Handling-missing-values.R

patriciamaya

2020-12-06

```
#HANDLING MISSING VALUES

data <- read.csv("~/Downloads/hmeq.csv")
summary(data)
```

```
##       BAD              LOAN          MORTDUE           VALUE
##  Min.   :0.0000   Min.   : 1100   Min.   :  2063   Min.   :  8000
##  1st Qu.:0.0000   1st Qu.:11100   1st Qu.: 46276   1st Qu.: 66076
##  Median :0.0000   Median :16300   Median : 65019   Median : 89236
##  Mean   :0.1995   Mean   :18608   Mean   : 73761   Mean   :101776
##  3rd Qu.:0.0000   3rd Qu.:23300   3rd Qu.: 91488   3rd Qu.:119824
##  Max.   :1.0000   Max.   :89900   Max.   :399550   Max.   :855909
##                                   NA's   :518      NA's   :112
##     REASON              JOB                 YOJ             DEROG
##  Length:5960        Length:5960        Min.   : 0.000   Min.   : 0.0000
##  Class :character   Class :character   1st Qu.: 3.000   1st Qu.: 0.0000
##  Mode  :character   Mode  :character   Median : 7.000   Median : 0.0000
##                                        Mean   : 8.922   Mean   : 0.2546
##                                        3rd Qu.:13.000   3rd Qu.: 0.0000
##                                        Max.   :41.000   Max.   :10.0000
##                                        NA's   :515      NA's   :708
##      DELINQ           CLAGE             NINQ             CLNO
##  Min.   : 0.0000   Min.   :   0.0   Min.   : 0.000   Min.   : 0.0
##  1st Qu.: 0.0000   1st Qu.: 115.1   1st Qu.: 0.000   1st Qu.:15.0
##  Median : 0.0000   Median : 173.5   Median : 1.000   Median :20.0
##  Mean   : 0.4494   Mean   : 179.8   Mean   : 1.186   Mean   :21.3
##  3rd Qu.: 0.0000   3rd Qu.: 231.6   3rd Qu.: 2.000   3rd Qu.:26.0
##  Max.   :15.0000   Max.   :1168.2   Max.   :17.000   Max.   :71.0
##  NA's   :580       NA's   :308      NA's   :510      NA's   :222
##     DEBTINC
##  Min.   :  0.5245
##  1st Qu.: 29.1400
##  Median : 34.8183
##  Mean   : 33.7799
##  3rd Qu.: 39.0031
##  Max.   :203.3121
##  NA's   :1267
```

```
# Handling missing values of NUMERICAL variables

sum(is.na(data$NINQ)) # This give you the number of missing values in the variable NINQ
```

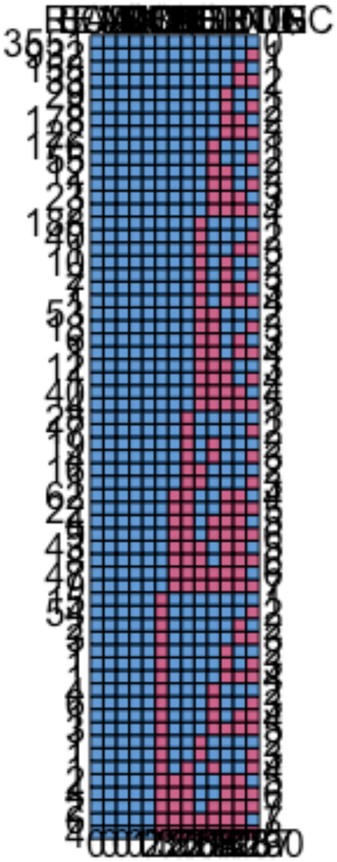
```
## [1] 510
```

```
sum(complete.cases(data$NINQ)) # Count of complete cases in the variable NINQ
```

```
## [1] 5450
```

```
sum(!complete.cases(data$NINQ)) # Count of NOT complete cases in the variable
NINQ

## [1] 510

which(!complete.cases(data$NINQ)) # Which cases (row numbers) are NOT complet
e

##   [1]    4   11   18   52   64   74   96  106  113  116  128  140  144  14
5  146
##  [16]  153  155  160  165  166  170  172  174  187  191  212  218  222  22
7  230
