

Multipel imputation och mice

Lars Lindhagen
tipsR, 2014-05-16

Inledning

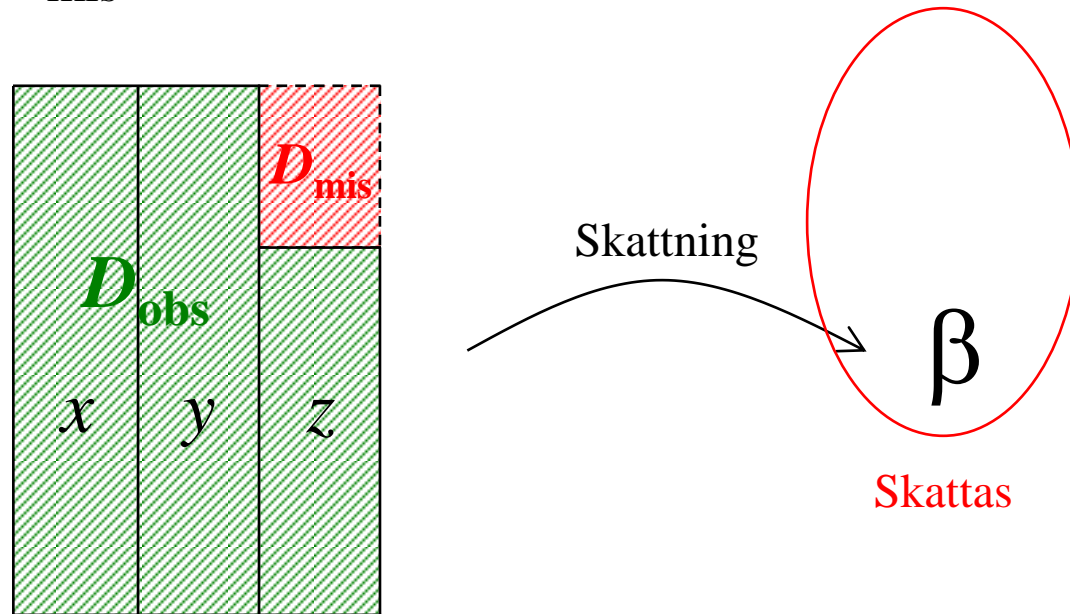
- Lite teori:
 - Multipel imputation
 - MICE
- R-paketet `mice`

Teori

- Multipel imputation hittades på av Rubin på 80-talet.
- Klassisk referens: *Multiple Imputation for Nonresponse in Surveys* (Wiley 1987).
- Bayesiansk trestegsraket:
 1. Imputation
 2. Analys
 3. Poolning

Observerat och saknat data

- Datasetet består av observerat data D_{obs} och saknat data D_{mis}



- Vill basera inferensen på hela D_{obs} men inte på något annat.

Satsen om total sannolikhet

- Posterior för β :

$$p(\beta|D_{\text{obs}}) = \sum_{D_{\text{mis}}} p(\beta|D_{\text{obs}}, D_{\text{mis}}) \times p(D_{\text{mis}}|D_{\text{obs}})$$

- Summan kan approximeras genom att sampla från posteriorn för D_{mis} : $p(D_{\text{mis}}|D_{\text{obs}})$ (Monte Carlo-integration).

Steg 1: Imputation

- Sampla på något sätt m enheter från denna fördelning: $D_{\text{mis}}^{(1)}, \dots, D_{\text{mis}}^{(m)}$

Imputerat dataset nr j
(ingen missing)

$$p(\beta | D_{\text{obs}}) \approx \frac{1}{m} \sum_{j=1}^m p\left(\beta \left| D_{\text{obs}}, \overbrace{D_{\text{mis}}^{(j)}}^{\text{Imputerat dataset nr } j \text{ (ingen missing)}}\right.\right)$$

- Rubin rör sig ganska fritt mellan de Bayesianska och frekventistiska världarna.
- Allt antas normalfördelat med $\sigma = \text{standardfel}$.


Steg 2: Analys

- De m imputerade dataseten analyseras frekventistiskt var för sig (t.ex. logistisk regression).
- Ger skattningar $\hat{\beta}_j$ med standardfel σ_j .

