R Commands for Association Rules

```
> install.packages("arules")
> library(arules)
% To change working directory
> setwd("/Users/claudia/Documents/Teaching/SAD/datafiles")
> ex_tr=read.transactions("example.csv",format="basket", sep=",")
> inspect(ex_tr);
> is <- apriori(ex_tr, parameter = list(support= 0.2, target="frequent"))
> inspect(is)
> adult=read.transactions("adultDB.csv", format="basket", sep=",")
## Mine association rules.
> rules <- apriori(adul, parameter = list(supp = 0.5, conf = 0.9, target = "rules"))
> inspect(rules)

## In VOTE attributes are binary, and so almost all values are Y and N, it is required to replace with different values
> vote=read.transactions("vote.csv",format="basket", sep=",")
Warning message:
In asMethod(object) : removing duplicated items in transactions
> library(readr)
> vote <- read_csv("~/Documents/Teaching/SAD/datafiles/vote.csv")
> rules <- apriori(vote, parameter = list(supp = 0.5, conf = 0.9, target = "rules"))
Error in asMethod(object) :
  column(s) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 not logical or a factor. Discretize the columns first.
## Instalar package dplyr
> vote = mutate_if(vote, is.character, as.factor)
> rules <- apriori(vote, parameter = list(supp = 0.1, conf = 1, maxlen=100, target = "rules"))

## In SUPERMARKET there are lots of Missing Values and all variables are binary ➔proceed in the same way
> library(readr)
> supermarket <- read_csv("~/Documents/Teaching/SAD/datafiles/supermarket.csv")
> rules <- apriori(supermarket, parameter = list(supp = 0.9, conf = 1, maxlen=100, target = "rules"))
Error in asMethod(object) :
> supermarket = mutate_if(supermarket, is.character, as.factor)
>
> rules <- apriori(supermarket, parameter = list(supp = 0.9, conf = 1, maxlen=10, target = "rules"))
> write(rules[1:10],"",sep="\n ", quote=TRUE)

## In GLASS all attributes are numeric ➔proceed with discretization CREATE A R SCRIPT
discretize_all = function(table_d, type, n){
  for (i in 1:ncol(table_d)) {
    if (is.numeric(table_d[[i]])) {
      table_d[[i]] = discretize(table_d[[i]], method = type, categories = n,
ordered=TRUE)
    }
  }
  print(summary(table_d))
  return(table_d);
}

glass_d_int = discretize_all(glass, "interval",3)
glass_d_fr = discretize_all(glass, "frequency",3)
```