

Data Missing Imputation

Ali Valiyev

2024-03-22

```
#ilk olarak gerekli paketleri ve mice paketini sistemimize indirmeli ve kurmalyiz  
if(!require(mice)){  
  install.packages("mice"); library(mice)}
```

```
## Loading required package: mice
```

```
##
```

```
## Attaching package: 'mice'
```

```
## The following object is masked from 'package:stats':
```

```
##
```

```
##      filter
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      cbind, rbind
```

```
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##      filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
```

```
## v forcats 1.0.0      v readr  2.1.4
```

```
## v ggplot2  3.4.4      v stringr 1.5.0
```

```
## v lubridate 1.9.3      v tibble  3.2.1
```

```
## v purrr    1.0.2      v tidyr   1.3.0
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks mice::filter(), stats::filter()
## x dplyr::lag() masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

#Üzerinde işlemler yapacağımız Verisetine kayıp değerler ekleyelim

```
library(dplyr)
clone_diamonds = ggplot2::diamonds
first_diamonds = head(clone_diamonds, 1000 )

last_diamonds = tail(clone_diamonds, 20000)

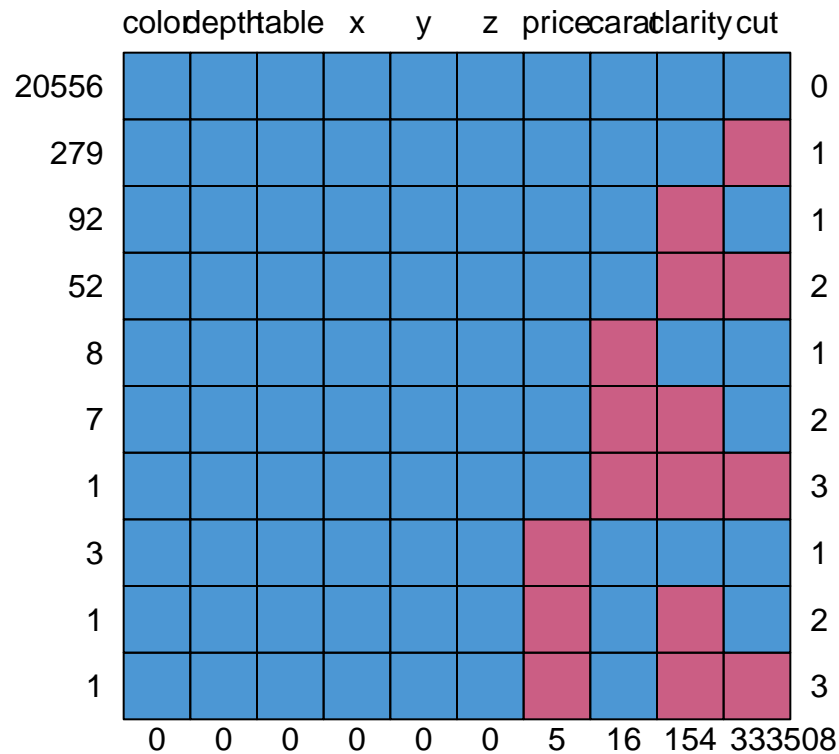
first_diamonds$price[first_diamonds$price == 405] <- NA
first_diamonds$carat[first_diamonds$carat == 1] <- NA
first_diamonds$cut[first_diamonds$cut == "Ideal"] <- NA
first_diamonds$color[first_diamonds$color == "L"] <- NA
first_diamonds$clarity[first_diamonds$clarity == "SI2"] <- NA

subData = full_join(first_diamonds, last_diamonds)
```

```
## Joining with 'by = join_by(carat, cut, color, clarity, depth, table, price, x,
## y, z)'
```

#md.pattern fonksiyonuyla patern-e bakalım

```
md.pattern(subData)
```



```
##      color depth table x y z price carat clarity cut
## 20556    1     1     1 1 1 1     1     1     1 1 0
## 279      1     1     1 1 1 1     1     1     1 0 1
## 92       1     1     1 1 1 1     1     1     0 1 1
## 52       1     1     1 1 1 1     1     1     0 0 2
## 8        1     1     1 1 1 1     1     0     1 1 1
## 7        1     1     1 1 1 1     1     0     0 1 2
## 1        1     1     1 1 1 1     1     0     0 0 3
## 3        1     1     1 1 1 1     0     1     1 1 1
## 1        1     1     1 1 1 1     0     1     0 1 2
## 1        1     1     1 1 1 1     0     1     0 0 3
##          0     0     0 0 0 0     5    16    154 333 508
```

