Example Usage for the `discretize()` Function

Jania Vandevoorde

What is the `discretize()` function?

The `discretize()` function introduced by this library is used tro transform continuous variables into discrete categories based on specified quantiles. It operates on a dataset of features `X` with their corresponding labels `y` and applies discretization to selected continuous colums. The intention is to discretize a dataset before running `risk_mod()` which is introduced by the riskscores R package.

The function uses logistic regression to assess the impact of discretization on the model's performance by calculating the Negative Log Likelihood. It also uses `risk_mod()` and `obj_fcn()` provided by the riskscores package when splitting each column into buckets.

Arguments

- `X`: A dataframe containing the features. The columns from `continuous_cols` will be discretized, while all other columns remain the same.
- `y`: The corresponding labels for `X` in the classification task.
- `threshold`: A numeric value (default `0.01`) representing the percentage improvement in the NLL or objective function for evaluating discretization.
- `continuous_cols`: A vector of column names in `X` to be discretized.
- `n_quantiles`: A vector of positive integers specifying the number of quantiles to divide each continuous variable into. The length of this vector must match the number of columns in `continuous_cols`. If `NULL`, the default is 10 quantiles for each column.

Usage

```r
source('discretize.R')

X <- data.frame(
  age = c(25, 30, 35, 40, 45, 50, 55),
  salary = c(30000, 40000, 50000, 60000, 70000, 80000, 90000)
)

y <- c(0, 1, 0, 1, 1, 0, 1)

continuous_cols <- c("age", "salary")

result <- discretize(X, y, threshold = 0.01, continuous_cols = continuous_cols,
                     n_quantiles = c(3, 5))

print(result)
```

```
##   salary_1 salary_2 salary_3 salary_4 salary_5 age_1 age_2 age_3
## 1        1        0        0        0        0     1     0     0
## 2        1        0        0        0        0     1     0     0
## 3        0        1        0        0        0     1     0     0
## 4        0        0        1        0        0     0     1     0
## 5        0        0        0        1        0     0     1     0
## 6        0        0        0        0        1     0     0     1
## 7        0        0        0        0        1     0     0     1
```