=Male}               => {Survived=No}  0.191731031
##      confidence coverage    lift     count
## [1]  1.0000000  0.010904134 3.095640  24
## [2]  1.0000000  0.005906406 3.095640  13
## [3]  0.9724138  0.065879146 3.010243 141
## [4]  0.9722222  0.065424807 3.009650 140
## [5]  0.8773585  0.048159927 2.715986  93
## [6]  0.8695652  0.010449796 2.691861  20
## [7]  0.8695652  0.010449796 2.691861  20
## [8]  0.8602151  0.042253521 2.662916  80
## [9]  0.9166667  0.076328941 1.354083 154
## [10] 0.8603352  0.081326670 1.270871 154
## [11] 0.8376623  0.209904589 1.237379 387
## [12] 0.8274510  0.231712858 1.222295 422
```

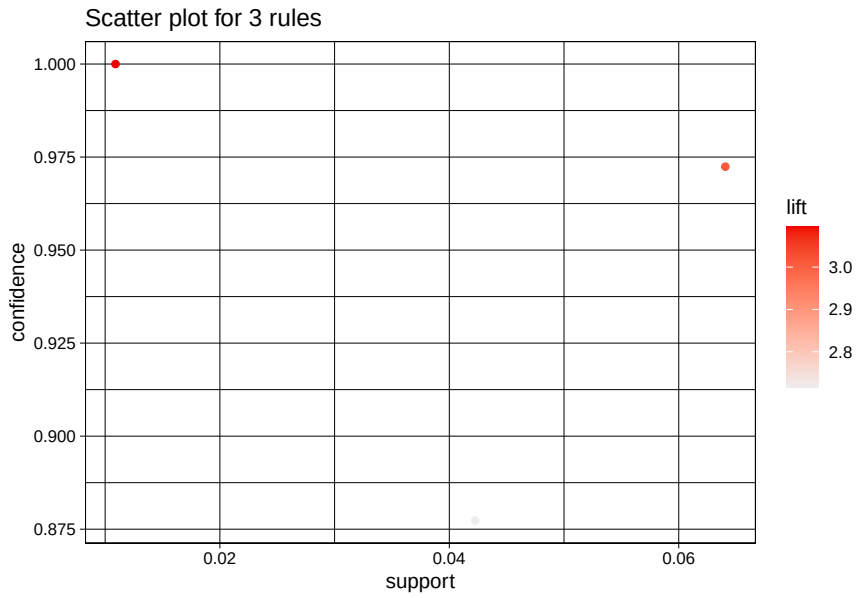
```
# Removing Redundancy, find redundant rules
redundant <- is.redundant(rules.sorted)
which(redundant)
```

```
## [1] 2 4 7 8
```

```
# remove redundant rules
rules.pruned <- rules.sorted[!redundant]
inspect(rules.pruned)
```

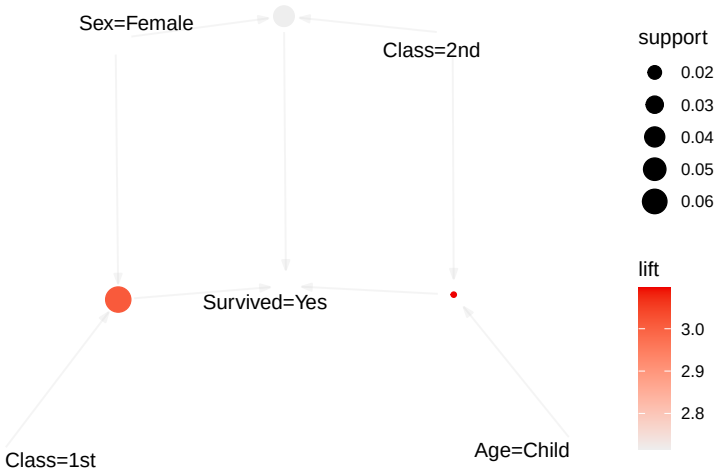
```
##     lhs                                rhs            support     confidence
## [1] {Class=2nd, Age=Child}          => {Survived=Yes} 0.010904134 1.0000000
## [2] {Class=1st, Sex=Female}         => {Survived=Yes} 0.064061790 0.9724138
## [3] {Class=2nd, Sex=Female}         => {Survived=Yes} 0.042253521 0.8773585
## [4] {Class=Crew, Sex=Female}        => {Survived=Yes} 0.009086779 0.8695652
## [5] {Class=2nd, Sex=Male, Age=Adult} => {Survived=No}  0.069968196 0.9166667
## [6] {Class=2nd, Sex=Male}           => {Survived=No}  0.069968196 0.8603352
## [7] {Class=3rd, Sex=Male, Age=Adult} => {Survived=No}  0.175829169 0.8376623
## [8] {Class=3rd, Sex=Male}           => {Survived=No}  0.191731031 0.8274510
##     coverage   lift     count
## [1] 0.01090413 3.095640  24
## [2] 0.06587915 3.010243 141
## [3] 0.04815993 2.715986  93
## [4] 0.01044980 2.691861  20
## [5] 0.07632894 1.354083 154
## [6] 0.08132667 1.270871 154
## [7] 0.20990459 1.237379 387
## [8] 0.23171286 1.222295 422
```

```
# Visualizing Association Rules
library(arulesViz)
plot(rules.pruned[1:3])
```




```
plot(rules.pruned[1:3], method="graph", control=list(type="items"))
```

```
## Available control parameters (with default values):
## layout    =  stress
## circular  =  FALSE
## ggraphdots    =  NULL
## edges     =  <environment>
## nodes     =  <environment>
## nodetext  =  <environment>
## colors    =  c("#EE0000FF", "#EEEEEEFF")
## engine    =  ggplot2
## max   =  100
## verbose   =  FALSE
```



```
plot(rules.pruned[1:3], method="paracoord", control=list(reorder=TRUE))
```