Blog 5: Data Imputation Methods

Eric Lehmphul

5/22/2022

```
library(tidyverse)
```

Introduction

When working with real world data, it is inevitable that you will encounter missing values. This blog will discuss and demonstrate how to implement different data imputation techniques in R.

Data Imputation

There are two main ways to handle missing data: remove rows that contain NA values and replace NA values with a guestimate. Removing rows is generally not recommended as it removes the other variables as well and can lead to a loss of predictive information. Imputing the data is typically a better approach to handling missing values. There are many different approaches to handling missng values. Some are:

- For Numeric Data:
  - Replace with arbitrary value
  - Replace with mean
  - Replace with median
  - Forward fill
  - Backward fill
  - Machine Learning model
- For Categorical Variables:
  - Replace with most frequent value
  - Create new category for missing values

Create Missing Values

```
data("iris")

iris$Sepal.Length[sample(1:length(iris$Sepal.Length), 25)] <- NA
iris$Sepal.Width[sample(1:length(iris$Sepal.Width), 10)] <- NA
iris$Petal.Length[sample(1:length(iris$Petal.Length), 10)] <- NA
iris$Petal.Width[sample(1:length(iris$Petal.Width), 10)] <- NA
iris$Species[sample(1:length(iris$Species), 10)] <- NA
```

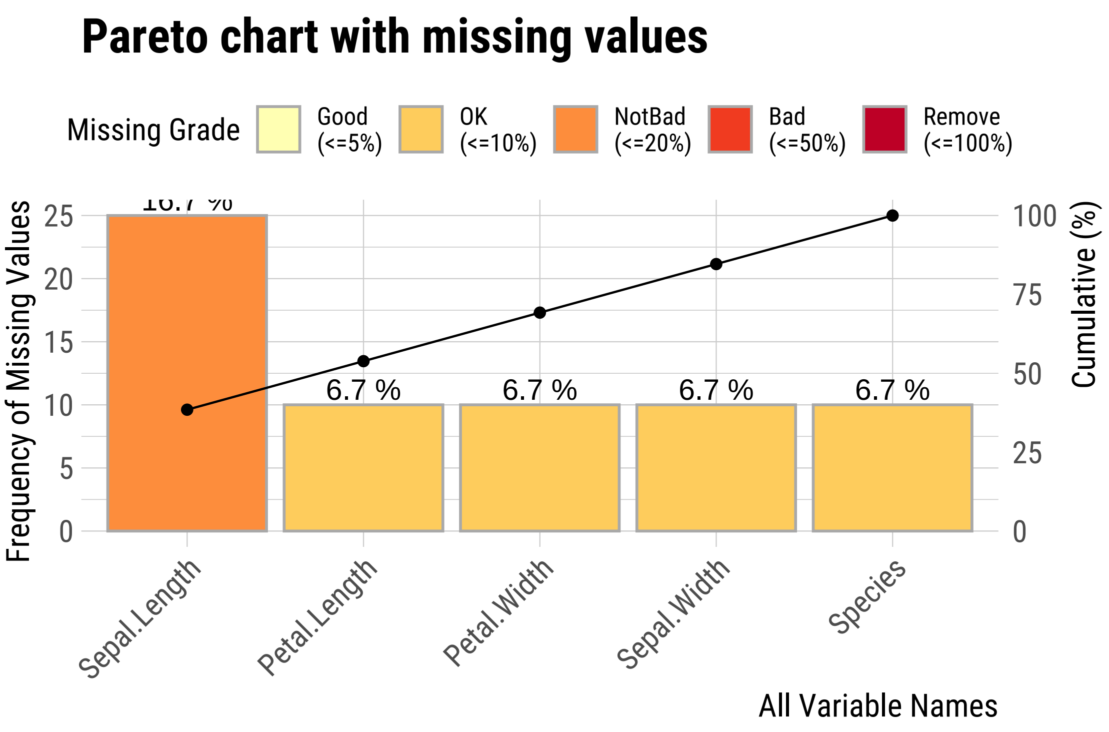
```
paste0("Number of Missing Values in Dataset: ", sum(is.na(iris)))
```

```
## [1] "Number of Missing Values in Dataset: 65"
```

