×

-

(http://play.google.com/store/apps/details?id=com.analyticsvidhya.android)

LOGIN / REGISTER (HTTPS://ID.ANALYTICSVIDHYA.COM/ACCOUNTS/LOGIN/?F%20STATS/TUTORIAL%20ON%205%20POWERFUL%20PACKAGES%20USED%20FOR%20IMPUTING%20MISSING%20VALUES%20IN%20R.HTML).ANALYTICSVIDHYA.COM/ACCOUNTS/LOGIN/?NEXT=HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/2016/03/TUTORIAL-POWERFUL-PACKAGES-IMPUTING-MISSING-VALUES/)

Analytics Vidhya
Learn everything about analytics
(https://www.analyticsvidhya.com/blog/)

-

DATA SCIENCE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/DATA-SCIENCE/)
MACHINE LEARNING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/MACHINE-LEARNING/)
R (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/R/)

Tutorial on 5 Powerful R Packages used for imputing missing values

ANALYTICS VIDHYA CONTENT TEAM (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/AUTHOR/AVCONTENTTEAM/), MARCH 4, 2016 LOGI...

Overview

- Learn the methods to impute missing values in R for data cleaning and exploration
- Understand how to use packages like amelia, missForest, hmisc, mi and mice which use bootstrap sampling and predictive modeling

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 (https://www.analyticsvidhya.com/blog/2015/09/random-forest-algorithm-multiple-challenges/)** 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.

Do you know R has robust packages for missing value imputations?

Yes! R Users have something to cheer about. We are endowed with some incredible R packages for missing values imputation. These packages arrive with some inbuilt functions and a simple syntax to impute missing data at once. Some packages are known best working with continuous variables and others for categorical. With this article, you can make a better decision choose the best suited package.

In this article, I've listed 5 R packages popularly known for missing value imputation. There might be more packages. But, I decided to focus on these ones. I've tried to explain the concepts in simplistic manner with practice examples in R.

Tutorial on 5 Powerful R Packages used for imputing missing values

List of R Packages

1. MICE
2. Amelia
3. missForest
4. Hmisc
5. mi

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.

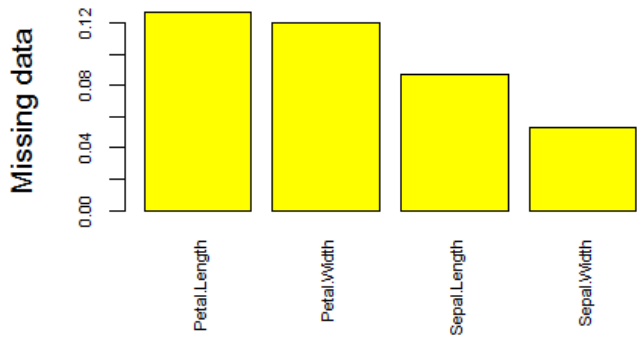
```
> md.pattern(iris.mis)
```

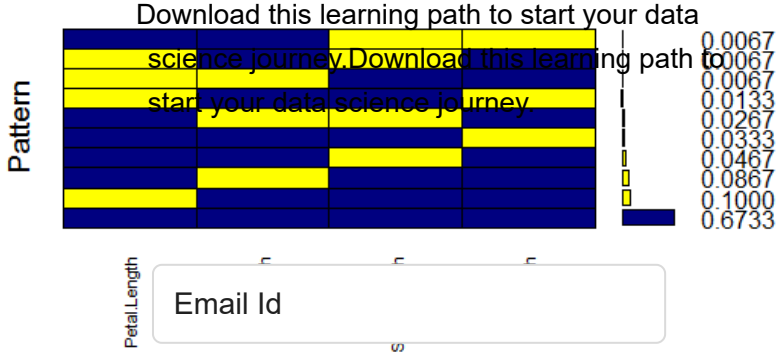
	Sepal.Length	Sepal.Width	Petal.Width	Petal.Length	
98	1	1	1	1	0
10	0	1	1	1	1
13	1	0	1	1	1
12	1	1	1	0	1
12	1	1	0	1	1
2	0	1	1	0	2
1	1	0	0	1	2
1	1	1	0	0	2
1	0	1	0	0	3
	13	14	15	16	58

