Lesson 3. Missing values

Data Analysis in R

Valentina Andrade

Contents

Presentation

Objective Unit 1

The purpose of the development of the guide is to review some basic procedures of data processing with R, which are necessary to then be able to apply the more specific contents of this course.

In this course we will distinguish **three moments** of the work with data: manipulation, analysis and presentation.

- **Manipulation**: corresponds to what is generally known as "cleaning", that is to say, making the necessary modifications to be able to carry out the analyses. These modifications previous to the analysis are necessary since the original data with which you are going to work in general do not come perfectly adapted to the analyses that you want to do. Therefore, in terms of data we also make the distinction between original data and processed data.

We will review this mainly in *Unit 1* in Lesson 1, 2 and 3

- **Analysis**: it is mainly related to descriptive analyses associated to the research questions and also data modelling to test research hypotheses.

We will review this mainly in *Unit 2 and 3* in Lesson 4, 5 and 6.

- **Presentation of results**: it is related to how the analyses and results will be shown in articles, conferences or even for the same work as a researcher.

This will be revieW *visualization of data* in *Unit 4* in 7, 8 and 9. However, the visualization of the data will be a content to review in a general way in the lessons that will be dictated from now on.

The manipulation, analysis and presentation of results are recorded in a code document, in this case an R-code (usually a file with the extension .R). The processing code document has 7 parts, plus an initial identification section:

0. Identification and general description: Title, author(s), date, brief information about the content of the document
1. Main libraries (of R) to be used in the analysis
2. Opening database
3. Explore database
4. Selection of variables to be used
5. Manipulation of variables: at this point, for each variable

- General description
- Recoding lost data
- Recoding of values (if necessary)
- Labelling / relabelling (if necessary)

6. Data Manipulation

- Sorting
- Merging
- Aggregating
- Subsetting
- Covert Data
- Tidy Data

7. Missing Values

- Missing data classification
- Imputation with the average
- Imputation by regression
- Imputation by stochastic regression
- LOCF imputation
- Multiple imputation
- Random

Objective Lesson 3

- Missing values: treatment and imputation
- Visualization of Missing values

Attention! The course is not a class on statistics. Accordingly, the lessons review and explain how different imputation and missing data treatment processes are performed in R, that is, they are not lessons on what each of the imputation methods are and when to use them.

If you are unfamiliar with the basic principles and main approaches to missing data statistics, I recommend this recent book:

Little, Roderick JA, and Donald B. Rubin. Statistical analysis with missing data. Vol. 793. John Wiley & Sons, 2019

In Lesson 3 we will use the process data from Lesson 2

0. Identification and general description

Nhanes Survey (National Health and Nutrition Examination Survey) - The National Health and Nutrition Examination Survey (NHANES) is a program of studies designed to assess the health and nutritional status of adults and children in the United States. The survey is unique in that it combines interviews and physical examinations. NHANES is a major program of the National Center for Health Statistics (NCHS). NCHS is part of the Centers for Disease Control and Prevention (CDC) and has the responsibility for producing vital and health statistics for the Nation. The NHANES interview includes demographic, socioeconomic, dietary, and health-related questions. The examination component consists of medical, dental, and physiological measurements, as well as laboratory tests administered by highly trained medical personnel. The diseases, medical conditions, and health indicators to be studied include: Anemia, Cardiovascular disease, Diabetes, Environmental exposures, Eye diseases, Hearing loss, Infectious diseases, Kidney disease, Nutrition, Obesity, Oral health, Osteoporosis, Physical fitness and physical functioning, Reproductive history and sexual behavior, Respiratory disease (asthma, chronic bronchitis, emphysema), Sexually transmitted diseases, Vision. 10000 individuals are surveyed to represent US statistics.

Data manipulation with NHANES

1. Main libraries (of R) to be used in the analysis

The libraries that we are going to use mainly are `tidyr` and `devtools` (tidy/management data), and `MICE`, `VIM` and `lattice` (missing data).

In the case of the libraries, it is recommended to use the sign `#` to comment and make clear the content/function that each one of them will fulfill in our work.

```r
#Install pacman package that allows us to massively load our libraries

if (!require("pacman"))install.packages("pacman")

#Libraries to be used

#Manipulation variable (Lesson 1)
pacman::p_load("dplyr",
               "sjmisc",
               "car",
                "readxl",
                "haven")

#Tidy data
pacman::p_load("tidyverse") #is a collection contains ggplot2, dplyr, tidyr, readr, purrr, tibble, stri

#Missing data
pacman::p_load("VIM",
                "mice",
                "lattice",
                "MASS")
```

2. Opening database

Work space

Before we load our database, we run the following lines:

```
rm(list=ls()) # delete all objects in the workspace
#options(scipen=999) # values without scientific note
```

Data

The databases can be loaded from a local file or online. In this case we will use a local file that comes in .RData format: **nhanes2.RData**.

Working Directory

First we must tell the program where we are working, that is, the **Working Directory**. The easiest way is to use the command CTRL + Shift + H. Then, your documents will be opened from the computer and you will have to enter the folder where the databases are stored.

Another option is

```
setwd("~/3. Docencia/Data Analysis in R - Karolinska Institute/Lesson 3")
```

```
#For the exercise

load("nhanes3.RData")
```

3. Explore database

And we run a basic check on data input: names of variables and size of the base in terms of cases and variables (in this example, 10175 cases and 10 variables).

```
dim(nhanes) # base dimension
```

```
## [1] 10175    10
```

```
class(nhanes) # data type
```

```
## [1] "data.frame"
```

```r
View(nhanes)
```

4. Selection relevant variables

This initial part of the analysis is usually the most tedious and longest, and consists of leaving the data ready for analysis. The usual procedures are selection and renaming of variables, identification of missing cases, recoding and generation of simple indexes.