Visualize Missing Values

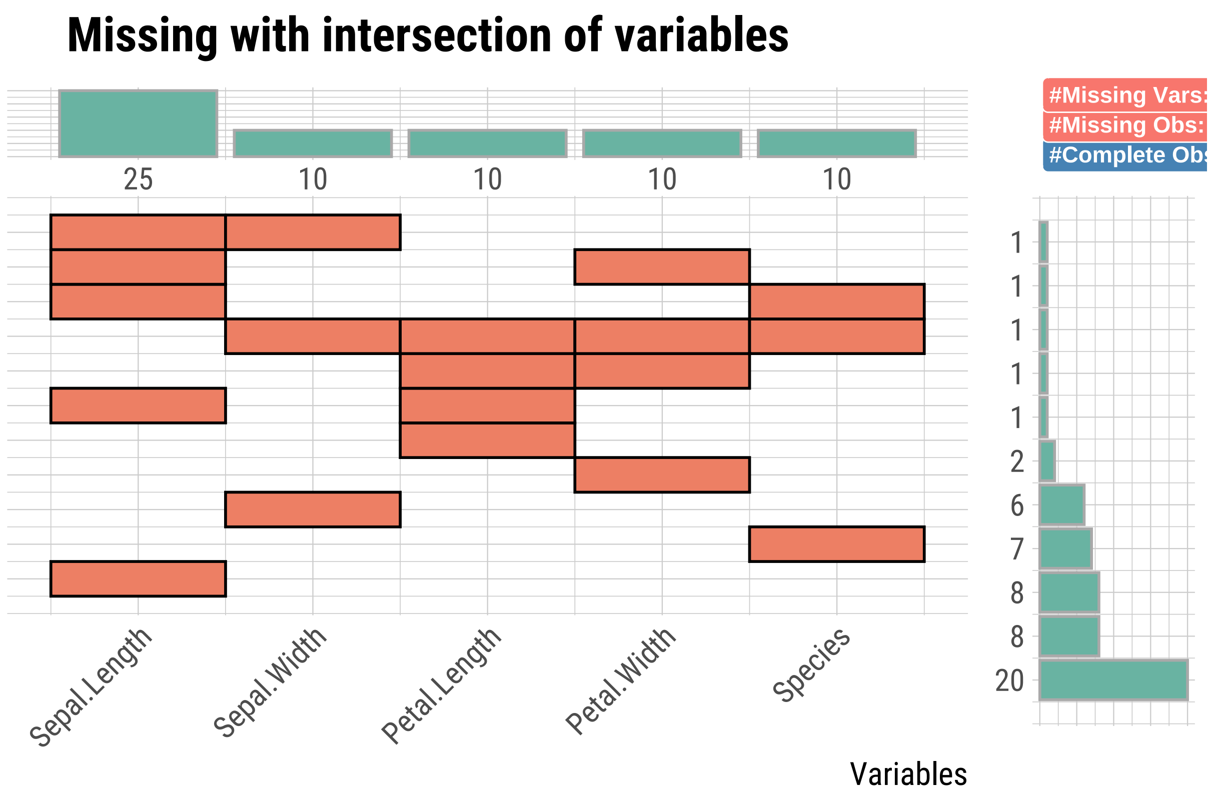
The dlookr package provides many handy functions in dealing with missing values and visualizing missing values. The Pareto chart shows the percentage of missing data per variable.

```
library(dlookr)

plot_na_pareto(iris)
```



The na_intersect function finds missing values that are in the same row as another missing value.

```
plot_na_intersect(iris)
```



Imputation Techniques

Replace with Arbitrary Value, Mean, Median

The Hmisc package has an impute function to easily carry out the simplest form of data imputation using the mean median, and an arbitrary value.

```r
library(Hmisc)

impute1 <- iris
impute1$Sepal.Length <- impute(impute1$Sepal.Length, 5)
impute1$Sepal.Width <- impute(impute1$Sepal.Width, mean)
impute1$Petal.Length <- impute(impute1$Petal.Length, median)

paste0("Number of Missing Values in Dataset: ", sum(is.na(impute1)))
```

```
## [1] "Number of Missing Values in Dataset: 20"
```

