```r
###########################################################################
##############################################
# Apriori.r
#
#Input the databases, generate association rules based on past data, store
the rules in a dataframe for easy acccess
###########################################################################
#############################################3

students.ece.fall <- read.transactions(file = "Student_fall.csv", format =
"basket", sep = ",")
students.ece.spring <- read.transactions(file = "Student_spring.csv", format
= "basket", sep = ",")
courses <- read.csv(file = "courses_list.csv", header = TRUE)
courses.by.tig.ece <- read.csv(file = "ece_courses_by_tig.csv", header =
TRUE)
#courses.by.tig.cs <- read.csv(file = "cs_courses_by_tig.csv", header = TRUE)
tig.new <- read.csv(file = "tig_list.csv", header = TRUE)
courses.list <- as.list(courses)
tig.list <- as.list(tig.new)
rules.fall <- apriori(students.ece.fall, parameter = list(support = 0.11,
confidence = 0.40))
rules.spring <- apriori(students.ece.spring, parameter = list(support = 0.11,
confidence = 0.40))
rules.sorted <- sort(rules.fall, by = "confidence")
is.redundant(rules.sorted)
rules.pruned <- rules.sorted[!is.redundant(rules.sorted)]
rules2 = data.frame(lhs = labels(lhs(rules.fall)), rhs =
labels(rhs(rules.fall)))
rules2.spring = data.frame(lhs = labels(lhs(rules.spring)), rhs =
labels(rhs(rules.spring)))
```