About sampling - sample\_random: splits train/test sets and creates folds via random draws, preserving only expected proportions on average.

Environment setup.

# Sampling dataset  
  
# installation   
#install.packages("daltoolbox")  
  
# loading DAL  
library(daltoolbox)

Load dataset and view class distribution.

iris <- datasets::iris  
head(iris)

## Sepal.Length Sepal.Width Petal.Length Petal.Width Species  
## 1 5.1 3.5 1.4 0.2 setosa  
## 2 4.9 3.0 1.4 0.2 setosa  
## 3 4.7 3.2 1.3 0.2 setosa  
## 4 4.6 3.1 1.5 0.2 setosa  
## 5 5.0 3.6 1.4 0.2 setosa  
## 6 5.4 3.9 1.7 0.4 setosa

table(iris$Species)

##   
## setosa versicolor virginica   
## 50 50 50

Split into train/test with random sampling (distributions may vary by draw).

# Split into train and test  
  
# using random sampling  
tt <- train\_test(sample\_random(), iris)  
  
# training distribution  
print(table(tt$train$Species))

##   
## setosa versicolor virginica   
## 41 40 39

# test distribution  
print(table(tt$test$Species))

##   
## setosa versicolor virginica   
## 9 10 11

Create random k-folds and check distribution per fold.

# Split the dataset into folds  
  
# preparing the dataset into four folds  
sample <- sample\_random()  
folds <- k\_fold(sample, iris, 4)  
  
# folds distribution  
tbl <- NULL  
for (f in folds) {  
tbl <- rbind(tbl, table(f$Species))  
}  
print(tbl)

## setosa versicolor virginica  
## [1,] 12 15 10  
## [2,] 14 8 15  
## [3,] 14 12 12  
## [4,] 10 15 13

References - Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. IJCAI.