This step is optional and consists of creating a subset of data to continue with the analyses, instead of the complete database. To do this:

1. **Identify** the name of the variables.

```r
sjmisc::find_var(data = nhanes, "calcium")
```

```
##   col.nr    var.name   var.label
## 1      4 calciumvitd calciumvitd
```

```r
names(nhanes) #variable names (columns)
```

```
##  [1] "id"          "gender"      "dpills"      "calciumvitd" "folic"
##  [6] "dysldl"      "start"       "end"         "treatment"   "HCQ_dosis"
```

```r
summary(nhanes)
```

```
##        id          gender          dpills      calciumvitd
##  Min.   :73557   Man  :5003   Saturday :2228   Min.   :  0.000
##  1st Qu.:76101   Woman:5172   Sunday   :1961   1st Qu.:  1.012
##  Median :78644                Monday   :1460   Median :  2.047
##  Mean   :78644                Wednesday: 887   Mean   :  2.638
##  3rd Qu.:81188                Thursday : 821   3rd Qu.:  3.220
##  Max.   :83731                (Other)  :1426   Max.   :100.000
##                               NA's     :1392   NA's   :1644
##      folic           dysldl              start                 end
##  Min.   :  0.00   Length:10175       Min.   :2017-01-05   Min.   :2020-01-15
##  1st Qu.:  6.90   Class :character   1st Qu.:2022-10-07   1st Qu.:2028-04-26
##  Median : 13.20   Mode  :character   Median :2031-12-30   Median :2034-04-15
##  Mean   : 18.03                      Mean   :2031-03-26   Mean   :2033-08-14
##  3rd Qu.: 23.00                      3rd Qu.:2038-03-12   3rd Qu.:2040-04-02
##  Max.   :282.80                      Max.   :2044-05-24   Max.   :2046-03-22
##  NA's   :1644                        NA's   :1116         NA's   :1456
##   treatment           HCQ_dosis
##  Length:10175       Min.   :200.0
##  Class :character   1st Qu.:200.0
##  Mode  :character   Median :200.0
##                     Mean   :253.5
##                     3rd Qu.:400.0
##                     Max.   :400.0
##                     NA's   :472
```

Table 1: Variable description - NHANES (2013-2014)

Variable Name	Label	Code/Value
id	Respondent sequence number	id
gender	Gender	Male and Female
dpills	Intake day of the week	Monday, Tuesday, (. . .)
rpills	Intake day of the week	MON-TUES, (. . .)
diet	On special diet?	Yes - No
vitd	Vitamin D (D2 + D3) (mcg)	0 to 84.5
calcium	Calcium (mg)	0 to 11164
calciumvitd	vitd/calcium	0 to 100
folic	Folic acid (mcg)/10	0 to 282.8
periods	Had regular periods in past 12 months	Yes- No
dysldl	LDL-cholesterol (mg/dL)	Hypercholesterolemia (. . .)
HQC_dosis	Dosis of HQC	200 or 400
treatment	Main treatment	prednisone, rituximab, azathioprine . . .
start	Start day treatment	date
end	End treatment	date

2. Using the `select` function of `dplyr`, we **select** each of our most important variables.

```
nhanes1 <- nhanes %>% dplyr::select(id, gender, dpills, calciumvitd,  folic, dysldl, start, end, treatme
#Now we have a new data.frame (nhanes1):  6 variables (+ id variable)
nhanes <- nhanes1; remove(nhanes1)
```

5. Manipulation of variables

Introduction

In the case of working with R, the adjustment of the variables goes through attending to the different type of structure, usually numerical (vector) or categorical variable (factor). This definition establishes clear differences; for example, an average cannot be made with a factor, and numerical vectors do not have labels.

To see the most basic procedures, please go to **Lesson 1**

6. Data Manipulation

In lesson 2 we reviewed different packages and functions for manipulating the layout and structure of databases. These include:

An important concept to review was the **tidy data**. This concept of data manipulation allows us to reorient our databases based on the objectives we propose. The functions we reviewed were:

Now, let's review the exercise from last class. You must open your R codes where you performed the procedures. In case you have not done them, please open the R script called **Exercise 2**.

7. Missing data

Introduction

Methods for analyzing missing data require assumptions about the nature of the data and the reasons why it is incomplete. This requires a description of the **lost data patterns**.

Then, the most used methods to treat the lost data are

- Eliminate missing data: work with complete or available cases
- Lost data imputation: mean, regression, LOFC, multiple and random

8.1 Missing Data Patterns

Often the amount of data lost is not so important, but the distribution of the lost data is. This involves describing the possible relationships between the measured variables and the probability of missing data

The most known distribution patterns are: univariate pattern, non-response unit pattern, monotonic pattern, general pattern, planned missing pattern and latent variable pattern.