```
#Kayıp değerlere bakalım
View(subData)
head(is.na(subData),10)
```

```
##      carat  cut color clarity depth table price      x      y      z
## [1,] FALSE TRUE FALSE      TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [2,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [3,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [4,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [5,] FALSE FALSE FALSE      TRUE FALSE FALSE FALSE FALSE FALSE FALSE
## [6,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [7,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [8,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [9,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [10,] FALSE FALSE FALSE    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
```

is.na() fonksiyonu sonuç olarak bize TRUE ve False-lardan ibaret bir data frame gösterecektir. TRUE - Kayıp değer var, FALSE - kayıp değer yoktur.

```
#Kayıp gözlem numarasını bulma:
head(which(is.na(subData)),100)
```

```
## [1] 285 325 370 385 466 473 635 715 720 751 879 880
## [13] 896 898 899 900 21001 21012 21014 21017 21040 21041 21042 21052
## [25] 21053 21056 21061 21063 21064 21066 21067 21083 21084 21091 21093 21103
## [37] 21105 21106 21108 21109 21110 21111 21112 21115 21116 21118 21119 21120
## [49] 21121 21122 21131 21133 21139 21140 21145 21150 21152 21156 21157 21160
## [61] 21164 21165 21168 21169 21171 21174 21175 21180 21181 21182 21183 21184
## [73] 21186 21192 21199 21208 21210 21213 21214 21215 21217 21218 21221 21225
## [85] 21230 21234 21235 21238 21241 21249 21250 21251 21257 21259 21263 21266
## [97] 21270 21274 21275 21279
```

```
#Kayıp gözlem satırlarını saptama
head(unique(unlist(lapply(subData, function(price) which(is.na(price))))),100)
```

```
## [1] 285 325 370 385 466 473 635 715 720 751 879 880 896 898 899 900 1 12
## [19] 14 17 40 41 42 52 53 56 61 63 64 66 67 83 84 91 93 103
## [37] 105 106 108 109 110 111 112 115 116 118 119 120 121 122 131 133 139 140
## [55] 145 150 152 156 157 160 164 165 168 169 171 174 175 180 181 182 183 184
## [73] 186 192 199 208 210 213 214 215 217 218 221 225 230 234 235 238 241 249
## [91] 250 251 257 259 263 266 270 274 275 279
```

```
#Kayip gozlem oranini bulma:
```

```
NA_Diamonds = function(subData) {sum(is.na(subData))/length(subData)*100}  
head(apply(subData,1,NA_Diamonds),100)
```

```
## [1] 20 0 0 0 10 0 0 0 0 0 0 10 0 20 10 0 20 0 0 0 10 0 0 0 0  
## [26] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 20 20 10 0 0 10 0 10 10 0 10  
## [51] 0 10 10 0 0 30 20 10 10 10 10 0 10 10 10 10 10 0 0 0 0 0 0 0  
## [76] 0 0 0 0 0 0 0 10 20 0 0 0 0 0 0 10 10 10 0 10 0 0 10 0 0
```

```
#Kayip degerlerin ortalama ile tamamlama:
```

```
new_diamonds <- subData  
new_diamonds$price.x[is.na(subData$price)] <- mean(subData$price, na.rm = T)
```

```
## Warning: Unknown or uninitialised column: 'price.x'.
```

```
head(new_diamonds,10)
```

```
## # A tibble: 10 x 11  
##   carat cut      color clarity depth table price      x      y      z price.x  
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>    <dbl>  
## 1  0.23 <NA>      E      <NA>    61.5    55    326  3.95  3.98  2.43      NA  
## 2  0.21 Premium  E      SI1      59.8    61    326  3.89  3.84  2.31      NA  
## 3  0.23 Good     E      VS1      56.9    65    327  4.05  4.07  2.31      NA  
## 4  0.29 Premium  I      VS2      62.4    58    334  4.2   4.23  2.63      NA  
## 5  0.31 Good     J      <NA>    63.3    58    335  4.34  4.35  2.75      NA  
## 6  0.24 Very Good J      VVS2     62.8    57    336  3.94  3.96  2.48      NA  
## 7  0.24 Very Good I      VVS1     62.3    57    336  3.95  3.98  2.47      NA  
## 8  0.26 Very Good H      SI1      61.9    55    337  4.07  4.11  2.53      NA  
## 9  0.22 Fair     E      VS2      65.1    61    337  3.87  3.78  2.49      NA  
## 10 0.23 Very Good H      VS1      59.4    61    338  4     4.05  2.39      NA
```

