Cleaning Data - Imputation in R

Michael Kozloff

May 29, 2020

Baby Data - Loading in Data with NAs

```
nhanes
```

```
##    age  bmi hyp chl
## 1    1   NA  NA  NA
## 2    2 22.7   1 187
## 3    1   NA   1 187
## 4    3   NA  NA  NA
## 5    1 20.4   1 113
## 6    3   NA  NA 184
## 7    1 22.5   1 118
## 8    1 30.1   1 187
## 9    2 22.0   1 238
## 10   2   NA  NA  NA
## 11   1   NA  NA  NA
## 12   2   NA  NA  NA
## 13   3 21.7   1 206
## 14   2 28.7   2 204
## 15   1 29.6   1  NA
## 16   1   NA  NA  NA
## 17   3 27.2   2 284
## 18   2 26.3   2 199
## 19   1 35.3   1 218
## 20   3 25.5   2  NA
## 21   1   NA  NA  NA
## 22   1 33.2   1 229
## 23   1 27.5   1 131
## 24   3 24.9   1  NA
## 25   2 27.4   1 186
```

We have 25 observations in our dataset. These observations are babys' cardiovascular characteristics, in particular their BMI, hypertension, and cholesterol. BMI is a decimal value, hypertension is a dummy variable (1 being no hypertension, 2 being the existence of hypertension) and cholesterol levels a decimal value). The goal of this unclean data set is to clean it with imputated values.

Analyzing the Distribution of the Data

```
baby_data <- nhanes
summary(baby_data)
```

```
##       age            bmi             hyp             chl
##  Min.   :1.00   Min.   :20.40   Min.   :1.000   Min.   :113.0
```

```
##  1st Qu.:1.00   1st Qu.:22.65   1st Qu.:1.000   1st Qu.:185.0
##  Median :2.00   Median :26.75   Median :1.000   Median :187.0
##  Mean   :1.76   Mean   :26.56   Mean   :1.235   Mean   :191.4
##  3rd Qu.:2.00   3rd Qu.:28.93   3rd Qu.:1.000   3rd Qu.:212.0
##  Max.   :3.00   Max.   :35.30   Max.   :2.000   Max.   :284.0
##                 NA's   :9       NA's   :8       NA's   :10
```

```
# Tells us the amount of NA's we have for each variable
# Note hyp (hypertension) is a binary variable, so we want to make sure R sees it that way
baby_data$hyp= as.factor(baby_data$hyp)
summary(baby_data)
```

```
##       age             bmi           hyp          chl
##  Min.   :1.00   Min.   :20.40   1   :13   Min.   :113.0
##  1st Qu.:1.00   1st Qu.:22.65   2   : 4   1st Qu.:185.0
##  Median :2.00   Median :26.75   NA's: 8   Median :187.0
##  Mean   :1.76   Mean   :26.56             Mean   :191.4
##  3rd Qu.:2.00   3rd Qu.:28.93             3rd Qu.:212.0
##  Max.   :3.00   Max.   :35.30             Max.   :284.0
##                 NA's   :9                 NA's   :10
```

Simple Imputation with Mean Substitution

```
#Goal: substitute the missing bmi values with the mean of the existing bmi values.
baby_data$bmi
```

```
##  [1]   NA 22.7   NA   NA 20.4   NA 22.5 30.1 22.0   NA   NA   NA 21.7 28.7
## [15] 29.6   NA 27.2 26.3 35.3 25.5   NA 33.2 27.5 24.9 27.4
```

Our summary function told us that there are 9 NA's, reflected by subsetting the bmi column

```
#Step 1: Point R to the NA values in the bmi column, and replace it with mean values
baby_data$bmi[which(is.na(baby_data$bmi))]= mean(baby_data$bmi, na.rm =TRUE)
# the na.rm = TRUE statement removes the NA values from the data set when calculating the mean.
#We must do this, otherwise the mean runs as NA.

baby_data$bmi
```

```
##  [1] 26.5625 22.7000 26.5625 26.5625 20.4000 26.5625 22.5000 30.1000
##  [9] 22.0000 26.5625 26.5625 26.5625 21.7000 28.7000 29.6000 26.5625
## [17] 27.2000 26.3000 35.3000 25.5000 26.5625 33.2000 27.5000 24.9000
## [25] 27.4000
```

```
#now, the missing values have been replaced with the mean (26.5625)
```

The same logic follows for the cholesterol function (chl)

```
baby_data$chl[which(is.na(baby_data$chl))]= mean(baby_data$chl, na.rm =TRUE)
baby_data$chl
```

```
##  [1] 191.4 187.0 187.0 191.4 113.0 184.0 118.0 187.0 238.0 191.4 191.4
## [12] 191.4 206.0 204.0 191.4 191.4 284.0 199.0 218.0 191.4 191.4 229.0
## [23] 131.0 191.4 186.0
```

Now that we have done the simple mean imputation, lets see how the mice package can make our lives easier.