To address the structure of missing data we will use the `VIM` library. Initially we can explore the missing data:

1. Count the total number of NAs in the database

```
sum(is.na(nhanes))
```

```
## [1] 15161
```

```
#15161 NAs
```

2. Know the number of NAs per column

```
colSums(is.na(nhanes))
```

```
##          id      gender      dpills calciumvitd       folic      dysldl 
##           0           0        1392        1644        1644        7070 
##       start         end   treatment    HCQ_dosis 
##        1116        1456         367         472
```

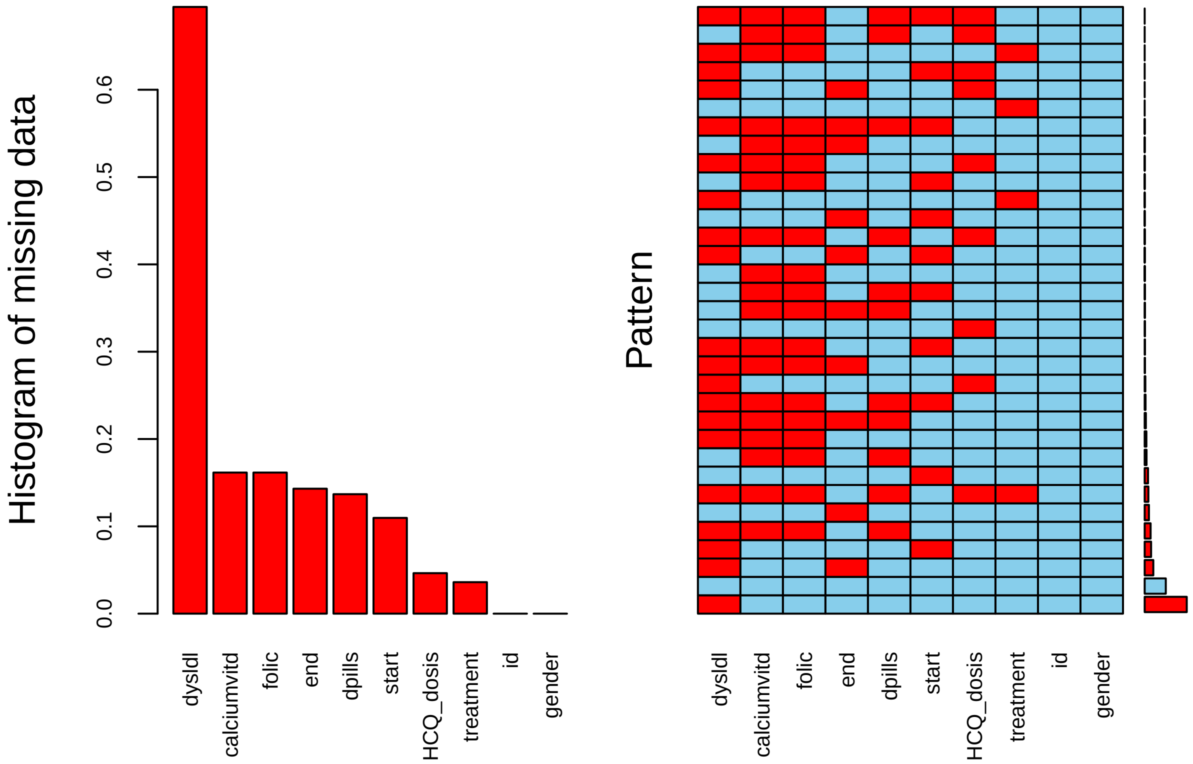
Classification of lost data

- **MCAR (Missing Completely At Random):** The probability that a response to a variable is missing data is independent of both the value of this variable and the value of other variables in the data set.
- **MAR (Missing At Random):** The probability that a response is missing data is independent of the values of the same variable but is dependent on the values of other variables in the data set.
- **NMAR (Not Missing At Random):** The probability that a response to a variable is missing data is dependent on the values of the variable.

To address missing data distribution in R we are going to use `aggr` function from `VIM` library. This plot the amount of missing/imputed values in each variable and the amount of missing/imputed values in certain combinations of variables.

```
aggr_plot <- aggr(nhanes,numbers=TRUE,sortVar=TRUE, labels= names(nhanes), cex.axis=.7, gap=3, ylab=c("
```

```
## Warning in plot.aggr(res, ...): not enough vertical space to display frequencies
## (too many combinations)
```



```
##
##  Variables sorted by number of missings:
##     Variable      Count
##       dysldl 0.69484029
##  calciumvitd 0.16157248
##        folic 0.16157248
##          end 0.14309582
##       dpills 0.13680590
##        start 0.10968059
##    HCQ_dosis 0.04638821
##    treatment 0.03606880
##           id 0.00000000
##       gender 0.00000000
```

The plot helps us understanding that almost 69% of the samples are not missing any information, 4% are missing the HCQ_dosis value, and the remaining ones show other missing patterns.

8.2 Listwise deletion or Complete case analysis

- Omit records or observations
- Problems
  - The models are without ANS and differences in coefficients occur
  - The population cannot be generalized
  - Produces an insufficient amount of records

Excluding Missing Values from Analyses

1. Arithmetic functions on missing values yield missing values.

```
mean(nhanes$calciumvitd) # returns NA
```

```
## [1] NA
```

```
mean(nhanes$calciumvitd, na.rm=TRUE) # returns 2.638
```

```
## [1] 2.638139
```

2. The function **complete.cases()** returns a logical vector indicating which cases are complete. List rows of data that have missing values
3. The function **na.omit()** returns the object with **Listwise deletion of missing values**