```
#imputeTS ile kayip deger Tamamlayalim
```

```
library(imputeTS)
```

```
## Registered S3 method overwritten by 'quantmod':
```

```
##   method      from  
##   as.zoo.data.frame zoo
```

```
new_diamonds = na.mean(subData, option = "mean")
```

```
## Warning: na.mean will be replaced by na_mean.
```

```
##   Functionality stays the same.
```

```
##   The new function name better fits modern R code style guidelines.
```

```
##   Please adjust your code accordingly.
```

```
## Warning: na_mean: No imputation performed for column 2 of the input dataset.
```

```
##   Reason: Input x is not numeric.
```

```
## Warning: na_mean: No imputation performed for column 4 of the input dataset.
```

```
##   Reason: Input x is not numeric.
```

```
head(new_diamonds,10)
```

```
## # A tibble: 10 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  0.23 <NA>      E      <NA>  61.5   55   326  3.95  3.98  2.43
## 2  0.21 Premium  E      SI1    59.8   61   326  3.89  3.84  2.31
## 3  0.23 Good     E      VS1    56.9   65   327  4.05  4.07  2.31
## 4  0.29 Premium  I      VS2    62.4   58   334  4.2   4.23  2.63
## 5  0.31 Good     J      <NA>  63.3   58   335  4.34  4.35  2.75
## 6  0.24 Very Good J      VVS2    62.8   57   336  3.94  3.96  2.48
## 7  0.24 Very Good I      VVS1    62.3   57   336  3.95  3.98  2.47
## 8  0.26 Very Good H      SI1    61.9   55   337  4.07  4.11  2.53
## 9  0.22 Fair     E      VS2    65.1   61   337  3.87  3.78  2.49
## 10 0.23 Very Good H      VS1    59.4   61   338  4     4.05  2.39
```

```
#imputeTS ile kayip deger Tamamlayalim
new_diamonds <- na_mean(subData, option = "median")
```

```
## Warning: na_mean: No imputation performed for column 2 of the input dataset.
##           Reason: Input x is not numeric.
```

```
## Warning: na_mean: No imputation performed for column 4 of the input dataset.
##           Reason: Input x is not numeric.
```

```
head(new_diamonds,10)
```

```
## # A tibble: 10 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>  <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 <NA>      E      <NA>  61.5   55   326  3.95  3.98  2.43
## 2  0.21 Premium  E      SI1    59.8   61   326  3.89  3.84  2.31
## 3  0.23 Good     E      VS1    56.9   65   327  4.05  4.07  2.31
## 4  0.29 Premium  I      VS2    62.4   58   334  4.2   4.23  2.63
## 5  0.31 Good     J      <NA>  63.3   58   335  4.34  4.35  2.75
## 6  0.24 Very Good J      VVS2    62.8   57   336  3.94  3.96  2.48
## 7  0.24 Very Good I      VVS1    62.3   57   336  3.95  3.98  2.47
## 8  0.26 Very Good H      SI1    61.9   55   337  4.07  4.11  2.53
## 9  0.22 Fair     E      VS2    65.1   61   337  3.87  3.78  2.49
## 10 0.23 Very Good H      VS1    59.4   61   338  4     4.05  2.39
```

```
#imputeTS ile kayip deger Tamamlayalim
new_diamonds <- na_mean(subData, option = "mode")
```

```
## Warning: na_mean: No imputation performed for column 2 of the input dataset.
##           Reason: Input x is not numeric.
```

```
## Warning: na_mean: No imputation performed for column 4 of the input dataset.
##           Reason: Input x is not numeric.
```

