Five example on how to convert datasets in different format to transactions datatype for arules Rpackage.

https://rdrr.io/cran/arules/man/transactions-class.html

```
## example 1: creating transactions form a list
a_list <- list(
      c("a","b","c"),
      c("a","b"),
      c("a","b","d"),
      c("c","e"),
      c("a","b","d","e")
      )

## set transaction names
names(a_list) <- paste("Tr",c(1:5), sep = "")
a_list

## coerce into transactions
trans1 <- as(a_list, "transactions")

## analyze transactions
summary(trans1)
image(trans1)

## example 2: creating transactions from a matrix
a_matrix <- matrix(c(
  1,1,1,0,0,
         1,1,0,0,0,
         1,1,0,1,0,
         0,0,1,0,1,
         1,1,0,1,1
  ), ncol = 5)

## set dim names
dimnames(a_matrix) <- list(c("a","b","c","d","e"),
         paste("Tr",c(1:5), sep = ""))

a_matrix

## coerce
trans2 <- as(a_matrix, "transactions")
trans2
inspect(trans2)

## example 3: creating transactions from data.frame
a_df <- data.frame(
         age   = as.factor(c(6, 8, NA, 9, 16)),
         grade = as.factor(c("A", "C", "F", NA, "C")),
  pass  = c(TRUE, TRUE, FALSE, TRUE, TRUE))
## note: factors are translated differently to logicals and NAs are ignored
a_df

## coerce
trans3 <- as(a_df, "transactions")
```

```r
inspect(trans3)
as(trans3, "data.frame")

## example 4: creating transactions from a data.frame with
## transaction IDs and items (by converting it into a list of transactions first)
a_df3 <- data.frame(
  TID = c(1,1,2,2,2,3),
  item=c("a","b","a","b","c", "b")
  )
a_df3
trans4 <- as(split(a_df3[,"item"], a_df3[,"TID"]), "transactions")
trans4
inspect(trans4)

## Note: This is very slow for large datasets. It is much faster to
## read transactions using read.transactions() with format = "single".
## This can be done using an anonymous file.
write.table(a_df3, file = tmp <- file(), row.names = FALSE)
trans4 <- read.transactions(tmp, format = "single",
  header = TRUE, cols = c("TID", "item"))
close(tmp)
inspect(trans4)

## example 5: create transactions from a dataset with numeric variables
## using discretization.
data(iris)

irisDisc <- discretizeDF(iris)
head(irisDisc)
trans5 <- as(irisDisc, "transactions")
trans5
inspect(head(trans5))
```