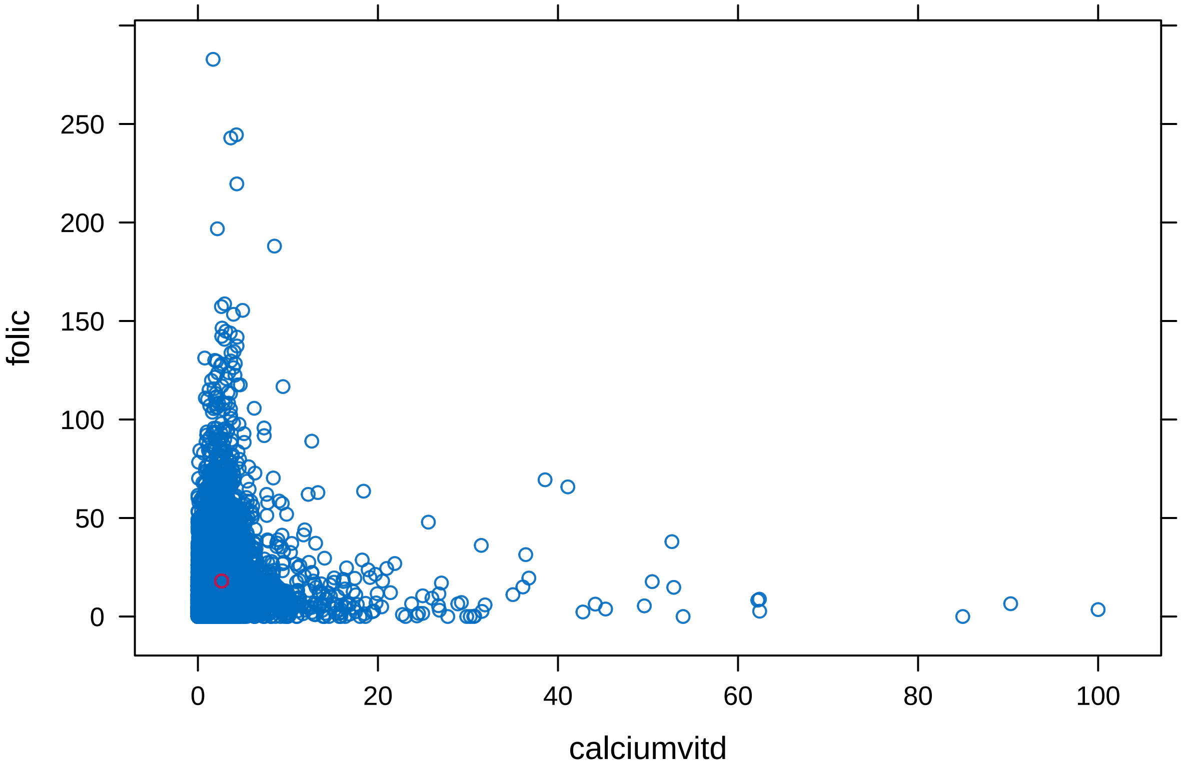
```
nhanes.naomit <- na.omit(nhanes)
#Show rows and columns
dim(nhanes.naomit)
```

```
## [1] 2070   10
```

8.3 Mean imputation

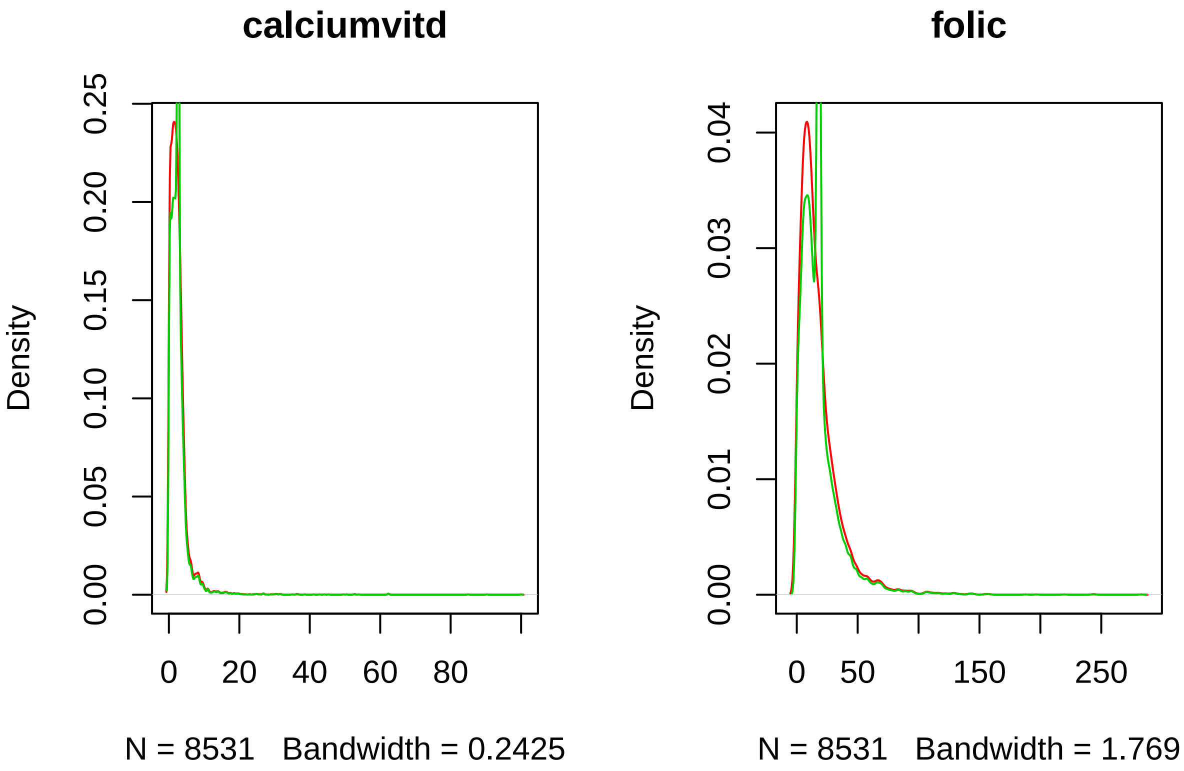
- Imputes missing values by the **average** of the variable in which the values are found.
- We use `mice` library
- Problems
  - Underestimate the variance, alter the relationships between variables, bias almost any estimate other than the mean, and bias the estimate of the mean when the data is not MCAR.

```


#1. Select the variables from which we will get the average
#calciumvitd and dysldl with more NAs
columns <- c("calciumvitd", "folic")

#2. Create imputation method for the database

nhanes_imputedmean <- mice(nhanes[,names(nhanes) %in% columns],m = 1,maxit = 1, method = "mean",seed =

#3. Complete cells by mean

complete.data <- mice::complete(nhanes_imputedmean)

xyplot(nhanes_imputedmean,folic ~calciumvitd)
```