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)
```

Please note that I've used the command above just for demonstration purpose. You can replace the variable values at your end and try it.

Amelia

This package (Amelia II) is named after Amelia Earhart, the first female aviator to fly solo across the Atlantic Ocean. History says, she got mysteriously disappeared (missing) while flying over the pacific ocean in 1937, hence this package was named to solve missing value problems.

This package also performs multiple imputation (generate imputed data sets) to deal with missing values. Multiple imputation helps to reduce bias and increase efficiency. It is enabled with bootstrap based EMB algorithm which makes it faster and robust to impute many variables including cross sectional, time series data etc. Also, it is enabled with parallel imputation feature using multicore CPUs.

It makes the following assumptions:

1. All variables in a data set have Multivariate Normal Distribution (MVN). It uses means and covariances to summarize data.
2. Missing data is random in nature (Missing at Random)

It works this way. First, it takes m bootstrap samples and applies EMB algorithm to each sample. The m estimates of mean and variances will be different. Finally, the fi e first set of missing values using regression, then second set of estimates are used for second set and so on.

On comparing with MICE, MVN lags on some crucial aspects such as:

1. MICE imputes data on variable by variable basis whereas MVN uses a joint modeling approach based on multivariate normal distribution.
2. MICE is capable of handling different types of variables whereas the variables in MVN need to be normally distributed or transformed to approximate normality.
3. Also, MICE can manage imputation of variables defined on a subset of data whereas MVN cannot.

Hence, this package works best when data has multivariable normal distribution. If not, transformation is to be done to bring data close to normality.

Let's understand it practically now.

```
#install package and load library
> install.packages("Amelia")
> library(Amelia)

#load data
> data("iris")
```

The only thing that you need to be careful about is classifying variables. It has 3 parameters:

1. idvars – keep all ID variables and other variables which you don't want to impute
2. noms – keep nominal variables here

```
#seed 10% missing values
> iris.mis <- prodNA(iris, noNA = 0.1)
> summary(iris.mis)

#specify columns and run amelia
> amelia_fit <- amelia(iris.mis, m=5, parallel = "multicore", noms = "Species")

#access imputed outputs
> amelia_fit$imputations[[1]]
> amelia_fit$imputations[[2]]
> amelia_fit$imputations[[3]]
> amelia_fit$imputations[[4]]
> amelia_fit$imputations[[5]]
```

To check a particular column in a data set, use the following commands

```
> amelia_fit$imputations[[5]]$Sepal.Length

#export the outputs to csv files
> write.amelia(amelia_fit, file.stem = "imputed_data_set")
```

missForest

As the name suggests, missForest is an implementation of random forest (https://www.analyticsvidhya.com/blog/2015/09/random-forest-algorithm-multiple-challenges/) algorithm. It's a non parametric imputation method applicable to various variable types. So, what's a non parametric method ?

Non-parametric method does not make explicit assumptions about functional form of *f* (any arbitary function). Instead, it tries to estimate *f* such that it can be as close to the data points without seeming impractical.

How does it work ? In simple words, it builds a random forest model for each variable. Then it uses the model to predict missing values in the variable with the help of observed values.

It yield OOB (out of bag) imputation error estimate. Moreover, it provides high level of control on imputation process. It has options to return OOB separately (for each variable) instead of aggregating over the whole data matrix. This helps to look more closely as to how accurately the model has imputed values for each variable.

Let's understand it practically. Since bagging works well on categorical variable too, we don't need to remove them here. It very well takes care of missing value pertaining to their variable types:

```
#missForest
> install.packages("missForest")
> library(missForest)

#load data
> data("iris")

#seed 10% missing values
> iris.mis <- prodNA(iris, noNA = 0.1)
> summary(iris.mis)

#impute missing values, using all parameters as default values
> iris.imp <- missForest(iris.mis)
```

```
#check imputed values
> iris.imp$ximp

#check imputation error
> iris.imp$OOBerror

NRMSE         PFC
0.14148554 0.02985075
```

NRMSE is normalized mean squared error. It is used to represent error derived from imputing continuous values. PFC (proportion of falsely classified) is used to represent error derived from imputing categorical values.

```
#comparing actual data accuracy
> iris.err <- mixError(iris.imp$ximp, iris.mis, iris)
>iris.err

NRMSE      PFC
0.1535103 0.0625000
```

This suggests that categorical variables are imputed with 6% error and continuous variables are imputed with 15% error. This can be improved by tuning the values of *mtry* and *ntree* parameter. mtry refers to the number of variables being randomly sampled at each split. ntree refers to number of trees to grow in the forest.

Hmisc

Hmisc is a multiple purpose package useful for data analysis, high – level graphics, imputing missing values, advanced table making, model fitting & diagnostics (linear regression, logistic regression & cox regression) etc. Amidst, the wide range of functions contained in this package, it offers two prominent functions for imputing missing values. These are *impute()* and *aregImpute()*. Though, it also has transcan() function, but aregImpute() is better to use.

impute() function simply imputes missing value using user defined statistical method (mean, max, mean). It's default is median. On the other hand, *aregImpute()* allows mean imputation using additive regression, bootstrapping, and predictive mean matching.

In bootstrapping, different bootstrap resamples are used for each of multiple imputations. Then, a flexible additive model (non parametric regression method) is fitted on a bootstrap sample of original data and missing values (acts as dependent variable) are predicted using non-missing values (independent variable).

Then, it uses predictive mean matching (default) to impute missing values. Predictive mean matching works well for continuous and categorical (binary & multi-level) without the need for computing residuals and maximum likelihood fit.

Here are some important highlights of this package:

1. It assumes linearity in the variables being predicted.
2. Fisher's optimum scoring (https://en.wikipedia.org/wiki/Scoring_algorithm) method is used for predicting categorical variables.

Let's understand it practically.