Mice Imputation

```
# Mice has a lot of different methods you can use to impute your data. For categorical variables (such

#All the methods of mice can be found using the code: methods(mice)

mdata <- nhanes
mdata$hyp <- as.factor(mdata$hyp)
# we must transform this column to a factor as mentioned before.
baby_imp <- mice(mdata,m=5,method=c("","pmm","logreg","pmm"),maxit=20)
```

```
##
##  iter imp variable
##   1   1  bmi  hyp  chl
##   1   2  bmi  hyp  chl
##   1   3  bmi  hyp  chl
##   1   4  bmi  hyp  chl
##   1   5  bmi  hyp  chl
##   2   1  bmi  hyp  chl
##   2   2  bmi  hyp  chl
##   2   3  bmi  hyp  chl
##   2   4  bmi  hyp  chl
##   2   5  bmi  hyp  chl
##   3   1  bmi  hyp  chl
##   3   2  bmi  hyp  chl
##   3   3  bmi  hyp  chl
##   3   4  bmi  hyp  chl
##   3   5  bmi  hyp  chl
##   4   1  bmi  hyp  chl
##   4   2  bmi  hyp  chl
##   4   3  bmi  hyp  chl
##   4   4  bmi  hyp  chl
##   4   5  bmi  hyp  chl
##   5   1  bmi  hyp  chl
##   5   2  bmi  hyp  chl
##   5   3  bmi  hyp  chl
##   5   4  bmi  hyp  chl
##   5   5  bmi  hyp  chl
##   6   1  bmi  hyp  chl
##   6   2  bmi  hyp  chl
##   6   3  bmi  hyp  chl
```

```
##   6   4  bmi  hyp  chl
##   6   5  bmi  hyp  chl
##   7   1  bmi  hyp  chl
##   7   2  bmi  hyp  chl
##   7   3  bmi  hyp  chl
##   7   4  bmi  hyp  chl
##   7   5  bmi  hyp  chl
##   8   1  bmi  hyp  chl
##   8   2  bmi  hyp  chl
##   8   3  bmi  hyp  chl
##   8   4  bmi  hyp  chl
##   8   5  bmi  hyp  chl
##   9   1  bmi  hyp  chl
##   9   2  bmi  hyp  chl
##   9   3  bmi  hyp  chl
##   9   4  bmi  hyp  chl
##   9   5  bmi  hyp  chl
##   10   1  bmi  hyp  chl
##   10   2  bmi  hyp  chl
##   10   3  bmi  hyp  chl
##   10   4  bmi  hyp  chl
##   10   5  bmi  hyp  chl
##   11   1  bmi  hyp  chl
##   11   2  bmi  hyp  chl
##   11   3  bmi  hyp  chl
##   11   4  bmi  hyp  chl
##   11   5  bmi  hyp  chl
##   12   1  bmi  hyp  chl
##   12   2  bmi  hyp  chl
##   12   3  bmi  hyp  chl
##   12   4  bmi  hyp  chl
##   12   5  bmi  hyp  chl
##   13   1  bmi  hyp  chl
##   13   2  bmi  hyp  chl
##   13   3  bmi  hyp  chl
##   13   4  bmi  hyp  chl
##   13   5  bmi  hyp  chl
##   14   1  bmi  hyp  chl
##   14   2  bmi  hyp  chl
##   14   3  bmi  hyp  chl
##   14   4  bmi  hyp  chl
##   14   5  bmi  hyp  chl
##   15   1  bmi  hyp  chl
##   15   2  bmi  hyp  chl
##   15   3  bmi  hyp  chl
##   15   4  bmi  hyp  chl
##   15   5  bmi  hyp  chl
##   16   1  bmi  hyp  chl
##   16   2  bmi  hyp  chl
##   16   3  bmi  hyp  chl
##   16   4  bmi  hyp  chl
##   16   5  bmi  hyp  chl
##   17   1  bmi  hyp  chl
##   17   2  bmi  hyp  chl
```

```
##   17   3  bmi  hyp  chl
##   17   4  bmi  hyp  chl
##   17   5  bmi  hyp  chl
##   18   1  bmi  hyp  chl
##   18   2  bmi  hyp  chl
##   18   3  bmi  hyp  chl
##   18   4  bmi  hyp  chl
##   18   5  bmi  hyp  chl
##   19   1  bmi  hyp  chl
##   19   2  bmi  hyp  chl
##   19   3  bmi  hyp  chl
##   19   4  bmi  hyp  chl
##   19   5  bmi  hyp  chl
##   20   1  bmi  hyp  chl
##   20   2  bmi  hyp  chl
##   20   3  bmi  hyp  chl
##   20   4  bmi  hyp  chl
##   20   5  bmi  hyp  chl
```

```
#Lets look at bmi. From our summary at the beginning, our mean is 26.56. For the 5 imputated columns we
baby_imp$imp$bmi
```

```
##       1    2    3    4    5
## 1  27.4 27.2 27.2 33.2 27.2
## 3  33.2 29.6 22.0 35.3 28.7
## 4  24.9 25.5 30.1 22.5 25.5
## 6  25.5 27.4 21.7 24.9 27.4
## 10 29.6 27.4 20.4 22.7 22.7
## 11 30.1 29.6 22.5 27.2 35.3
## 12 22.7 27.2 22.0 22.5 27.4
## 16 27.2 29.6 24.9 27.2 22.5
## 21 29.6 30.1 30.1 30.1 22.5
```

```
# Column two has the least amount of deviation from the mean. Lets choose that one.
cleaned_data <- complete(baby_imp,2)
cleaned_data
```

```
##    age  bmi hyp chl
## 1    1 27.2   1 187
## 2    2 22.7   1 187
## 3    1 29.6   1 187
## 4    3 25.5   2 186
## 5    1 20.4   1 113
## 6    3 27.4   2 184
## 7    1 22.5   1 118
## 8    1 30.1   1 187
## 9    2 22.0   1 238
## 10   2 27.4   1 184
## 11   1 29.6   1 187
## 12   2 27.2   2 187
## 13   3 21.7   1 206
## 14   2 28.7   2 204
## 15   1 29.6   1 229
```

```
## 16   1 29.6   1 187
## 17   3 27.2   2 284
## 18   2 26.3   2 199
## 19   1 35.3   1 218
## 20   3 25.5   2 218
## 21   1 30.1   1 229
## 22   1 33.2   1 229
## 23   1 27.5   1 131
## 24   3 24.9   1 206
## 25   2 27.4   1 186
```

References https://www.youtube.com/watch?v=sNNoTd7xI-4