Imputation

```
par(mfrow=c(1,2))

plot(density(nhanes$calciumvitd,na.rm = T),col=2,main="calciumvitd")
lines(density(complete.data$calciumvitd),col=3)

plot(density(nhanes$folic,na.rm = T),col=2,main="folic")
lines(density(complete.data$folic),col=3)
```

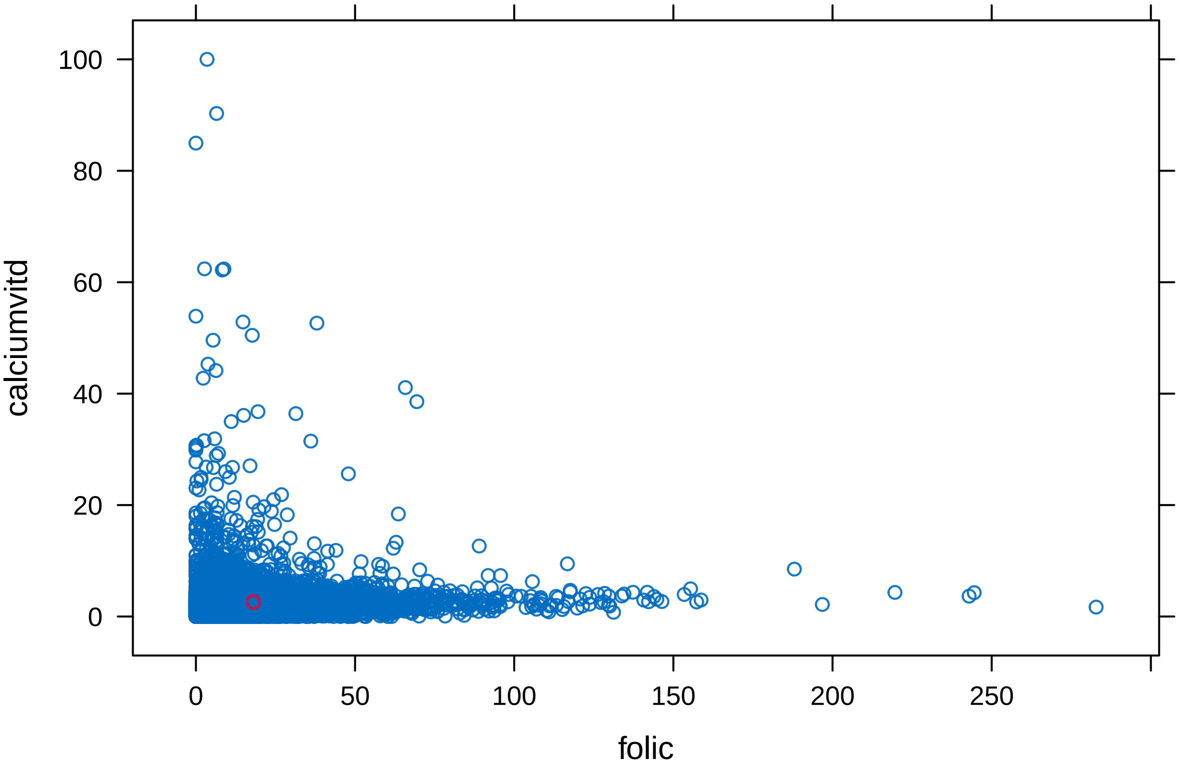


Plot

8.4 Imputation by regression

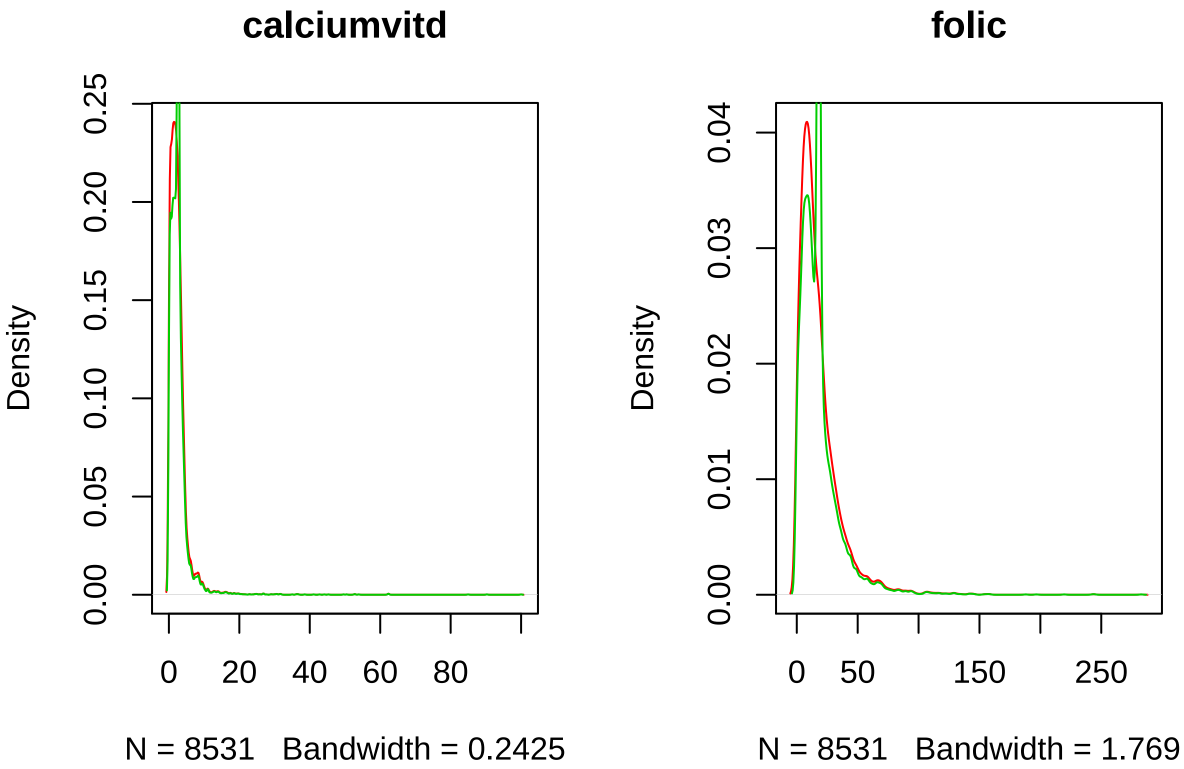
- It incorporates the knowledge of other variables with the idea of producing more sophisticated imputations. The imputed values are the most probable, according to the regression.
- Regression imputation produces unbiased estimates of the means under MCAR, regression weights are unbiased in the MAR if the factors that influence absence are part of the regression model.
- Problems
  - It artificially strengthens the relationships in the data. Regression imputation is a formula for false positive and spurious relationships.
  - Correlations are biased upwards.
  - Variability is underestimated.
  - Imputations are too good to be true.

```
nhanes_impute2  <- mice(nhanes[,names(nhanes) %in% columns],m = 1,
  maxit = 1, method = "norm.predict",seed = 2018,print=F)

nhanes_complete2 <- mice::complete(nhanes_impute2)

xyplot(nhanes_impute2,calciumvitd ~folic)
```



Imputation

```
par(mfrow=c(1,2))

plot(density(nhanes$calciumvitd,na.rm = T),col=2,main="calciumvitd")
lines(density(nhanes_complete2$calciumvitd),col=3)

plot(density(nhanes$folic,na.rm = T),col=2,main="folic")
lines(density(nhanes_complete2$folic),col=3)
```