```
#install package and load library
> install.packages("Hmisc")
> library(Hmisc)

#load data
> data("iris")

#seed missing values ( 10% )
> iris.mis <- prodNA(iris, noNA = 0.1)
> summary(iris.mis)

# impute with mean value
> iris.mis$imputed_age <- with(iris.mis, impute(Sepal.Length, mean))

# impute with random value
> iris.mis$imputed_age2 <- with(iris.mis, impute(Sepal.Length, 'random'))

#similarly you can use min, max, median to impute missing value

#using argImpute
> impute_arg <- aregImpute(~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width +
Species, data = iris.mis, n.impute = 5)
```

argImpute() automatically identifies the variable type and treats them accordingly.

```
> impute_arg
```

```
Multiple Imputation using Bootstrap and PMM

aregImpute(formula = ~Sepal.Length + Sepal.Width + Petal.Length +
    Petal.Width + Species, data = iris.mis, n.impute = 5)

n: 150  p: 5    Imputations: 5          nk: 3

Number of NAs:
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
          21           12           12           14           16

             type d.f.
Sepal.Length    s    2
Sepal.Width     s    2
Petal.Length    s    2
Petal.Width     s    2
Species         c    2

Transformation of Target Variables Forced to be Linear

R-squares for Predicting Non-Missing Values for Each Variable
Using Last Imputations of Predictors
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
       0.865        0.670        0.984        0.958        0.988
```

The output shows R^2 values for predicted missing values. Higher the value, better are the values predicted. You can also check imputed values using the following command

```
#check imputed variable Sepal.Length
> impute_arg$imputed$Sepal.Length
```

mi

mi (Multiple imputation with diagnostics) package provides several features for dealing with missing values. Like other packages, it also builds multiple imputation models to access quality of imputations. And, uses predictive mean matching method.

Though, I've already explained predictive mean matching (pmm) above, but if you haven't understood yet, here's a simpler version: For each observation in a variable with missing value, we find observation (from available values) with the closest predictive mean to that variable. The observed value from this "match" is then used as imputed value.

Below are some unique characteristics of this package:

1. It allows graphical diagnostics of imputation models and convergence of imputation process.
2. It uses bayesian version of regression models to handle issue of separation.

3. Imputation model specification is similar to regression output in R
4. It automatically detects irregularities in data such as high collinearity among variables.
5. Also, it adds noise to imputation process to solve the problem of additive constraints.

Let's understand it practically.

```
#install package and load library
> install.packages("mi")
> library(mi)

#load data
> data("iris")

#seed missing values ( 10% )
> iris.mis <- prodNA(iris, noNA = 0.1)
> summary(iris.mis)

#imputing missing value with mi
> mi_data <- mi(iris.mis, seed = 335)
```

I've used default values of parameters namely:

1. rand.imp.method as "bootstrap"
2. n.imp (number of multiple imputations) as 3
3. n.iter (number of iterations) as 30

```
> summary(mi_data)
```

```
$Sepal.Length
$Sepal.Length$is_missing
missing
FALSE  TRUE
  129    21

$Sepal.Length$imputed
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.0900 -0.6416 -0.4038 -0.2237  0.1847  1.5550

$Sepal.Length$observed
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.93460 -0.43070 -0.05273  0.00000  0.32520  1.14400


$Sepal.Width
$Sepal.Width$is_missing
missing
FALSE  TRUE
  138    12

$Sepal.Width$imputed
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-0.85220 -0.23360  0.08939  0.08501  0.40860  1.30000