Steg 3: Poolning

- Skattningarna poolas till sist enligt Rubins regler ("Allans formler"):

$$\hat{\beta} = \frac{1}{m} \sum_{j=1}^m \hat{\beta}_j$$

$\text{Var}(\beta)$ 

$$\sigma^2 = \underbrace{\frac{1}{m} \sum_{j=1}^m \sigma_j^2}_{\text{E(Var}(\beta|j))} + \frac{m+1}{m} \times \underbrace{\frac{1}{m-1} \sum_{j=1}^m (\hat{\beta}_j - \hat{\beta})^2}_{\text{Var(E}(\beta|j))}$$

1. Imputera

Ursprungligt dataset

Rökare	Hjärtsvikt	Död
Ja	Nej	Ja
.	Nej	Nej
Ja	Nej	Ja
Nej	Ja	Nej
Nej	Ja	Ja
.	Nej	Ja

Rökare	Hjärtsvikt	Död
Ja	Nej	Ja
Ja	Nej	Nej
Ja	Nej	Ja
Nej	Ja	Nej
Nej	Ja	Ja
Nej	Nej	Ja

Rökare	Hjärtsvikt	Död
Ja	Nej	Ja
Nej	Nej	Nej
Ja	Nej	Ja
Nej	Ja	Nej
Nej	Ja	Ja
Nej	Nej	Ja

Rökare	Hjärtsvikt	Död
Ja	Nej	Ja
Nej	Nej	Nej
Ja	Nej	Ja
Nej	Ja	Nej
Nej	Ja	Ja
Ja	Nej	Ja

2. Analysera

Variabel	OR	95% CI
Hjärtsvikt	4.78	4.13 – 5.54
Rökning	1.57	1.35 – 1.83

Variabel	OR	95% CI
Hjärtsvikt	4.83	4.18 – 5.60
Rökning	1.58	1.35 – 1.84

Variabel	OR	95% CI
Hjärtsvikt	4.81	4.15 – 5.57
Rökning	1.59	1.36 – 1.85

3. Poola

Variabel	OR	95% CI
Hjärtsvikt	4.81	4.11 – 5.61
Rökning	1.58	1.32 – 1.87

Den sista pusselbiten

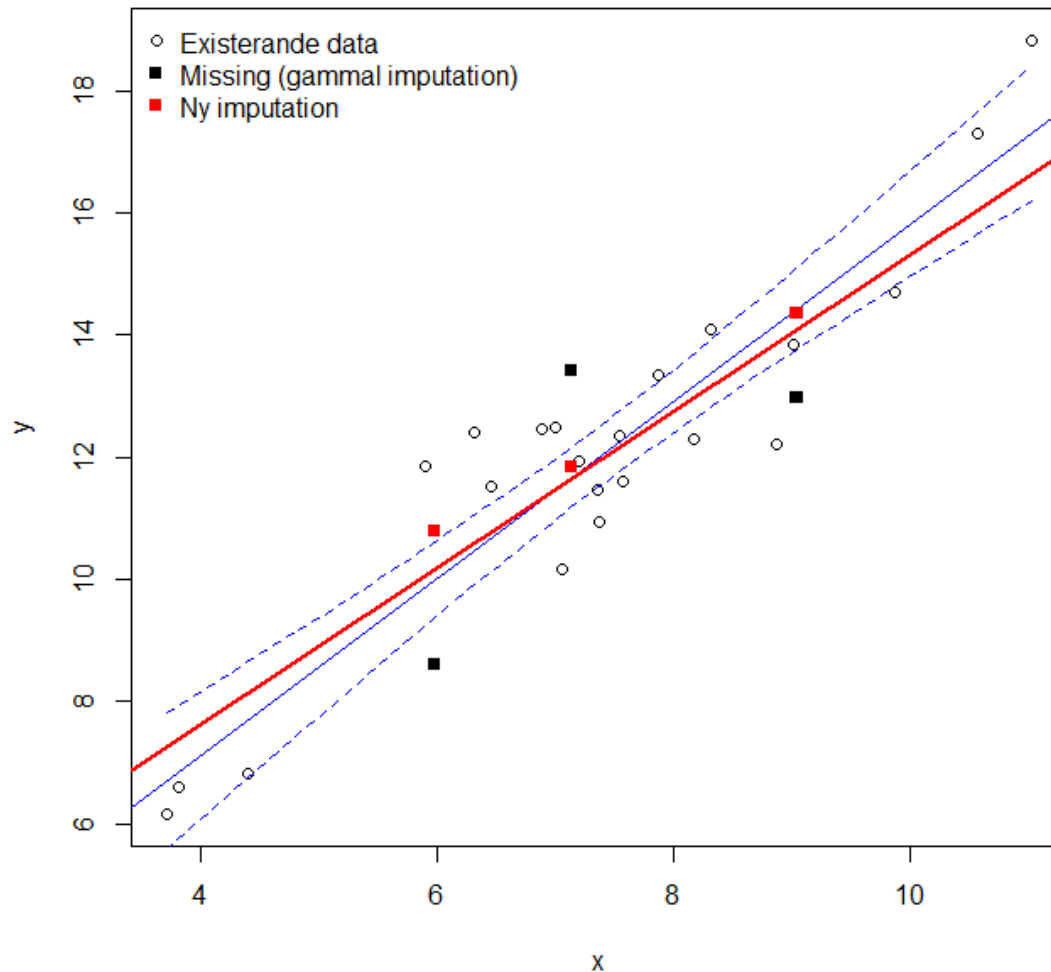
- Då återstår ”bara” problemet att sampla D_{mis} .
- MICE (**M**ultiple **I**mputation using **C**hained **E**quations) är en teknik för att göra detta.