Replace using Forward and Backward fill

Forward fill looks at the next value and fills the NA with that value. Backward fill looks at the previous value and fills the NA with that value.

```
library(tidyr)


impute2 <- iris

paste0("Number of Missing Values in Before: ", sum(is.na(impute2)))
```

```
## [1] "Number of Missing Values in Before: 65"
```

```
impute2 <- impute2 %>% fill(Sepal.Length, .direction = "down")
impute2 <- impute2 %>% fill(Sepal.Width, .direction = "up")
paste0("Number of Missing Values in After: ", sum(is.na(impute2)))
```

```
## [1] "Number of Missing Values in After: 30"
```

Replace using Machine Learning

RPART - Recursive Partitioning and Regression Trees

This imputation algorithm leverages machine learning by using decision trees to predict the missing value.

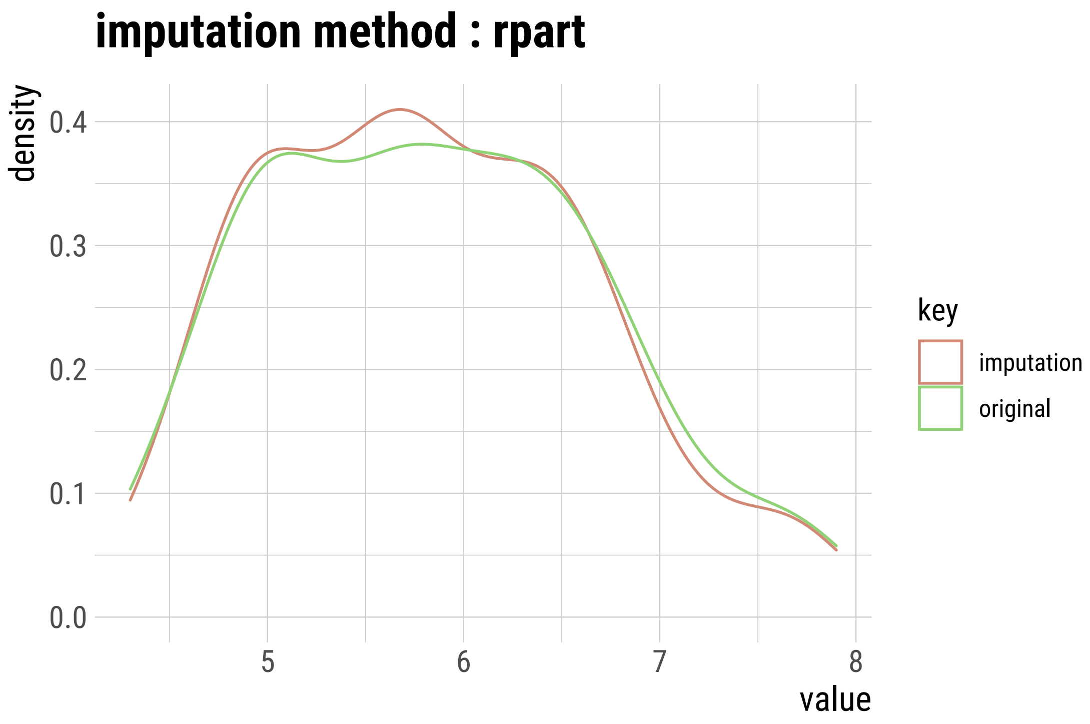
```
impute3 <- iris

paste0("Number of Missing Values in Before: ", sum(is.na(impute3$Sepal.Length)))
```

```
## [1] "Number of Missing Values in Before: 25"
```

```
impute3$Sepal.Length <- imputate_na(iris, xvar = Sepal.Length, yvar = "Species", method = "rpart", no_a
paste0("Number of Missing Values in After: ", sum(is.na(impute3$Sepal.Length)))
```

```
## [1] "Number of Missing Values in After: 0"
```

```r
imputate_na(iris, xvar = Sepal.Length, yvar = "Species", method = "rpart") %>% plot()
```