```
head(new_diamonds,10)
```

```
## # A tibble: 10 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  0.23 <NA>      E     <NA>   61.5    55   326   3.95   3.98   2.43
## 2  0.21 Premium  E     SI1     59.8    61   326   3.89   3.84   2.31
## 3  0.23 Good     E     VS1     56.9    65   327   4.05   4.07   2.31
## 4  0.29 Premium  I     VS2     62.4    58   334   4.2    4.23   2.63
## 5  0.31 Good     J     <NA>   63.3    58   335   4.34   4.35   2.75
## 6  0.24 Very Good J     VVS2    62.8    57   336   3.94   3.96   2.48
## 7  0.24 Very Good I     VVS1    62.3    57   336   3.95   3.98   2.47
## 8  0.26 Very Good H     SI1     61.9    55   337   4.07   4.11   2.53
## 9  0.22 Fair     E     VS2     65.1    61   337   3.87   3.78   2.49
## 10 0.23 Very Good H     VS1     59.4    61   338   4      4.05   2.39
```

```
#Hmisc ile kayip deger Tamamlayalim
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##   src, summarize

## The following objects are masked from 'package:base':
##
##   format.pval, units
```

```
head(impute(subData$price, mean),100)
```

```
##      1      2      3      4      5      6      7      8
## 326.000 326.000 327.000 334.000 335.000 336.000 336.000 337.000
##      9     10     11     12     13     14     15     16
## 337.000 338.000 339.000 340.000 342.000 344.000 345.000 345.000
##     17     18     19     20     21     22     23     24
## 348.000 351.000 351.000 351.000 351.000 352.000 353.000 353.000
##     25     26     27     28     29     30     31     32
## 353.000 354.000 355.000 357.000 357.000 357.000 402.000 402.000
##     33     34     35     36     37     38     39     40
## 402.000 402.000 402.000 402.000 402.000 402.000 403.000 403.000
##     41     42     43     44     45     46     47     48
## 403.000 403.000 403.000 403.000 403.000 403.000 403.000 403.000
##     49     50     51     52     53     54     55     56
## 404.000 404.000 404.000 404.000 404.000 404.000 404.000 1524.504*
##     57     58     59     60     61     62     63     64
## 1524.504* 1524.504* 1524.504* 1524.504* 552.000 552.000 552.000 552.000
##     65     66     67     68     69     70     71     72
## 552.000 553.000 553.000 553.000 553.000 553.000 553.000 554.000
##     73     74     75     76     77     78     79     80
```

```
## 554.000 554.000 554.000 554.000 554.000 554.000 554.000 554.000
## 81 82 83 84 85 86 87 88
## 554.000 554.000 554.000 554.000 554.000 554.000 554.000 554.000
## 89 90 91 92 93 94 95 96
## 554.000 554.000 2757.000 2757.000 2757.000 2759.000 2759.000 2759.000
## 97 98 99 100
## 2759.000 2759.000 2760.000 2760.000
```

```
#Hmisc ile kayıp deger Tamamlayalım
head(impute(clone_diamonds$price, median),100)
```

```
## [1] 326 326 327 334 335 336 336 337 337 338 339 340 342 344 345
## [16] 345 348 351 351 351 351 352 353 353 353 354 355 357 357 357
## [31] 402 402 402 402 402 402 402 402 402 403 403 403 403 403 403
## [46] 403 403 403 404 404 404 404 404 404 404 405 405 405 405 405
## [61] 552 552 552 552 552 553 553 553 553 553 553 554 554 554 554
## [76] 554 554 554 554 554 554 554 554 554 554 554 554 554 554 554
## [91] 2757 2757 2757 2759 2759 2759 2759 2759 2760 2760
```

```
NumOfNA = which(is.na(subData))
length(NumOfNA)
```

```
## [1] 508
```

Bu işlem sayesinde kaç tane kayıp gözlemin olduğuna bakabiliriz. 508 Kayıp gözlemimiz var.

```
newData = na.omit(subData)
head(newData,10)
```

```
## # A tibble: 10 x 10
##   carat cut      color clarity depth table price      x      y      z
##   <dbl> <ord>    <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.21 Premium E      SI1      59.8 61 326 3.89 3.84 2.31
## 2 0.23 Good E      VS1      56.9 65 327 4.05 4.07 2.31
## 3 0.29 Premium I      VS2      62.4 58 334 4.2 4.23 2.63
## 4 0.24 Very Good J      VVS2      62.8 57 336 3.94 3.96 2.48
## 5 0.24 Very Good I      VVS1      62.3 57 336 3.95 3.98 2.47
## 6 0.26 Very Good H      SI1      61.9 55 337 4.07 4.11 2.53
## 7 0.22 Fair E      VS2      65.1 61 337 3.87 3.78 2.49
## 8 0.23 Very Good H      VS1      59.4 61 338 4 4.05 2.39
## 9 0.3 Good J      SI1      64 55 339 4.25 4.28 2.73
## 10 0.22 Premium F      SI1      60.4 61 342 3.88 3.84 2.33
```

na.omit() fonksiyonu NA kayıp gözlemlerini ortadan kaldırmış oluyoruz. naomit() işlemi Kayıp değer olduğu sadece bir değişkenin değerini değil, Kayıp değer olduğu tüm satırı kaldırmış oluyor. Bu yöntem yeteri kadar işlevsel bir yöntem olduğu söylenemez

