

Pakiet mice

Martyna Majchrzak, Agata Makarewicz, Jacek Wiśniewski

26 03 2020



Wprowadzenie

- ▶ Multivariate Imputation by Chained Equations
- ▶ wielowymiarowa imputacja za pomocą równań łańcuchowych

Wykorzystanie

Pakiet mice zawiera funkcje służące do:

- ▶ sprawdzenia wzorca brakujących danych (-> md.pattern, ...)
- ▶ imputacji brakujących danych (wielokrotnie) (-> mice)
- ▶ diagnozowania jakości imputowanych wartości (jakie funkcje?)
- ▶ analizy każdego uzupełnionego zbioru danych (?)
- ▶ zebrania wyników powtarzanych analiz (-> pool)
- ▶ przechowywania i eksportowania imputowanych danych w różnych formatach (?)
- ▶ generowania symulowanych niekompletnych danych (-> ampute)
- ▶ (Incorporate custom imputation methods)?

Imputacja danych

Funkcja mice

W zależności od typu brakujących danych, funkcja mice przyjmuje jako parametr inne metody imputacji danych.

Dane podzielone są na 4 kategorie:

- ▶ dane numeryczne (ciągłe)
- ▶ dane binarne (dane typu factor z dwoma poziomami)
- ▶ nieuporządkowane dane katagoryczne (dane typu factor z więcej niż 2 poziomami)
- ▶ uporządkowane dane katagoryczne (dane typu factor z więcej niż 2 poziomami uporządkowanymi)

Zbiory danych

- ▶ boys (wzrost, waga, wiek . . . duńskich chłopców)
- ▶ brandsma (dane o uczniach z różnych szkół)
- ▶ pattern (4 proste zbiory danych z różnymi wzorcami braków danych)

```
# nie mam pojecia co zrobic zeby output sie zmiescil na sl  
# zajmujemy sie boys bo maja ordered/unordered factor - n  
str(boys)
```

```
## 'data.frame':    748 obs. of  9 variables:  
## $ age: num  0.035 0.038 0.057 0.06 0.062 0.068 0.068 0.  
## $ hgt: num  50.1 53.5 50 54.5 57.5 55.5 52.5 53 55.1 54  
## $ wgt: num  3.65 3.37 3.14 4.27 5.03 ...  
## $ bmi: num  14.5 11.8 12.6 14.4 15.2 ...  
## $ hc : num  33.7 35 35.2 36.7 37.3 37 34.9 35.8 36.8 38  
## $ gen: Ord.factor w/ 5 levels "G1"<"G2"<"G3"<...: NA NA  
## $ phb: Ord.factor w/ 6 levels "P1"<"P2"<"P3"<...: NA NA  
## $ tv : int   NA NA NA NA NA NA NA NA NA NA ...  
## $ reg: Factor w/ 5 levels "north","east",...: 4 4 4 4 4
```

```
summary(boys)
```

```
##           age           hgt           wgt  
## Min.      : 0.035    Min.      : 50.00    Min.      :  3.14    Min.  
## 1st Qu.: 1.581    1st Qu.: 84.88    1st Qu.: 11.70    1st  
## Median :10.505    Median :147.30    Median : 34.65    Med
```

Dowolne dane

Niektóre metody imputacji możemy zastosować do każdego typu danych.

- ▶ pmm (predictive mean matching/ predykcyjne dopasowanie średniej)
- ▶ midastouch (weighted predictive mean matching/ ?)
- ▶ sample (losowa próbka)
- ▶ cart (drzewo klasyfikacyjne i regresji (?))
- ▶ rf (random forest/lasy losowe)
- ▶ 2lonly.pmm (Level-2 class predictive mean matching) <- ?