##  [31]  232  238  240  242  246  252  266  269  285  293  300  303  305  31
0  318
##  [46]  323  331  334  337  339  343  344  347  351  353  357  358  359  36
2  366
##  [61]  367  375  381  382  390  392  396  397  399  402  414  418  421  42
5  432
##  [76]  435  444  465  469  473  482  490  503  527  532  536  537  545  55
0  561
##  [91]  566  567  597  601  604  609  619  634  643  645  649  654  669  68
8  692
## [106]  703  711  717  726  735  737  748  749  752  763  765  770  772  77
8  783
## [121]  786  790  812  818  830  844  854  858  865  868  882  899  922  92
9  932
## [136]  933  935  947  970  974  975  980  987  988  992 1011 1031 1040 104
8 1076
## [151] 1084 1092 1094 1106 1123 1138 1145 1146 1153 1155 1157 1182 1196 121
0 1224
## [166] 1236 1238 1244 1249 1254 1257 1276 1296 1321 1333 1336 1339 1348 136
1 1364
## [181] 1373 1389 1395 1402 1406 1411 1417 1422 1426 1427 1434 1468 1488 148
9 1505
## [196] 1508 1532 1554 1556 1568 1571 1573 1589 1592 1628 1635 1645 1646 166
3 1675
## [211] 1688 1690 1695 1736 1766 1775 1788 1790 1824 1842 1864 1878 1895 189
8 1960
## [226] 1961 1965 1967 1989 1998 2062 2067 2073 2102 2108 2113 2121 2127 215
4 2166
## [241] 2168 2218 2244 2246 2266 2267 2297 2304 2309 2310 2342 2355 2357 237
9 2397
## [256] 2412 2417 2427 2440 2450 2464 2473 2476 2515 2518 2543 2551 2588 262
6 2675
## [271] 2680 2686 2690 2743 2752 2759 2814 2876 2891 2906 2936 2970 2984 298
8 2993
## [286] 2998 3006 3047 3051 3062 3077 3097 3125 3135 3136 3141 3152 3157 319
7 3250
## [301] 3308 3351 3366 3385 3392 3478 3491 3518 3556 3582 3601 3621 3623 362
4 3631
```

```
## [316] 3652 3673 3683 3695 3697 3706 3721 3737 3744 3746 3747 3762 3768 376
9 3777
## [331] 3817 3822 3828 3835 3840 3842 3844 3856 3860 3874 3886 3918 3937 394
3 3953
## [346] 3955 3978 3993 4003 4013 4017 4036 4072 4073 4074 4097 4113 4119 412
2 4123
## [361] 4128 4131 4142 4157 4160 4162 4178 4200 4207 4211 4230 4242 4252 425
8 4273
## [376] 4274 4279 4281 4295 4303 4309 4321 4322 4323 4325 4343 4344 4351 435
2 4361
## [391] 4364 4366 4382 4385 4387 4393 4402 4414 4419 4421 4425 4430 4432 443
8 4439
## [406] 4476 4480 4482 4500 4542 4544 4549 4567 4573 4575 4576 4582 4585 458
9 4600
## [421] 4610 4633 4645 4657 4661 4671 4672 4681 4683 4698 4716 4748 4759 478
4 4785
## [436] 4786 4790 4791 4795 4801 4803 4818 4846 4851 4852 4858 4866 4868 487
0 4881
## [451] 4900 4910 4932 4942 4946 4948 4973 4975 4976 4989 4991 5000 5045 504
9 5061
## [466] 5072 5076 5102 5107 5113 5127 5148 5165 5200 5229 5230 5248 5253 526
5 5274
## [481] 5275 5314 5341 5347 5366 5386 5418 5423 5434 5452 5464 5467 5472 549
2 5496
## [496] 5699 5701 5702 5719 5746 5749 5751 5752 5755 5757 5760 5763 5766 580
9 5838