```
nrow(subData)
```

```
## [1] 21000
```

```
nrow(newData)
```

```
## [1] 20556
```

yeni nesnemizde 556 kadar veri kaldırılmış durumdadır. Kaldırılmış verilerin sayına bakarsak bu yöntemin yeteri kadar işlevsel bir yöntem olmadığını görüyoruz. Eğer aynı satırın çok fazla sayıda değişkenleri Kayıp değere sahipse na.omit() fonksiyonunu kullanarak bu satırı Kaldıramamız na.omit() fonksiyonu için uygun bir kullanım noktası olmuş olur. Şu an ki veriseti için başka yöntemler deneyeceğiz.

```
#Kayip gozlem oranlarına ve Pattern-e bakalım:  
library(VIM)
```

```
## Loading required package: colorspace
```

```
## Loading required package: grid
```

```
## VIM is ready to use.
```

```
## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues
```

```
##
```

```
## Attaching package: 'VIM'
```

```
## The following object is masked from 'package:datasets':
```

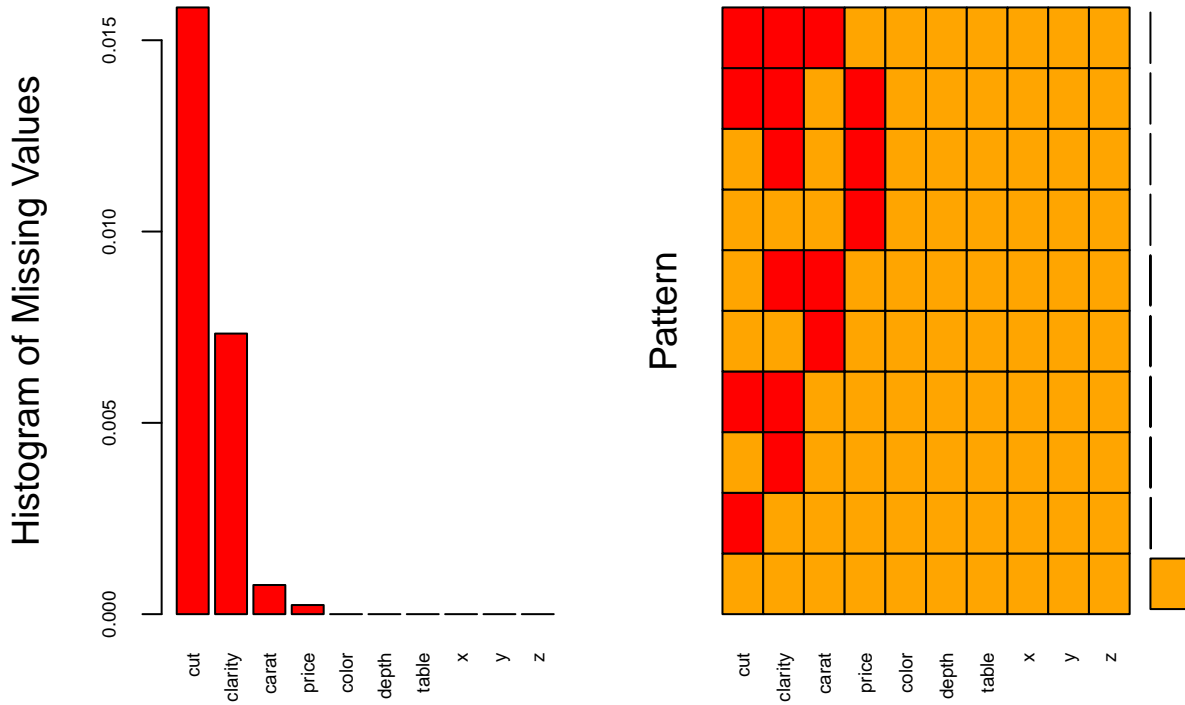
```
##
```