Dane numeryczne

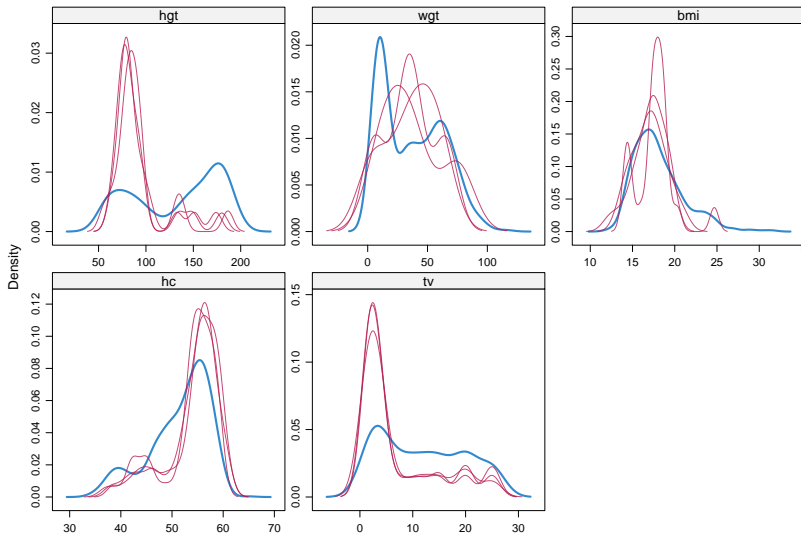
- ▶ pmm (domyślna)
- ▶ mean (średnia)
- ▶ norm (Bayesian linear regression/regresja liniowa)
 - ▶ norm.nob (linear regression ignoring model error)
 - ▶ norm.boot (linear regression using bootstrap)
 - ▶ norm.predict (linear regression, predicted values)
- ▶ quadratic (imputation of quadratic terms)
- ▶ ri (random indicator for nonignorable data) #Nie do końca czaje te wszystkie 2l. coś tam, nw czy je chcemy
- ▶ 2l.norm (Level-1 normal heteroscedastic)
- ▶ 2l.lmer (Level-1 normal homoscedastic, lmer)
- ▶ 2l.pan (Level-1 normal homoscedastic, pan)
- ▶ 2lonly.mean (Level-2 class mean)
- ▶ 2lonly.norm (Level-2 class normal)

```
dutch_boys <- boys
imp <- mice(dutch_boys[, -c(6,7,9)],
           method="pmm", m=3, maxit=3)
```

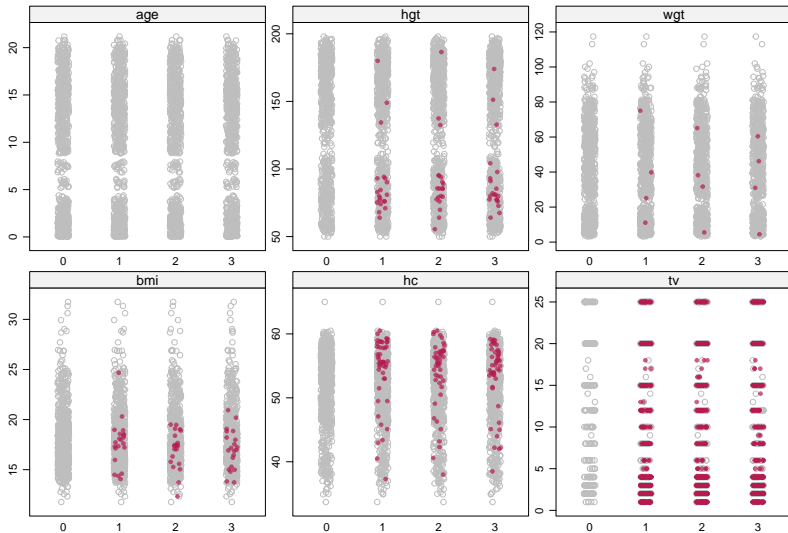
```
##
##  iter imp variable
##    1    1 hgt  wgt  bmi  hc  tv
##    1    2 hgt  wgt  bmi  hc  tv
##    1    3 hgt  wgt  bmi  hc  tv
##    2    1 hgt  wgt  bmi  hc  tv
##    2    2 hgt  wgt  bmi  hc  tv
##    2    3 hgt  wgt  bmi  hc  tv
##    3    1 hgt  wgt  bmi  hc  tv
##    3    2 hgt  wgt  bmi  hc  tv
##    3    3 hgt  wgt  bmi  hc  tv
```

```
dutch_boys[, -c(6,7,9)] <- complete(imp)
```

densityplot(imp)



```
stripplot(imp,col=c("grey",mdc(2)),pch=c(1,20))
```



Dane binarne

- ▶ logreg (logistic regression/regresja logistyczna) (domyślna)
- ▶ logreg.boot (logistic regression with bootstrap)
- ▶ 2l.bin (Level-1 logistic, glmer)