$Sepal.Width$observed
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-1.23600 -0.30270 -0.06934  0.00000  0.28070  1.56400
```

Here is a snapshot o summary output by mi package after imputing missing values. As shown, it uses summary statistics to define the imputed values.

End Notes

So, which is the best of these 5 packages ? I am sure many of you would be asking this! Having created this tutorial, I felt Hmisc should be your first choice of missing value imputation followed by missForest and MICE.

Hmisc automatically recognizes the variables types and uses bootstrap sample and predictive mean matching to impute missing values. You don't need to separate or treat categorical variable, just like we did while using MICE package. However, missForest can outperform Hmisc if the observed variables supplied contain sufficient information.

In this article, I explain using 5 different R packages for missing value imputation. Such advanced methods can help you score better accuracy in building predictive models.

Did you find this article useful ? Which package do you genera share your experience / suggestions in the comments section below.

You want to apply your analytical skills and test your potential? Then participate in our Hackathons (http://datahack.analyticsvidhya.com/contest/all) and compete with Top Data Scientists from all over the world.

You can also read this article on Analytics Vidhya's Android APP 

(https://play.google.com/store/apps/details?id=com.analyticsvidhya.android&utm_source=blog_article&utm_campaign=blog&pcampaignid=MKT-Other-global-all-co-prtnr-py-PartBadge-Mar2515-1)

Share this:

(https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/?share=linkedin&nb=1&nb=1)

(https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/?share=facebook&nb=1&nb=1)

(https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/?share=twitter&nb=1&nb=1)

(https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/?share=pocket&nb=1&nb=1)

(https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/?share=reddit&nb=1&nb=1)

Related Articles

(https://www.analyticsvidhya.com/blog/2015/09/build-predictive-model-10-minutes-python/)
Build a Predictive Model in 10 Minutes (using Python) (https://www.analyticsvidhya.com/blog/2015/09/build-predictive-model-10-minutes-python/)
September 23, 2015

(https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/)
A Comprehensive Guide to Data Exploration (https://www.analyticsvidhya.com/blog/2016/01/guide-data-exploration/)
January 10, 2016
In "Business Analytics"

(https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/)
How to perform feature selection (i.e. pick important variables) using Boruta Package in R ? (https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/)

In "Business Analytics"

March 22, 2016
In "R"

TAGS : AMELIA PACKAGE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/AMELIA-PACKAGE/), BOOTSTRAP SAMPLING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/BOOTSTRAP-SAMPLING/), BOOTSTRAPPING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/BOOTSTRAPPING/), DATA CLEANING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/DATA-CLEANING/), DATA EXPLORATION (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/DATA-EXPLORATION/), EDA (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/EDA/), HMISC PACKAGE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/HMISC-PACKAGE/), IMPUTE MISSING VALUES (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/IMPUTE-MISSING-VALUES/), IRIS DATA (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/IRIS-DATA/), MI PACKAGE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/MI-PACKAGE/), MICE PACKAGE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/MICE-PACKAGE/), MISSFOREST PACKAGE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/MISSFOREST-PACKAGE/), MULTIPLE IMPUTATION (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/MULTIPLE-IMPUTATION/), OUT OF BAG ERROR (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/OUT-OF-BAG-ERROR/), PREDICTIVE MEAN MATCHING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/TAG/PREDICTIVE-MEAN-MATCHING/)

NEXT ARTICLE

10 Questions R Users always ask while using ggplot2 package

(https://www.analyticsvidhya.com/blog/2016/03/questions-ggplot2-package-r/)

PREVIOUS ARTICLE

Data Visualizer – Gurgaon (1+ years of experience)

(https://www.analyticsvidhya.com/blog/2016/03/data-visualizer-gurgaon-5-7-years-experience/)

(https://www.analyticsvidhya.com/blog/author/avcontentteam/)

Analytics Vidhya Content Team (Https://Www.Analyticsvidhya.Com/Blog/Author/Avcontentteam/)

Analytics Vidhya Content team

This article is quite old and you might not get a prompt response from the author. We request you to post this comment on Analytics Vidhya's **Discussion portal** (https://discuss.analyticsvidhya.com/) to get your queries resolved

48 COMMENTS

SURYA1987

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106559)

March 4, 2016 at 7:15 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106559)

Hi Manish, thanks for spending your precious time in writing this nice article. I have one doubt whether transformation has to be done after or before imputing missing values. Secondly is there any method to impute outliers.

ANALYTICS VIDHYA CONTENT TEAM

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106563)

March 4, 2016 at 8:26 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106563)

Hi Surya
In case of Amelia, if the data does not have multivariate normal distribution, transformation is required. Alternatively, you can use aregImpute() function from Hmisc pac tching, bootstrapping and addition regression methods.

SURYA1987 Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106569)

March 4, 2016 at 10:36 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106569)

Thank you Manish

SADASHIV Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106564)

March 4, 2016 at 8:40 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106564)

Thanks Manish for an excellent article. . For a feature, how much % of values if missing should be considered for imputation ? What I mean is – if a feature has values in 5-10 % of total rows – it is good to drop the feature. Please correct my understanding if I am wrong.

Thanks again!

SURYA PRAKASH Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106589)

March 4, 2016 at 6:14 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106589)

```
newdata<-
read.csv(file="C:\\Users\\e885735\\Desktop\\Prakash\\train_u6lujuX.csv",head=TRUE,sep=",",stringsAsFactors =
TRUE,na.strings=c("","NA", "-", "?"))
newdata1<-na.omit(newdata)
newdata$Credit_History<-as.factor(newdata$Credit_History)
install.packages("missForest")
library(missForest)
newdata.imp<-missForest(newdata[c(2,3,4,5,6,7,8,9,10,11,12,13)])
```

Now I am comparing actual data accuracy. However I got the below error