```
##      sleep
```

```
fig = agr(subData , col=c("orange" , "red") , labels = names(subData),  
          numbers = TRUE , sortVars = TRUE, cex.axis = 0.6 ,  
          ylab = c("Histogram of Missing Values" , "Pattern"))
```

```
## Warning in plot.aggr(res, ...): not enough horizontal space to display
```

```
## frequencies
```

```
##
## Variables sorted by number of missings:
## Variable      Count
##      cut 0.0158571429
##      clarity 0.0073333333
##      carat 0.0007619048
##      price 0.0002380952
##      color 0.0000000000
##      depth 0.0000000000
##      table 0.0000000000
##      x 0.0000000000
##      y 0.0000000000
##      z 0.0000000000
```

Mice Paketini kullanarak kayıp gözlemleri dolduralım mice paketini nasıl kullanavagimizi öğrenelim mice (data - bu kısmı bir veriseti ilave ediyoruz

m - bu kısımda yapılacak olan imputasyonların sayını belirtiyoruz

method - imputasyon yaparken kullacağımız methodların isimlerini belirtiyoruz. Birden çok method ismi yazarsak her bir değişkene uygun methodları kullanacaktır.

defaultMethod - method = null olduğunda bu kısım çalışıyor. 4 değer veriyoruz. 1.- numeric data için, 2. - 2 seviyeli faktörlere uygulanan method olacak, 3. - üçden daha fazla faktör seviyeli ve sıralı olmayanlar için 4.- ikiden fazla faktör seviyeli ve sıralı olan değişkenler için (“any” , “numeric” , “Unordered” “Ordered”)

maxit - imputation zamanı kaç kere iterasyon yapılacağını belirtiyoruz

Methodlar: pmm - Predictive mean matching, her türlü datada kullanılabılır midastouch - Weighted predictive mean matching, her türlü veri türünde kullanılabılır rf - Random forest Imputation, her türlü veri türünde kullanılabılır mean - Unconditional mean imputation. numeric tipdeki veri türünde kullanılabılır

```
#degiskenimizin tiplerine bakalim  
#ordered:  
head(subData$color,10)
```

```
## [1] E E E I J J I H E H  
## Levels: D < E < F < G < H < I < J
```

```
head(subData$clarity,10)
```

```
## [1] <NA> SI1 VS1 VS2 <NA> VVS2 VVS1 SI1 VS2 VS1  
## Levels: I1 < SI2 < SI1 < VS2 < VS1 < VVS2 < VVS1 < IF
```

```
head(subData$cut,10)
```

```
## [1] <NA> Premium Good Premium Good Very Good Very Good  
## [8] Very Good Fair Very Good  
## Levels: Fair < Good < Very Good < Premium < Ideal
```

```
#numeric  
head(subData$carat,10)
```

```
## [1] 0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23
```

```
head(subData$depth,10)
```

```
## [1] 61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4
```

```
head(subData$table,10)
```

```
## [1] 55 61 65 58 58 57 57 55 61 61
```

```
head(subData$x,10)
```

```
## [1] 3.95 3.89 4.05 4.20 4.34 3.94 3.95 4.07 3.87 4.00
```

```
head(subData$y,10)
```

```
## [1] 3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05
```

```
head(subData$z,10)
```

```
## [1] 2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39
```

```
ImpData = mice(data = subData, m = 3 , maxit = 3,
               method = NULL, defaultMethod = c("norm.nob" , "logreg", "rf" , "pmm"))
```

```
##
## iter imp variable
## 1 1 carat cut clarity price
## 1 2 carat cut clarity price
## 1 3 carat cut clarity price
## 2 1 carat cut clarity price
## 2 2 carat cut clarity price
## 2 3 carat cut clarity price
## 3 1 carat cut clarity price
## 3 2 carat cut clarity price
## 3 3 carat cut clarity price
```

mice fonksiyonu çalışırken 3 output ortaya çıkmış oluyor. Bu output-lar 1. imputasyonları daha sonra imputasyonları ve en sonda da değişkenleri gösteriyor. 1. İterasyonda 1. İmputasyon ve hangi değişkenler için yapıldığı gösteriliyor. daha sonra ikinci ve üçüncüsü imputasyonları yapıyor. 1. İterasyonu bitirdikten sonra 2. İmputasyona geçiyor. 3. İterasyon ve 3.İmputasyona vardığında sonlanacaktır. Çünkü biz maxit = 3 ve m = 3 olarak belirttik. Tüm İşlemler sonuçlandıktan sonra da sonuçları ImpData nesnesine atıyor.