Nieuporządkowane dane kategoryczne

- ▶ polyreg (Polytomous logistic regression) (domyślna)
- ▶ lda (liniowa analiza dyskryminacyjna)

```
imp <- mice(dutch_boys[, -9], method="lda", m=3, maxit=3)
```

```
##
```

```
##   iter imp variable
```

```
##    1    1  gen   phb
```

```
##    1    2  gen   phb
```

```
##    1    3  gen   phb
```

```
##    2    1  gen   phb
```

```
##    2    2  gen   phb
```

```
##    2    3  gen   phb
```

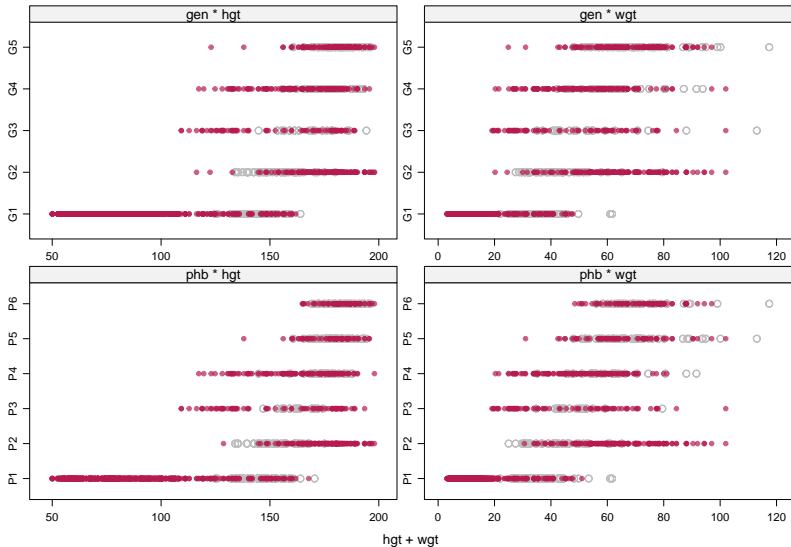
```
##    3    1  gen   phb
```

```
##    3    2  gen   phb
```

```
##    3    3  gen   phb
```

```
dutch_boys[, -9] <- complete(imp)
```

```
xyplot(imp, gen+phb ~ hgt+wgt,
       cex=1,col=c("grey",mdc(2)),pch=c(1,20))
```



Uporządkowane dane kategoryczne

- ▶ polr (Proportional odds model) (domyślna)

```
imp <- mice(dutch_boys, method="polr", m=3, maxit=3)
```

```
##
```

```
##   iter imp variable
```

```
##    1    1   reg
```

```
##    1    2   reg
```

```
##    1    3   reg
```

```
##    2    1   reg
```

```
##    2    2   reg
```

```
##    2    3   reg
```

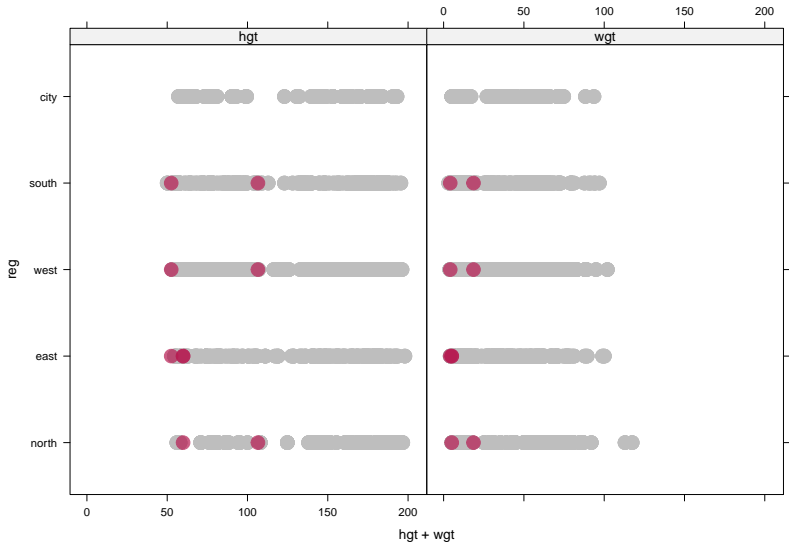
```
##    3    1   reg
```

```
##    3    2   reg
```

```
##    3    3   reg
```

```
dutch_boys <- complete(imp)
```

```
xyplot(imp, reg ~ hgt + wgt,  
       cex=3, col=c("grey", mdc(2)), pch=20)
```