```
newdata.err <- mixError(newdata.imp$ximp,newdata,newdata1)
```

:Error in ximp[mis] – xtrue[mis] : non-numeric argument to binary operator
In addition: Warning messages:
1: In as.character(as.matrix(ximp[, t.ind])) != as.character(as.matrix(xtrue[, :
longer object length is not a multiple of shorter object length
2: In as.character(as.matrix(ximp[, t.ind])) != as.character(as.ma
longer object length is not a multiple of shorter object length
3: In as.character(as.matrix(ximp[, t.ind])) != as.character(as.matrix(xtrue[, :

longer object length is not a multiple of shorter object length
4: In as.character(as.matrix(ximp[, t.ind])) != as.character(as.matrix(xtrue[, :
longer object length is not a multiple of shorter object length
5: In as.character(as.matrix(ximp[, t.ind])) != as.character(as.matrix(xtrue[, :
longer object length is not a multiple of shorter object length

---

ANALYTICS VIDHYA CONTENT TEAM

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106738)

March 7, 2016 at 12:59 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106738)

Hi Surya
The error “Longer object length is not a multiple of shorter object length” pops up when one tries to compare two data frames / vectors / arrays of unequal dimensions or sizes. In your case, newdata1 has only 641 observations as compared to newdata which has 981 observations. Since we don’t have complete data, it would be difficult to check the accuracy of imputed values. Alternatively, OOB error is also a good estimate of error accuracy. You can always check OOB error using newdata.imp$OOBerror

---

SURYA PRAKASH

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106876)

March 8, 2016 at 3:50 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106876)

Hi Manish,

Here I am not understanding what should be the arguments in mixError function. In the example which you have provided you have explicitly seeded missing value. However in my case newdata contains missing values. newdata.imp$ximp is the imputed dataset. What should I pass for the second argument in mixError function.

NALIN PASRICHA

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106685)

March 6, 2016 at 5:01 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106685)

great article Manish. I’ve been using some of these packages for a while but I wasn’t aware of many of the nuances you pointed out. Really useful.

ANALYTICS VIDHYA CONTENT TEAM

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106739)

March 7, 2016 at 1:00 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106739)

Thanks Nalin.

MARUTHI Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106783)
March 7, 2016 at 7:09 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106783)

Very good information Manish.Could you please throw light on similar methods along with outlier detection in python also?

LUIZ Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106928)
March 9, 2016 at 2:18 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106928)

Thank you, the tutotial is wonderful, but, I´ve a problem, this command isn´t ok
> combine <- pool(fit)
Error in pool(fit) : The object must have class 'mira'

ANALYTICS VIDHYA CONTENT TEAM Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106980)
March 10, 2016 at 6:56 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106980)

Hi Luiz

Generally, this error doesn't pops up. But you can solve it like this:
>combine <- pool(as.mira(fit))

VISHWA Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-109415)
April 13, 2016 at 6:33 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-109415)

HI, I tried combine<-pool(as.mira(fit)) and got this message: Error in pool(as.mira(fit)) : Object has no coef() method.

GAURAV Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115961)
September 12, 2016 at 7:21 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115961)

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

GEETA CHHABRA Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-110137)
April 28, 2016 at 8:40 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-110137)

Hi Manish
After using combine<-pool (as.mira(fit))
I get the error
Error in pool(as.mira(fit)) : Object has no coef() method

GEETA CHHABRA Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-110173)
April 29, 2016 at 4:02 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-110173)

hi manish

I find this error
Error in pool(as.mira(fit)) : Object has no coef() method.
Please sort this out
Thanks

NEERAJ AGRAWAL Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-114126)
July 27, 2016 at 11:19 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-114126)

Hi Manish,

I got the same error. But instead of iris.mis, I used data = imputed_data. If the input of with() is not mids object, it is invoking base with() function.

Please clarify if I am doing anything wrong.

Thanks,

MANIMARAN Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-114429)

August 4, 2016 at 3:13 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-114429)

Yes i am also getting same error, pls help me out of this.

Best Regards,
Manimaran

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116173)

RAY

September 18, 2016 at 5:19 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116173)

I got the same error, but when I modified my code as below. It works for me.

fit <- with(data = imputed_Data, exp = lm(chol ~ sbp+dbp.+bmi))
summary(pool(fit))

You just need to modify the inputed data and model.

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106945)

AZUL77

March 9, 2016 at 11:29 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106945)

Hi Manish ,

I tried to impute with
df2dosimputados<-aregImpute(~ .,data= df2dosPrestamoslimpio,n.impute=5)

my aim is to impute all my vars, but I obtain this error

Error in terms.formula(formula, specials = "I") :
'.' in formula and no 'data' argument

Do you have any idea to impute all my data frame?

Thanks
.

ANALYTICS VIDHYA CONTENT TEAM Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106979)