```
summary(ImpData)
```

```
## Class: mids
## Number of multiple imputations: 3
## Imputation methods:
##      carat      cut      color      clarity      depth      table      price
## "norm.nob"      "pmm"      ""      "pmm"      ""      ""      "norm.nob"
##      x      y      z
##      ""      ""      ""
## PredictorMatrix:
##      carat cut color clarity depth table price x y z
## carat      0  1    1      1      1      1      1 1 1 1
## cut        1  0    1      1      1      1      1 1 1 1
## color      1  1    0      1      1      1      1 1 1 1
## clarity    1  1    1      0      1      1      1 1 1 1
## depth      1  1    1      1      0      1      1 1 1 1
## table      1  1    1      1      1      0      1 1 1 1
```

Imputation methods - Imputation işlemi yaparken kullanılan metodları gösteriyor.

```
names(ImpData)
```

```
## [1] "data"      "imp"      "m"      "where"
## [5] "blocks"    "call"     "nmis"    "method"
## [9] "predictorMatrix" "visitSequence" "formulas" "post"
## [13] "blots"     "ignore"   "seed"    "iteration"
## [17] "lastSeedValue" "chainMean" "chainVar" "loggedEvents"
## [21] "version"   "date"
```

```
ImpData$m
```

```
## [1] 3
```

Yani İmputasyon sayısı 3-e eşittir

```
head(ImpData$imp$cut,10)
```

```
##           1           2           3
## 1      Good   Premium   Premium
## 12     Ideal Very Good   Ideal
## 14     Ideal     Ideal   Ideal
## 17     Ideal     Ideal   Ideal
## 40     Ideal   Premium Very Good
## 41 Very Good     Ideal   Ideal
## 42     Ideal     Ideal   Ideal
## 52     Ideal     Ideal   Ideal
## 53     Ideal     Ideal   Ideal
## 56 Very Good     Good Very Good
```

```
head(ImpData$imp$clarity,10)
```

```
##           1           2           3
## 1    VS1    VS2 VVS1
## 5   VVS2    IF    IF
## 14 VVS1 VVS1 SI1
## 15 VVS1 VVS1 VVS1
## 17    IF VVS1 VS1
## 21 VS1 VVS1 VS1
## 40 VS1 VS1 VVS1
## 41 VS1 VS1 VS2
## 45 VS2 VS1 VS1
## 47 VVS1 VS1 VVS1
```

```
head(ImpData$imp$color,10)
```

```
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
```

```
head(ImpData$imp$depth,10)
```

```
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
```

```
head(ImpData$imp$table,10)
```

```
## [1] 1 2 3
## <0 rows> (or 0-length row.names)
```

```
head(ImpData$imp$price,10)
```

```
##           1           2           3
## 56  15.83625 312.110501 212.7719
## 57   52.19438 135.433797 308.0918
## 58  490.16458 -82.770806 -118.1728
## 59  209.19162 -3.031182 -185.5136
## 60 -212.11722 177.643667 -144.6215
```

İmputasyon işlemlerinin çıkardığı sonuçları inceleyelim.

```
head(ImpData$imp$cut,10)
```

```
##           1           2           3
## 1      Good   Premium   Premium
## 12     Ideal Very Good   Ideal
## 14     Ideal   Ideal     Ideal
## 17     Ideal   Ideal     Ideal
## 40     Ideal   Premium Very Good
## 41 Very Good   Ideal     Ideal
## 42     Ideal   Ideal     Ideal
## 52     Ideal   Ideal     Ideal
## 53     Ideal   Ideal     Ideal
## 56 Very Good   Good Very Good
```

Mesela cut değişkenine bakalım. İnceleme sonucunda bazı değerlerin 1. imputasyon zamanı verilen değer daha sonraki imputasyonlarda yerini başka bir değere bırakmış durumdadır. Diğer değişkenlerde de böyle bir durum yaşanabilir.

Kayıp değerlere değer atamak için complete fonksiyonunu kullanıyoruz complete(hangi verisetini kullanacağımızı belirtiyoruz hangi imputasyon işleminin sonuçlarını kullanacağımızı belirtiyoruz.)

```
newData = complete(ImpData, 3)
```

```
View(newData)
```

Artık Kayıp değerlere değer atanmıştır