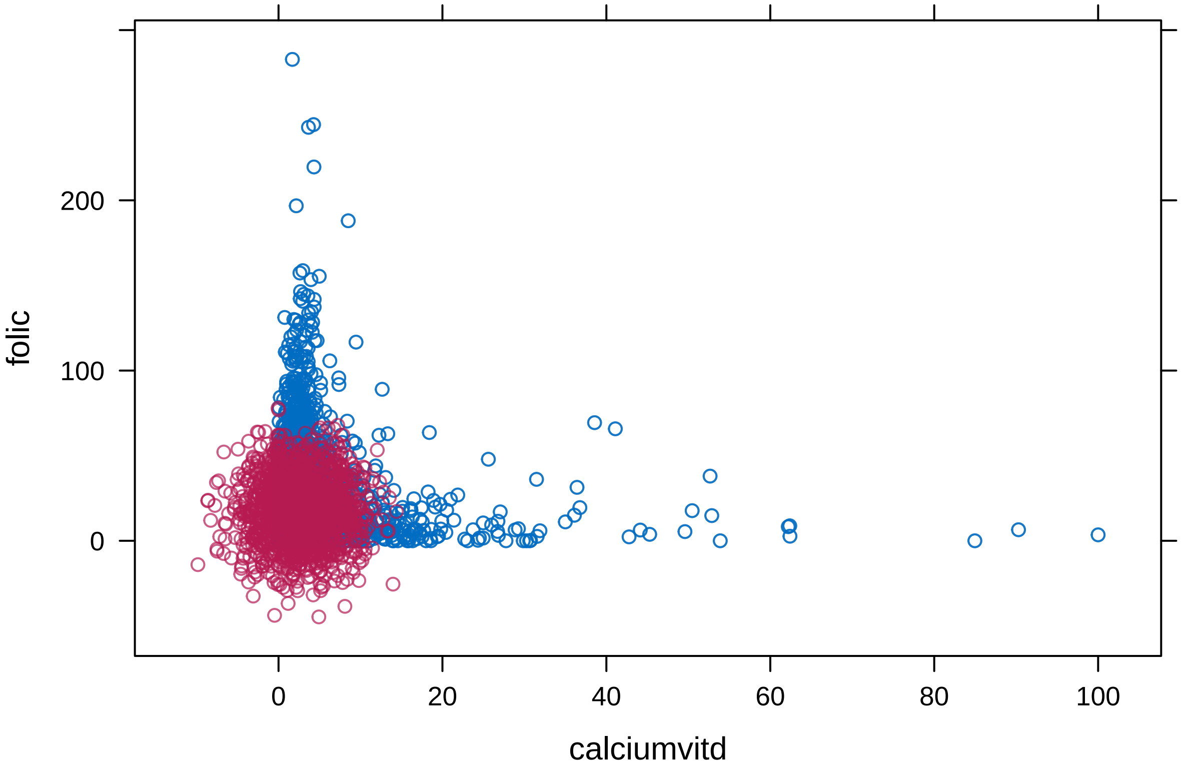
Plot

```
#Similar
```

8.5 Imputation by Stochastic regression

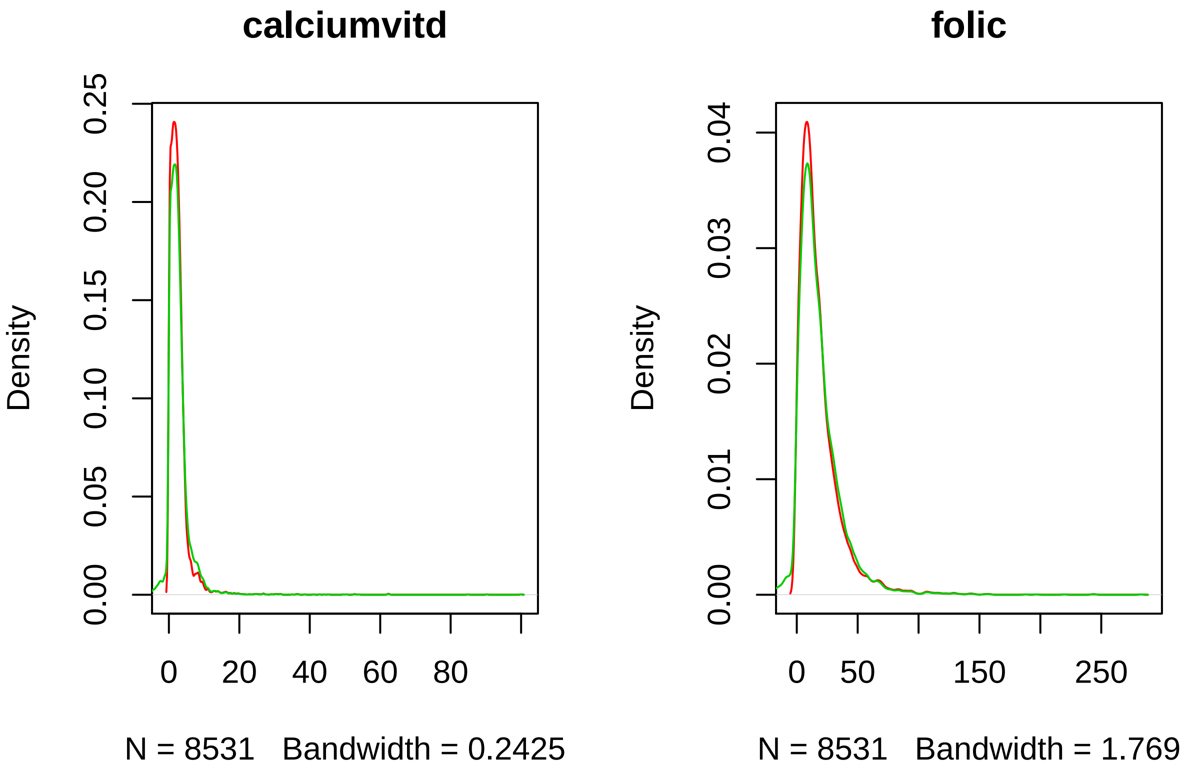
- It's an imputation that adds *noise* to the predictions
- It calculates the intercept, slope and residual variance in the linear model, then calculates the predicted value for each missing value and adds a random draw of the residual to the prediction.
- A well-executed stochastic regression imputation preserves not only the regression weights, but also the correlation between variables.

```
nhanes_impute3 <- mice(nhanes[,names(nhanes) %in% columns],m = 1,
  maxit = 1, method = "norm.nob",seed = 2018,print=F)

nhanes_complete3 <- mice::complete(nhanes_impute3)

xyplot(nhanes_impute3,folic ~ calciumvitd)
```



Imputation

```
par(mfrow=c(1,2))

plot(density(nhanes$calciumvitd,na.rm = T),col=2,main="calciumvitd")
lines(density(nhanes_complete3$calciumvitd),col=3)