MICE

- MICE bygger på en prediktionsmodell för varje variabel som ska imputeras.
- Från denna modell slumpar man:
 1. Koefficienter
 2. Imputerade värden
- Gibbs-sampler: Loopa igenom variablerna, imputera en i taget baserat på redan imputerade prediktorer.

Iterativ imputation

Imputera y baserat på x



— Regressionslinje
(baserad på
observerade y)

— Slumpad linje

■ Slumpade punkter
runt linjen

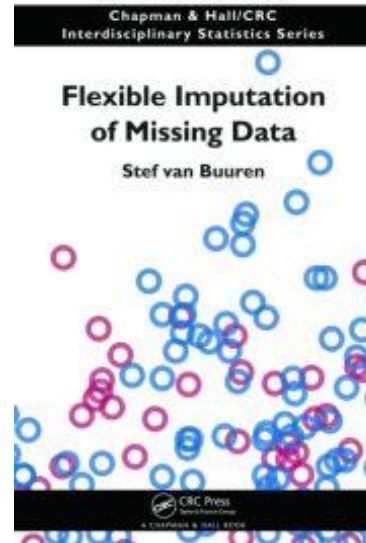
MICE och prediktionsmodeller

- Måste specificera prediktionsmodeller.
- Ta åtminstone med alla variabler man senare tänker använda i analysen, inklusive y .
- Får gärna stoppa in annat av prediktivt värde.
- Kan förstås använda splines, interaktioner etc.

R-paketet mice



Stef van Buuren



Hans bok

- Innehåller funktioner för steg 1 – 3 (imputation, analys och pooling).
- Visst stöd för deskription och diagnostik.

Lätt att använda

```
> head(dd)
  hfail death smoke
1 FALSE  TRUE  TRUE
2 FALSE FALSE   NA
3 FALSE  TRUE  TRUE
4  TRUE FALSE FALSE
5  TRUE  TRUE FALSE
6 FALSE  TRUE   NA
```

Dataset med
missing (NA)

```
> imp <- mice(dd, m=3)
```

Steg 1: Imputation

```
> an <- with(imp, glm(death ~ hfail + smoke, family=binomial))
```

Steg 2: Analys

```
> res <- pool(an)
```

Steg 3: Poolning

```
> summary(res)
```

	est	se	t	df	Pr(> t)	lo 95	hi 95
(Intercept)	-1.0058233	0.03837103	-26.213090	4882.140	0.00000e+00	-1.0810478	-0.9305989
hfailTRUE	1.5703386	0.07496107	20.948721	4526.111	0.00000e+00	1.4233783	1.7172989
smoke	0.4558637	0.07833817	5.819177	4720.985	6.30445e-09	0.3022843	0.6094430

Vi har ett resultat!

Komplikationer

- `mice` antar att alla variabler ska imputeras med alla andra variabler som linjära prediktorer.

Man får alltså göra kodning för splines och interaktioner själv!

- Vill mixtra med imputerat data före analys?
- Analysen misslyckas för några imputationer?
- Antaganden om normalfördelning bryter samman om t.ex. alla exponerade dör ($\beta = \infty$). LRT-CI är då bättre än Wald. Hur poola??
- Hanterar ej multicore.