Wykresy

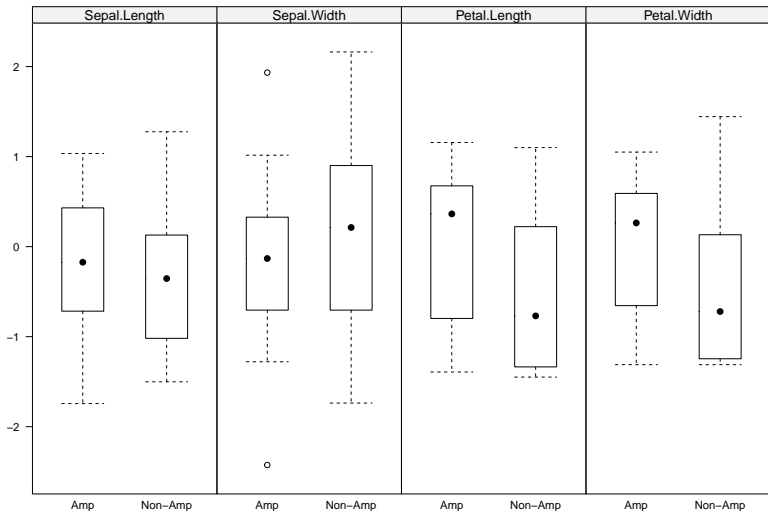
- ▶ `xyplot # <- done`
- ▶ `md.pattern # <- done`
- ▶ `fluxplot #` tak jak `md.pattern` - do brakujących - można na początku boys dac
- ▶ `densityplot # <- done`
- ▶ `stripplot`

Generowanie braków danych

- ▶ `ampute`
- ▶ generowanie brakujących danych potrzebnych do symulacji
 - ▶ określony procent danych zostaje zastąpiony NA (obserwacje są wybierane losowo)
 - ▶ różne mechanizmy: MAR, MCAR, MNAR
 - ▶ określenie wzorca braków danych oraz częstotliwości jego wystąpienia
- ▶ `bwplot`

```
set.seed(1)
```

```
iris_amp <- ampute(iris[, -5], prop = 0.5, mech = "MCAR")  
mice::bwplot(iris_amp, which.pat = 1)
```



Data distributions in pattern 1

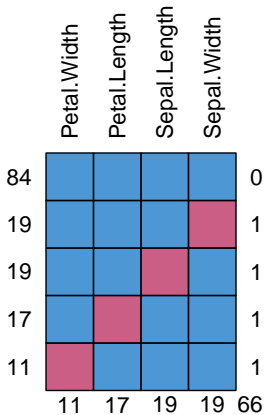
Sprawdzenie wzorca brakujących danych

- ▶ `md.pattern`
- ▶ wyświetlenie wzorca brakujących danych w formie wykresu (oraz tabeli - w konsoli)

```
md.pattern(iris_amp$amp, plot = TRUE, rotate.names = TRUE)
```

##	Petal.Width	Petal.Length	Sepal.Length	Sepal.Width		
## 84	1	1	1	1	0	
## 19	1	1	1	0	1	
## 19	1	1	0	1	1	
## 17	1	0	1	1	1	
## 11	0	1	1	1	1	
##	11	17	19	19	66	


```
md.pattern(iris_&$&, plot = TRUE, rotate.names = TRUE)
```



Zebranie wyników analiz

Funkcja pool