March 10, 2016 at 6:51 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-106979)

Hi Azul

Does all the variables of this data set has missing values ? That shouldn't be the case because when a data set has missing values in all columns, the imputed values are highly biased. Hence, I would suggest you to subset the missing columns and then use aregImpute formula. It should work then.

AZUL77 Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108683)

March 30, 2016 at 7:03 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108683)

Hi Manish thanks a lot, You're right,

I separeted my dataframe in two, the firstone with columns with nulls values and the second with not nulls values in the columns.

I applied the method to the columns with NA's, but now I have a new trouble, when I check the results, for example
dataframe$imputed$Ultimosmovimientos[,1], I only can see the imputed values but not all mi columns values.

Maybe that's no a problem with only one column, I think I could merge the values manually, but I have about 50 columns, so my quiestion is, Do you have and advice to "merge" the imputed values with the values that weren't being imputed.

Thanks

PALLAVI Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-107487)

March 16, 2016 at 10:58 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-107487)

Hi Manish,
Excellent ariticle! I have been imputing missing values for various projects. And I always used imputation based on some logic. However when you mentioned that we can measure the error in imputation, It made me think how can we check the error. Principally, the training data itself has m o the data

using appropriate logic to predict what's the best possible value. We would never know if the prediction is correct. But since we are measuring the accuracy of imputation, I am not sure what are we comparing the accuracy against?

ANALYTICS VIDHYA CONTENT TEAM Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-107572)

March 17, 2016 at 5:49 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-107572)

Hello
You are absolutely right. Missing values don't allow us to check their accuracy (predicted). However, missForest provides us out of bag error estimate. Stekhoven and Buhlmann [2011] showed that this estimate produces an appropriate representation of the true imputation error. Least is desirable.

Alternatively, you can use a long method too. Make different models by using multiple techniques (missForest, Hmisc, mean, median)for missing values imputation. I did it one day. I made 4 different models and found Hmisc performed better & faster.

PRIYATAMH (HTTP://WWW.THINKTRAINING.IN/SALESFORCE) Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-107562)

March 17, 2016 at 5:30 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-107562)

You are said another one valuable information,about the reports was really very great.After refer that post i get new more information,thanks for your valuable support to share that post.

SUDHAKAR T Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108311)

March 26, 2016 at 6:23 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108311)

Hi Manish,

I didn't apply all methods before as you describe above. It's new for me. In my case, I am facing a issue related imputation in my data set. I have more than 150 predictor variables and observation near 15000. In data set, half of predictor variables show completed cases (no missing case) where as second half predictor variables show 97% missing cases. Can you recommend which method is good for imputation in this condition?

RAHUL Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108315)

March 26, 2016 at 7:08 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108315)

Thanks Manish for nice artical

can you please help me with getting “iris” data set used in above example….

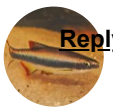
DEEPAK

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108342)

March 26, 2016 at 7:14 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108342)

Hi Manish,

I am using kNN method with K value 6 for NA values imputation. Is this method powerful for imputing missing data in both categorical and continuous variables.



DOUG DAME

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108363)

March 27, 2016 at 12:34 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-108363)

Very interesting article, much thanks.

In this case, since you created the missing values in the IRIS dataset yourself, “ground truth” is available. And thus you could show exactly how accurate each of the various methods’ imputations were.

Doesn’t mean those same results would necessarily extrapolate to other datasets, especially ones with more complicated data, but it’d be fun to see !

RAHUL

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-109888)

April 23, 2016 at 5:53 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-109888)

I am using mice() function in R but it keeps running out of memory.
I use 64 bit R, windows 7 and 8 Gb ram.

imp1 <-mice(train_data1, m=5)
Error: cannot allocate vector of size 34.8 Gb
In addition: Warning messages:

1: In rep.int(c(1, numeric(n)), n – 1L) :
Reached total allocation of 8072Mb: see help(memory.size)
2: In rep.int(c(1, numeric(n)), n – 1L) :
Reached total allocation of 8072Mb: see help(memory.size)
3: In rep.int(c(1, numeric(n)), n – 1L) :
Reached total allocation of 8072Mb: see help(memory.size)
4: In rep.int(c(1, numeric(n)), n – 1L) :
Reached total allocation of 8072Mb: see help(memory.size)

The data has about 70K obs. of 12 variables. What should I do?

Thanks

DAVE YANG (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117448)
October 24, 2016 at 2:30 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117448)

R stores everything is RAM and your file size seems to exceed its max capacity.

VAMSHI KRISHNA (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-110359)
May 3, 2016 at 11:38 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-110359)

HI all,
