Imputing missing values – using MICE Package

Tutorial on 5 Powerful R Packages used for imputing missing values

Introduction

Missing values are considered to be the first obstacle in predictive modeling. Hence, it’s important to master the methods to overcome them. Though, some machine learning algorithms claim to treat them intrinsically, but who knows how good it happens inside the ‘black box’.

The choice of method to impute missing values, largely influences the model’s predictive ability. In most statistical analysis methods, listwise deletion is the default method used to impute missing values. But, it not as good since it leads to information loss.

MICE Package

MICE (Multivariate Imputation via Chained Equations) is one of the commonly used package by R users. Creating multiple imputations as compared to a single imputation (such as mean) takes care of uncertainty in missing values.

MICE assumes that the missing data are Missing at Random (MAR), which means that the probability that a value is missing depends only on observed value and can be predicted using them. It imputes data on a variable by variable basis by specifying an imputation model per variable.

**For example:** Suppose we have X1, X2….Xk variables. If X1 has missing values, then it will be regressed on other variables X2 to Xk. The missing values in X1 will be then replaced by predictive values obtained. Similarly, if X2 has missing values, then X1, X3 to Xk variables will be used in prediction model as independent variables. Later, missing values will be replaced with predicted values.

By default, linear regression is used to predict continuous missing values. Logistic regression is used for categorical missing values. Once this cycle is complete, multiple data sets are generated. These data sets differ only in imputed missing values. Generally, it’s considered to be a good practice to build models on these data sets separately and combining their results.

Precisely, the methods used by this package are:

1. PMM (Predictive Mean Matching)  – For numeric variables
2. logreg(Logistic Regression) – For Binary Variables( with 2 levels)
3. polyreg(Bayesian polytomous regression) – For Factor Variables (>= 2 levels)
4. Proportional odds model (ordered, >= 2 levels)

Let’s understand it practically now.

> path <- "../Data/Tutorial"  
> setwd(path)

#load data  
> data <- iris

#Get summary  
> summary(iris)

Since, MICE assumes missing at random values. Let’s seed missing values in our data set using prodNA function. You can access this function by installing missForest package.

#Generate 10% missing values at Random   
> iris.mis <- prodNA(iris, noNA = 0.1)

#Check missing values introduced in the data  
> summary(iris.mis)

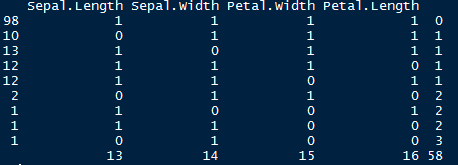
I’ve removed categorical variable. Let’s here focus on continuous values. To treat categorical variable, simply encode the levels and follow the procedure below.

#remove categorical variables  
> iris.mis <- subset(iris.mis, select = -c(Species))  
> summary(iris.mis)

#install MICE  
> install.packages("mice")  
> library(mice)

mice package has a function known as *md.pattern().*  It returns a tabular form of missing value present in each variable in a data set.

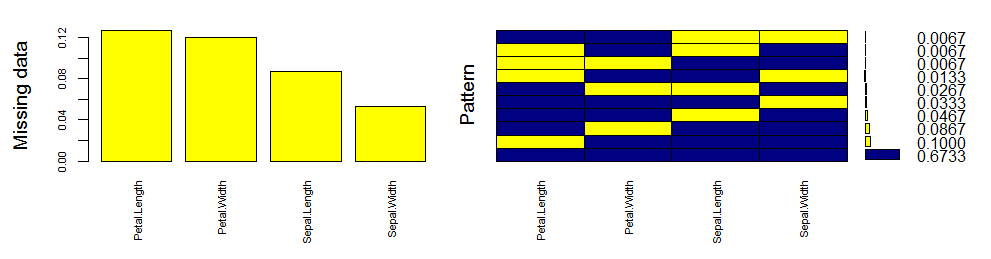
> md.pattern(iris.mis)



Let’s understand this table. There are 98 observations with no missing values. There are 10 observations with missing values in Sepal.Length. Similarly, there are 13 missing values with Sepal.Width and so on.

This looks ugly. Right ? We can also create a visual which represents missing values. It looks pretty cool too. Let’s check it out.

> install.packages("VIM")  
> library(VIM)  
> mice\_plot <- aggr(iris.mis, col=c('navyblue','yellow'),  
                    numbers=TRUE, sortVars=TRUE,  
                    labels=names(iris.mis), cex.axis=.7,  
                    gap=3, ylab=c("Missing data","Pattern"))



Let’s quickly understand this. There are 67% values in the data set with no missing value. There are 10% missing values in Petal.Length, 8% missing values in Petal.Width and so on. You can also look at histogram which clearly depicts the influence of missing values in the variables.

Now, let’s impute the missing values.

> imputed\_Data <- mice(iris.mis, m=5, maxit = 50, method = 'pmm', seed = 500)  
> summary(imputed\_Data)

Multiply imputed data set  
Call:  
mice(data = iris.mis, m = 5, method = "pmm", maxit = 50, seed = 500)  
Number of multiple imputations: 5  
Missing cells per column:  
Sepal.Length Sepal.Width Petal.Length Petal.Width   
13            14          16           15   
Imputation methods:  
Sepal.Length Sepal.Width Petal.Length Petal.Width   
"pmm"        "pmm"        "pmm"       "pmm"   
VisitSequence:  
Sepal.Length Sepal.Width Petal.Length Petal.Width   
1              2            3           4   
PredictorMatrix:  
              Sepal.Length Sepal.Width Petal.Length Petal.Width  
Sepal.Length        0          1            1            1  
Sepal.Width         1          0            1            1  
Petal.Length        1          1            0            1  
Petal.Width         1          1            1            0  
Random generator seed value: 500

Here is an explanation of the parameters used:

1. m  – Refers to 5 imputed data sets
2. maxit – Refers to no. of iterations taken to impute missing values
3. method – Refers to method used in imputation. we used predictive mean matching.

#check imputed values  
> imputed\_Data$imp$Sepal.Width

Since there are 5 imputed data sets, you can select any using *complete()* function.

#get complete data ( 2nd out of 5)  
> completeData <- complete(imputed\_Data,2)

Also, if you wish to build models on all 5 datasets, you can do it in one go using *with()* command. You can also combine the result from these models and obtain a consolidated output using *pool()* command.

#build predictive model  
> fit <- with(data = iris.mis, exp = lm(Sepal.Width ~ Sepal.Length + Petal.Width))

#combine results of all 5 models  
> combine <- pool(fit)  
> summary(combine)