MICE - Multivariate Imputation by Chained Equations

The MICE algorith utilizes a divide and conquer approach to impute missing values while leveraging the predictive power of a regression model.

```r
impute3 <- iris

paste0("Number of Missing Values in Before: ", sum(is.na(impute3$Sepal.Width)))
```

```
## [1] "Number of Missing Values in Before: 10"
```

```r
impute3$Sepal.Width <- imputate_na(iris, xvar = Sepal.Width, yvar = "Species", method = "mice", no_attr
```

```
## 
##  iter imp variable
##   1   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
```

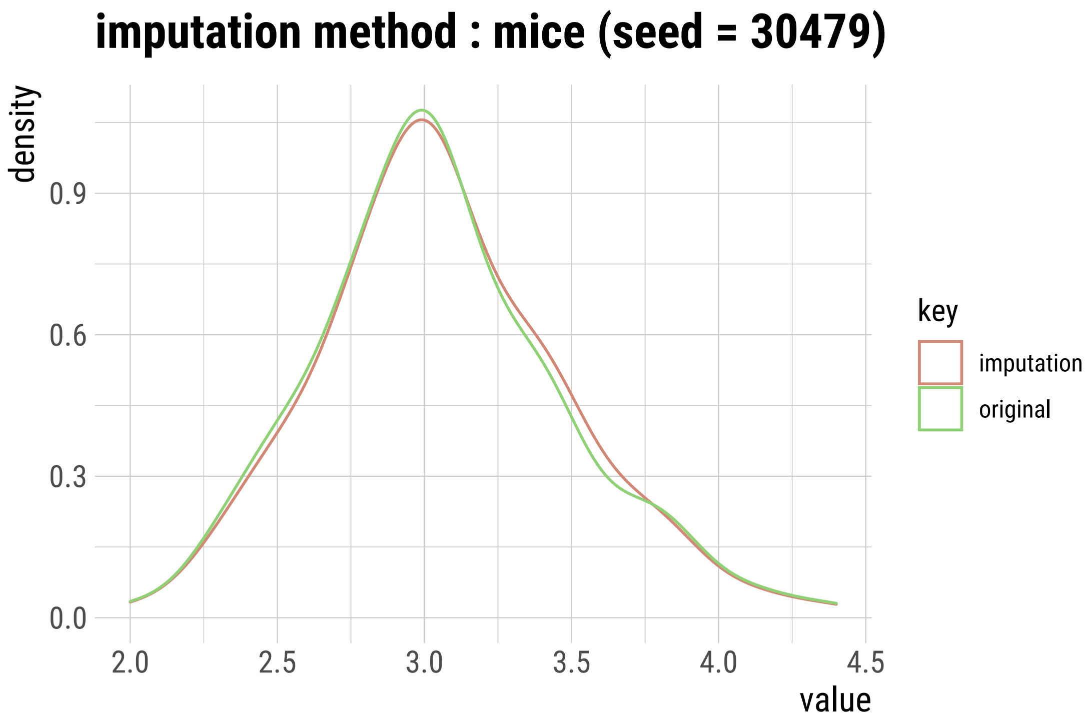
```
## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used
```

```
paste0("Number of Missing Values in After: ", sum(is.na(impute3$Sepal.Width)))
```

```
## [1] "Number of Missing Values in After: 0"
```

```
imputate_na(iris, xvar = Sepal.Width, yvar = "Species", method = "mice") %>% plot()
```

```
## 
##  iter imp variable
##   1   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   1   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   2   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   3   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   4   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   1  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   2  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   3  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   4  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width
##   5   5  Sepal.Length  Sepal.Width  Petal.Length  Petal.Width

## Warning in (function (kind = NULL, normal.kind = NULL, sample.kind = NULL) :
## non-uniform 'Rounding' sampler used
```


imputation method : mice (seed = 30479)
density
value
key
imputation
original