I'm Working on a retail project , I need missing value imputation code in R.
The Dataset is like.
Manufacture > Sub Category > Brand > Sub Brand> Units..
So Here I need to impute the missing values by Manufacture > Sub Category > Brand > Sub Brand wise.

Please Help me.

LINEARFOX (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-111159)
May 19, 2016 at 4:30 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-111159)

Nice article.
I mostly use the "irmi" and "kNN" imputation methods from VIM or if it's time series data the imputeTS package.

AVINASH Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-111690)

June 1, 2016 at 12:33 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-111690)

Hello manish
Like in using missForest model using data set of Big Mart Sale, I separated the numerical variables and applied missForest after which when I am trying to use cbind to join the numerical and factor variables to form the original data set it is showing
“Error in as.data.frame.default(x[[i]], optional = TRUE, stringsAsFactors = stringsAsFactors) : cannot coerce class ““missForest”” to a data.frame”
I even tried as.data.frame() to change class but it didn’t worked out

MUDIT Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-112149)

June 12, 2016 at 1:04 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-112149)

Hi,

After running the code using MICE package for imputation this is the error i get

completeData <- complete(imputed_Data1,2)
Error in (function (classes, fdef, mtable) :
unable to find an inherited method for function ‘complete’ for signature ‘"mids"’

Any idea or help

TRẦN TRỌNG KHẢI Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-113823)

July 22, 2016 at 10:12 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-113823)

Hi Manish Saraswat,
I have a data set at this link http://www.mediafire.com/download/i2nc2di5p4nfbsl/hmisc2.csv (http://www.mediafire.com/download/i2nc2di5p4nfbsl/hmisc2.csv)
It has all of data types.
I use Hmisc package to handle missing values.
My code is:

iris.mis=read.csv2("G:\\Thanh Phuong xlsl\\hmisc2.csv", sep=";", na.strings = "na", header=TRUE)
library(Hmisc)
impute_arg <- aregImpute(~ weight + oral + gcs + oi + ivdu + csw + previousTB + pulmonaryTB + TBMgrade + disability.base+ disability.2mo+ cd4count+ cd4.2mo+ hivrna.base+ hivrna.2mo, data = iris.mis, n.impute = 5)

and i have a notice:

Iteration 1
fewer than 3 unique knots. Frequency table of variable:
x
1 2 3
61 54 15
Error in rcspline.eval(z, knots = parms, nk = nk, inclx = TRUE) :
In addition: Warning messages:
1: In rcspline.eval(z, knots = parms, nk = nk, inclx = TRUE) :
could not obtain 3 interior knots with default algorithm.
Used alternate algorithm to obtain 3 knots
2: In rcspline.eval(z, knots = parms, nk = nk, inclx = TRUE) :
3 knots requested with 3 unique values of x. knots set to 1 interior values.

How can i handle this problem?
Thanks for your consideration.

G.S.RAGHURAM Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115168)
August 26, 2016 at 3:19 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115168)

Very Valuable Information thanks for sharing.

ROBERT Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115338)
August 30, 2016 at 8:38 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115338)

Hi Manish,

As always fantastic article. Your work is always top notch.

Probably a silly question but after I run my aregImpute model ho[illegible] my imputed values out of it? Do I just take the vector out and stitch it together in a new dataframe?

thanks.

SCOTT BRADSHAW Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115402)

September 1, 2016 at 9:17 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-115402)

Hi Manish,

With the above methods, how do you impute for data sets that you want to predict on? For example, if I take a simple imputation method like mean imputation (just using mean of non-missing values), I would put the mean value in my training data set and train my model. When I want to use my model to predict, I'd get the predict data set, replace the missing values with the mean value (that I derived from the training set) and run my model. So I'm doing the same imputation for train and predict data sets.

With the above methods (I've only tried missForest), I can't see how you apply the exact same imputation to train and predict data sets. Running imputation on just the predict data set wouldn't apply the same imputation as it did on the train data set (you could just have one row to predict).

Any thoughts?

SAJJAN Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116522)

September 27, 2016 at 6:56 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116522)

Nice article!

JITESH ARORA Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116536)

September 27, 2016 at 3:45 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116536)

Hi, Manish. A very well put article.
I have a doubt related to missing data values. Please throw some light, if you may.
What is the best way to deal with an attribute missing 30%(say) data?
I think these packages are useful only to some extent. Also, with a lot of missing data the time of execution of these imputations also reach to a very high magnitude. Is there a way to cater all of these problems?
Thanks in Advance!!

LOKESH Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116551)

September 28, 2016 at 12:16 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-116551)

mi package is taking a long time. I am using 64 bit machine with 8GB RAM. Is there any other way to use “mi”

MOHAMMADI Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117112)

October 12, 2016 at 5:35 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117112)

Hi.
What approach is best imputation of missing values ,for highly correlated data such as gene , Microarrys ? explain by example Please.
Thanks.

DAVE YANG Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117447)

October 24, 2016 at 2:26 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117447)