# The function “na.omit()” DELETES ALL instances with missing values and retu
rns
# the object with listwise deletion of missing values.

NINQ_Imputed = na.omit(data$NINQ) # Create new variable without missing value
s
sum(is.na(NINQ_Imputed))

## [1] 0

# REPLACE missing values by a particular value (mean)
data$NINQ[is.na(data$NINQ)] = mean(data$NINQ, na.rm=TRUE) # Recode all NA in
NINQ as the average value
sum(is.na(data$NINQ))

## [1] 0

#  REPLACING using mice for looking at missing DATA PATTERN
library(mice)

md.pattern(data)
```



```
##      BAD LOAN REASON JOB NINQ VALUE CLNO CLAGE YOJ MORTDUE DELINQ DEROG DEBTINC
## 3551   1    1      1   1    1     1    1     1   1       1      1     1       1
## 932    1    1      1   1    1     1    1     1   1       1      1     1       0
## 158    1    1      1   1    1     1    1     1   1       1      1     0       1
## 33     1    1      1   1    1     1    1     1   1       1      1     0       0
## 29     1    1      1   1    1     1    1     1   1       1      0     1       1
## 8      1    1      1   1    1     1    1     1   1       1      0     1       0
## 178    1    1      1   1    1     1    1     1   1       1      0     0       1
## 22     1    1      1   1    1     1    1     1   1       1      0     0       0
## 176    1    1      1   1    1     1    1     1   1       0      1     1       1
## 55     1    1      1   1    1     1    1     1   1       0      1     1       0
## 12     1    1      1   1    1     1    1     1   1       0      0     1       1
## 1      1    1      1   1    1     1    1     1   1       0      0     1       0
## 23     1    1      1   1    1     1    1     1   1       0      0     0       1
## 2      1    1      1   1    1     1    1     1   1       0      0     0       0
## 188    1    1      1   1    1     1    1     1   0       1      1     1       1
## 40     1    1      1   1    1     1    1     1   0       1      1     1       0
## 1      1    1      1   1    1     1    1     1   0       1      1     0       0
## 10     1    1      1   1    1     1    1     1   0       1      0     1       1
## 2      1    1      1   1    1     1    1     1   0       1      0     1       0
## 7      1    1      1   1    1     1    1     1   0       1      0     0       1
## 1      1    1      1   1    1     1    1     1   0       1      0     0       0
## 53     1    1      1   1    1     1    1     1   0       0      1     1       1
## 13     1    1      1   1    1     1    1     1   0       0      1     1       0
## 6      1    1      1   1    1     1    1     1   0       0      1     0       1
## 1      1    1      1   1    1     1    1     1   0       0      1     0       0
## 12     1    1      1   1    1     1    1     1   0       0      0     1       1
## 1      1    1      1   1    1     1    1     1   0       0      0     1       0
## 40     1    1      1   1    1     1    1     1   0       0      0     0       1
## 4      1    1      1   1    1     1    1     1   0       0      0     0       0
## 28     1    1      1   1    1     1    1     0   1       1      1     1       1
## 17     1    1      1   1    1     1    1     0   1       1      1     1       0
## 19     1    1      1   1    1     1    1     0   1       0      1     1       1
## 4      1    1      1   1    1     1    1     0   1       0      1     1       0
## 16     1    1      1   1    1     1    1     0   0       1      1     1       1
## 1      1    1      1   1    1     1    1     0   0       1      1     1       0
## 62     1    1      1   1    1     1    0     0   1       1      0     0       1
## 22     1    1      1   1    1     1    0     0   1       1      0     0       0
## 4      1    1      1   1    1     1    0     0   1       0      0     0       1
## 9      1    1      1   1    1     1    0     0   1       0      0     0       0
## 43     1    1      1   1    1     1    0     0   0       1      0     0       1
## 8      1    1      1   1    1     1    0     0   0       1      0     0       0
## 47     1    1      1   1    1     1    0     0   0       0      0     0       1
## 9      1    1      1   1    1     1    0     0   0       0      0     0       0
## 15     1    1      1   1    1     0    1     1   1       1      1     1       1
## 54     1    1      1   1    1     0    1     1   1       1      1     1       0
## 2      1    1      1   1    1     0    1     1   1       1      1     0       1
## 3      1    1      1   1    1     0    1     1   1       1      1     0       0
## 1      1    1      1   1    1     0    1     1   1       1      0     1       1
## 1      1    1      1   1    1     0    1     1   1       1      0     1       0
## 1      1    1      1   1    1     0    1     1   1       1      0     0       0
## 4      1    1      1   1    1     0    1     1   1       0      1     1       1
## 6      1    1      1   1    1     0    1     1   1       0      1     1       0
## 1      1    1      1   1    1     0    1     1   1       0      1     0       0
## 3      1    1      1   1    1     0    1     1   1       0      0     0       0
## 1      1    1      1   1    1     0    1     1   0       1      1     1       1
## 1      1    1      1   1    1     0    1     1   0       1      1     1       0
## 1      1    1      1   1    1     0    1     0   1       0      1     1       0
## 2      1    1      1   1    1     0    0     0   1       1      0     0       1
## 4      1    1      1   1    1     0    0     0   1       1      0     0       0
## 2      1    1      1   1    1     0    0     0   1       0      0     0       0
## 6      1    1      1   1    1     0    0     0   0       0      0     0       1
## 4      1    1      1   1    1     0    0     0   0       0      0     0       0
##        0    0      0   0    0   112  222   308 515     518    580   708    1267
##
## 3551    0
## 932     1
## 158     1
## 33      2
## 29      1
## 8       2
## 178     2
## 22      3
## 176     1
## 55      2
## 12      2
## 1       3
## 23      3
## 2       4
## 188     1
## 40      2
## 1       3
## 10      2
## 2       3
## 7       3
## 1       4
## 53      2
## 13      3
## 6       3
## 1       4
## 12      3
## 1       4
## 40      4
## 4       5
## 28      1
## 17      2
## 19      2
## 4       3
## 16      2
## 1       3
## 62      4
## 22     5
## 4      5
## 9      6
## 43     5
## 8      6
## 6      3
## 1      4
## 3      5
## 1      2
## 1      3
## 1      4
## 2      5
## 4      6
## 2      7
## 6      7
## 4      8
##     4230
```

```
# The output tells us that 3551 samples are complete, 932 samples miss only DEBTINC, 158 samples miss only the DEROG and so on.