plot(density(nhanes$folic,na.rm = T),col=2,main="folic")
lines(density(nhanes_complete3$folic),col=3)
```



Plot

```
#Better
```

8.6 Last Observation Carried Forward

- LOCF is an ad hoc allocation method for longitudinal data.
- The idea is to take the previous observed value as a replacement for the missing data.
- When several values are missing in succession, the method looks for the last observed value.
- To perform this method we will use the package `tidyr`

```
nhanes_impute4 <- tidyr::fill(nhanes, dysldl)

#Longitudinal data
```

- LOCF is convenient because it generates a complete data set. It can be applied with confidence in cases where we are sure what the missing values should be, for example, for administrative variables in longitudinal data.

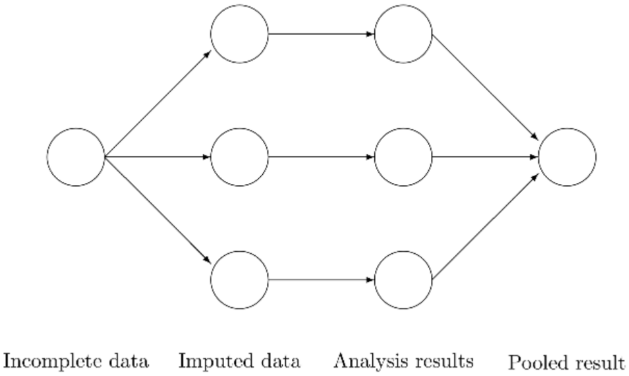
- The method has been used for a long time in clinical trials

8.6 Multiple imputation

- Multiple allocation creates m>*1* complete data sets.