Your through tutorial helps a lot. I have a quick question:

In MICE, I try to pool fit model and I encountered the following error message.

“Error in pool(fit) : The object must have class ‘mira’

When I define fit, I used ‘ data=completeData and everything else is the same.

This issue seems to exist as discussed in the link, https://stat.ethz.ch/pipermail/r-help/2007-May/132180.html (https://stat.ethz.ch/pipermail/r-help/2007-May/132180.html).

Please advise

MOYO Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117611)

October 28, 2016 at 7:00 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-117611)

Thanks Manish for the article, it is really helpful.

I applied the hmisc function but I expected a new dataset with the missing values imputed (the easy one :)) Is there any way I can do this with R effortlessly or am I missing something.

Thanks.

AHMED

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-118114)

November 8, 2016 at 11:14 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-118114)

Greetings Manish, Thanks for the helpful post. Is there an imputation method in R where I could use a Wiener Process?

TULIO BAGNOLI

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-118220)

November 10, 2016 at 2:15 pm (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-118220)

Thanks a lot for putting this together!

I've been using missForest for a while now, and I'm very happy with it. I can build solid predictions that would be simply impossible if I had to throw out each row with a missing value (I'm actually still baffled by the increase in general accuracy that can come from very sparsely populated variables).

I bumped into a limitation of missForest, though: it doesn't seem possible to *save* the filling algorithm it produces and simply apply it on a different set (of course having identical columns). It's a bummer for me, because it means that whenever I get new data (of the same sort), I need to train a new missForest model on it, instead of just applying the old one.

So my question: does any of the other models allow to save the filling criteria trained on a dataset, and apply *the same* to a new one, without learning how to fill from the new data?

UJJAYANT SINHA

Reply (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-128831)

May 21, 2017 at 9:22 am (https://www.analyticsvidhya.com/blog/2016/03/tutorial-powerful-packages-imputing-missing-values/#comment-128831)

How would can you use mice to apply the same method for imputing missing data in the test set as you used in your training set. ?

POPULAR POSTS

24 Ultimate Data Science Projects To Boost Your Knowledge and Skills (& can be accessed freely) (https://www.analyticsvidhya.com/blog/2018/05/24-ultimate-data-science-projects-to-boost-your-knowledge-and-skills/)

Commonly used Machine Learning Algorithms (with Python and R Codes) (https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/)

A Complete Python Tutorial to Learn Data Science from Scratch (https://www.analyticsvidhya.com/blog/2016/01/complete-tutorial-learn-data-science-python-scratch-2/)

7 Regression Techniques you should know! (https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/)

Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes) (https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/)

Complete Guide to Parameter Tuning in XGBoost with codes in Python (https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/)

Understanding Support Vector Machine algorithm from examples (along with code) (https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/)

A comprehensive beginner's guide to create a Time Series F (https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/)

Download Resource

RECENT POSTS

A Data Science Leader's Guide to Managing Stakeholders (https://www.analyticsvidhya.com/blog/2019/08/data-science-leader-guide-managing-stakeholders/)

AUGUST 1, 2019

Building a Recommendation System using Word2vec: A Unique Tutorial with Case Study in Python (https://www.analyticsvidhya.com/blog/2019/07/how-to-build-recommendation-system-word2vec-python/)

JULY 30, 2019

OpenAI's GPT-2: A Simple Guide to Build the World's Most Advanced Text Generator in Python (https://www.analyticsvidhya.com/blog/2019/07/openai-gpt2-text-generator-python/)

JULY 29, 2019

Introduction to Bayesian Adjustment Rating: The Incredible Concept Behind Online Ratings! (https://www.analyticsvidhya.com/blog/2019/07/introduction-online-rating-systems-bayesian-adjusted-rating/)

JULY 26, 2019

(https://courses.analyticsvidhya.com/courses/natural-language-processing-nlp?utm_source=Sticky_banner1&utm_medium=display&utm_campaign=NLPcourse)

(https://datamin.analyticsvidhya.com/?utm_source=Sticky_banner2&utm_medium=display&utm_campaign=datamin)

ANALYTICS VIDHYA

About Us (http://www.analyticsvidhya.com/about-me/)

Our Team (https://www.analyticsvidhya.com/about-me/team/)

Career (https://www.analyticsvidhya.com/career-analytics-vidhya/)

Contact Us (https://www.analyticsvidhya.com/contact/)

Write for us (https://www.analyticsvidhya.com/about-me/write/)

DATA SCIENTISTS

Blog (https://www.analyticsvidhya.com/blog/)

Hackathon (https://datahack.analyticsvidhya.com/)

Discussions (https://discuss.analyticsvidhya.com/)

Apply Jobs (https://www.analyticsvidhya.com/jobs/)

Leaderboard (https://datahack.analyticsvidhya.com/contest/all/)

COMPANIES

Post Jobs (https://www.analyticsvidhya.com/corporate/)

Training (https://trainings.analyticsvidhya.com/)

Hiring Hackathons (https://datahack.analyticsvidhya.com/)

Advertising

Reach Us (https://www.analyticsvidhya.com/contact/)

JOIN OUR COMMUNITY :

(https://www.facebook.com/analyticsvidhya) 21587 Followers

(https://twitter.com/analyticsvidhya) 48936 Followers

(https://www.linkedin.com/company/analytics-vidhya) Followers

(https://plus.google.com/+Analyticsvidhya) Followers

(https://in.linkedin.com/company/analytics-vidhya) Followers

Your Ultimate path for Becoming a DATA Scientist!

Download this learning path to start your data science journey.

Email Id >

Download Resource