Tänkbara lösningar

- Skrev egen analysfunktion som inte tar en formel, utan en funktion, där man kan göra vad man vill före/efter analys.
- Divergerande modeller \Rightarrow knök och bök, många if-satser.
- Fattigmans multicore: Starta 4 R-instanser och låt dem göra några imputationer var.

Val av prediktionsmodeller

- `mice` väljer modell utifrån den variabel som ska imputeras. Default:
- **Numeriska:** Predictive mean matching (PMM): Kopiera värde från individ med liknande prediktion.
- **Dikotoma:** Logistisk regression.
- **Ordinal/nominal:** Ordinal/nominal-regression.
- Finns en uppsjö andra modeller att välja bland.

Splines/interaktioner

- Om man vill bygga in splines eller interaktioner i sina prediktionsmodeller, får man skapa specialvariablerna själv.
- Även dessa kan behöva imputeras. Hur?

Splines/interaktioner

- Två strategier:
 1. **”Just another variable” (JAV)**. Strunta i funktionella samband. Imputera specialvariabler som vilka variabler som helst.

Bättre än det låter
 2. **Passivt**. Imputera bara grundvariabeln. Räkna sen ut specialvariabeln på vanligt sätt.

Kräver lite jobb

Passiv imputation

- Exempel: Spline-prediktor med 3 knutar.
Utöver x behövs x_2 .
1. Skriv en funktion f som räknar ut x_2 utifrån x .
 2. Skapa x_2 i datasetet via f .
 3. Imputera x_2 passivt: `method` sätts till " $\sim f(x)$ ".
 4. x_2 får inte predicera x : `predictorMatrix`.
 5. x_2 bör imputeras (= räknas ut) direkt efter x :
`visitSequence`.

Spline för hjärtfrekvens (HF)

```
> f <- function(hr) rcspline.eval(hr, knots = c(56, 75, 113), inclx = F)
```

```
> dd$hr2 <- f(dd$hr) 2. Skapa spline-variabel hr2
```

1. Definiera f



```
> head(dd)
```

	hr	smoke	hr2
1	72	Nej	1.260696
2	68	<NA>	0.531856
3	99	Nej	18.088950
4	68	Nej	0.531856
5	122	<NA>	40.666667
6	NA	Nej	NA

Ursprungligt data

```
> ini <- mice(dd, m=0, maxit=0)
```

```
> meth <- ini$method
```

```
> pred <- ini$predictorMatrix
```

```
> vis <- ini$visitSequence
```

mice-default

```
> meth
```

	hr	smoke	hr2
"pmm"	"logreg"		"pmm"

PMM, logistisk regression

```
> pred
```

	hr	smoke	hr2
hr	0	1	1
smoke	1	0	1
hr2	1	1	0

Alla predicerar alla

```
> vis
```

	hr	smoke	hr2
1	2	3	

In order of appearance

Fortsättning

```
> meth["hr2"] <- "~f(hr) "  
> meth  
      hr      smoke      hr2  
"pmm" "logreg" "~f(hr) "
```

3. Imputera hr2 passivt: $hr2 = f(hr)$

```
> pred["hr", "hr2"] <- 0  
> pred
```

```
      hr smoke hr2  
hr      0      1  0  
smoke   1      0  1  
hr2     1      1  0
```

4. hr2 får inte predicera hr

```
> vis <- c(1, 3, 2)  
> vis  
[1] 1 3 2
```

5. Uppdatera hr2 direkt efter hr

```
> imp <- mice(dd, m=5, meth=meth, pred=pred, visitSequence=vis)
```

Imputera!

```
> head(complete(imp, 4))
```

Det 4:e imputerade datasetet

```
      hr smoke      hr2  
1    72   Nej  1.260696  
2    68   Nej  0.531856  
3    99   Nej 18.088950  
4    68   Nej  0.531856  
5   122    Ja 40.666667  
6    68   Nej  0.531856
```

Patient 6 har kopierat
HR från nr 4.

Erfarenheter

- Allt blir lite besvärligare:
 - Ett extra steg i analysen. Imputationen behöver göras om ifall data ändras.
 - Analyser tar längre tid.
 - Trassel när modeller spårar ur.
 - Deskription
- Uppskattas av tidskrifter?