- Based on the "Rubin rules", the m results are grouped into a final point estimate plus a standard error.

- The figure illustrates the three main steps in multiple imputation: imputation, analysis, and clustering.



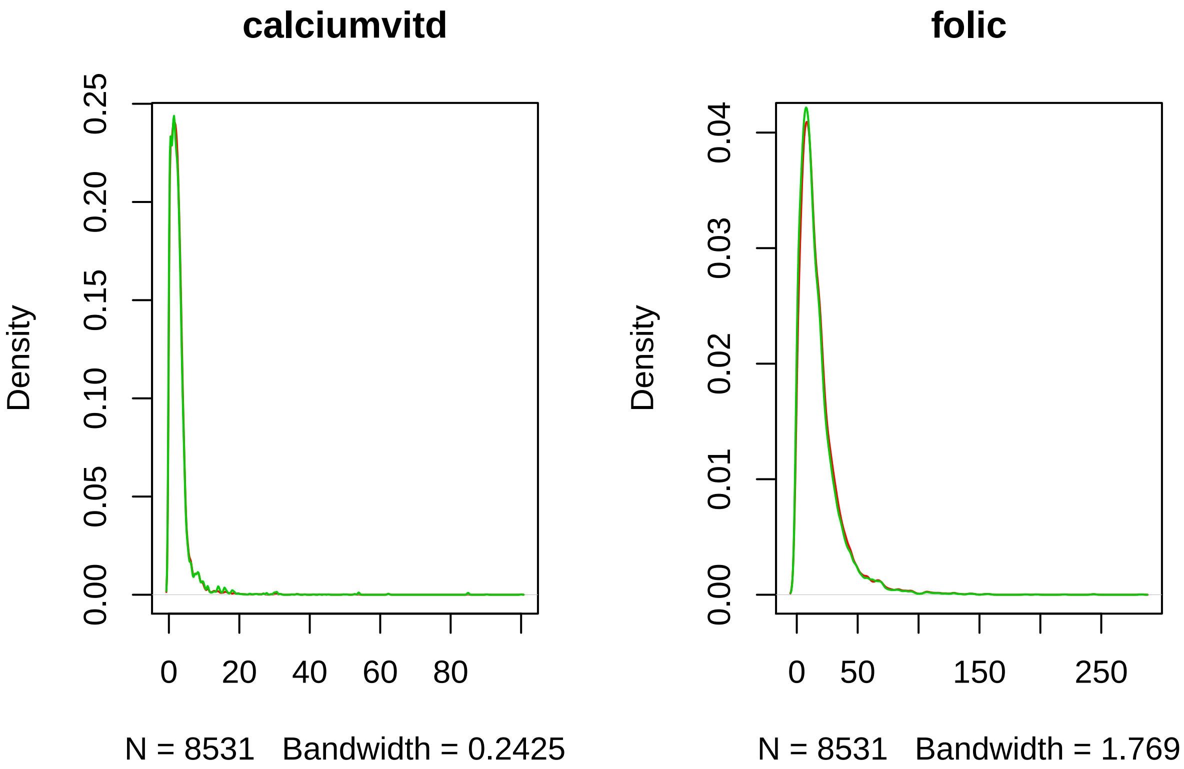
Steps

1. Start with observed data that are incomplete (contain ANS). Multiple imputation creates several complete versions of the data by replacing the missing values with plausible data values. These plausible values are extracted from a distribution modeled specifically for each missing cell.
2. Estimate the parameters of interest for each imputed dataset.
3. Join the m parameter estimates into a single estimate. Then, a total variance is estimated that combines the variance within and between the imputation.

```
nhanes_impute5 <- mice(nhanes[,names(nhanes) %in% columns], seed=2018,print = F, m = 30)

nhanes_complete5<- mice::complete(nhanes_impute5)
```

Imputation

```
par(mfrow=c(1,2))

plot(density(nhanes$calciumvitd,na.rm = T),col=2,main="calciumvitd")
lines(density(nhanes_complete5$calciumvitd),col=3)

plot(density(nhanes$folic,na.rm = T),col=2,main="folic")
lines(density(nhanes_complete5$folic),col=3)
```



Plot

```
#The best!
```

8.7 Random imputation

If we want the imputed data to be defined randomly, we proceed as follows

```
#It's complex (loop)
nhanes_impute6 <-function(x){

missing <- (is.na(x)) #vector booleano

n.missing <- sum(missing)# NA’s number

x.obs <- x[!missing]# Data frame without NA

imputed <- x

imputed[missing] <- sample(x.obs,n.missing,replace = T)

# Extract a random sample and replace NAs

return(imputed)}
```

```r
nhanes_complete6 <- nhanes_impute6(nhanes$calciumvitd)
nhanes_complete7 <- nhanes_impute6(nhanes$folic)
```

Imputation For more advanced imputation methods, I recommend to visit Flexible Imputation of Missing Data

9. Processed database generation for analysis

Nhanes with NAs

```r
names(nhanes) # Variable name
```

```
##  [1] "id"          "gender"      "dpills"      "calciumvitd" "folic"
##  [6] "dysldl"      "start"       "end"         "treatment"   "HCQ_dosis"
```

```r
dim(nhanes) # check
```

```
## [1] 10175    10
```

Save processed database

```r
save(nhanes,file = "nhanes3.RData") #With NA
```