# The “mice()” function takes care of imputing process.
NewData = mice(data, m=5, maxit=50, meth="pmm", seed=500)
```

```
summary(NewData)
```

```
## Class: mids
## Number of multiple imputations:  5 
## Imputation methods:
##     BAD    LOAN MORTDUE   VALUE  REASON     JOB     YOJ   DEROG  DELINQ   CLAGE 
##      ""      ""   "pmm"   "pmm"      ""      ""   "pmm"   "pmm"   "pmm"   "pmm" 
##    NINQ    CLNO DEBTINC 
##      ""   "pmm"   "pmm" 
## PredictorMatrix:
##         BAD LOAN MORTDUE VALUE REASON JOB YOJ DEROG DELINQ CLAGE NINQ CLNO
## BAD       0    1       1     1      0   0   1     1      1     1    1    1
## LOAN      1    0       1     1      0   0   1     1      1     1    1    1
## MORTDUE   1    1       0     1      0   0   1     1      1     1    1    1
## VALUE     1    1       1     0      0   0   1     1      1     1    1    1
## REASON    1    1       1     1      0   0   1     1      1     1    1    1
## JOB       1    1       1     1      0   0   1     1      1     1    1    1
##         DEBTINC
## BAD           1
## LOAN          1
## MORTDUE       1
## VALUE         1
## REASON        1
## JOB           1
## Number of logged events:  2 
##   it im dep     meth    out
## 1  0  0     constant REASON
## 2  0  0     constant    JOB
```

```
  # m=5 refers to the number of imputed datasets. Five is the default value
  # meth=’pmm’ refers to the imputation method.
     #In this case we are using predictive mean matching as imputation method

# We can get back the completed dataset using the complete() function
New_data <- as.data.frame(complete(NewData, 1))
head(New_data)
```

```
##   BAD LOAN MORTDUE  VALUE  REASON    JOB  YOJ DEROG DELINQ     CLAGE     N
INQ
## 1   1 1100   25860  39025 HomeImp  Other 10.5     0      0  94.36667 1.000
000
## 2   1 1300   70053  68400 HomeImp  Other  7.0     0      2 121.83333 0.000
000
## 3   1 1500   13500  16700 HomeImp  Other  4.0     0      0 149.46667 1.000
000
## 4   1 1500   72136  85100                22.0     0      0  62.35974 1.186
055
## 5   0 1700   97800 112000 HomeImp Office  3.0     0      0  93.33333 0.000
000
## 6   1 1700   30548  40320 HomeImp  Other  9.0     0      0 101.46600 1.000
000
##   CLNO  DEBTINC
## 1    9 30.60700
## 2   14 42.58162
## 3   10 37.33952
## 4   15 36.43872
## 5   14 29.29518
## 6    8 37.11361

#Handling missing values of CATEGORICAL variables:

#As far as categorical variables are concerned, replacing categorical variabl
es is usually not
#advisable. Some common practice include replacing missing categorical variab
les with the
#mode of the observed ones, however, it is questionable whether it is a good
choice.

data$REASON <- as.factor(data$REASON)
data$JOB <- as.factor(data$JOB)

#remove NA values if only instances of categorical variables are missing
Data <- data[complete.cases(data), ]  #removes ALL rows where a value is miss
ing


#we need to first deal with numerical missing values and then categorical var
iables.
```