MICE package in R

0. Reference

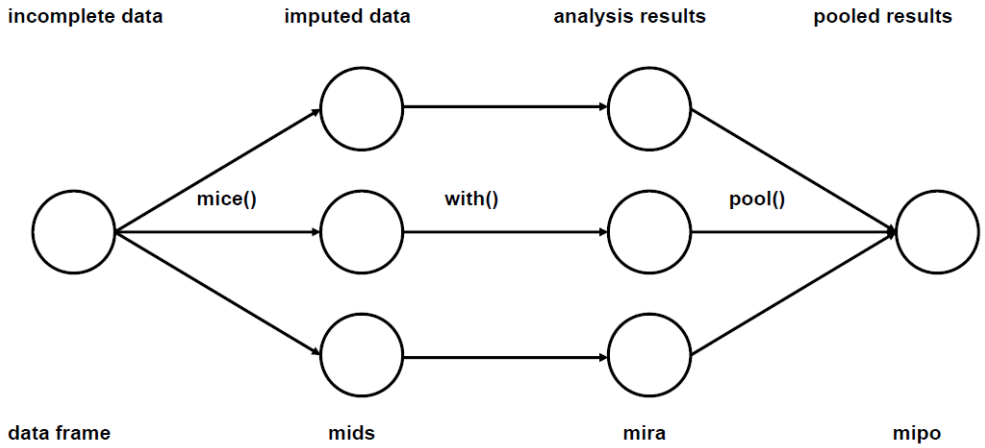
- mice: Multivariate Imputation by Chained Equations in R, Stef van Buuren et al.

1. Intro

MICE는 Rubin의 Multiple Imputation 아이디어를 차용하여 여러 형태의 결측치를 채워 넣는 알고리즘이다. MICE는 파이썬과 R 모두에 구현되어 있는데 여기서는 R package에서 구현된 MICE를 살펴본다. 그 이유는, 파이썬에서 만족할만한 패키지를 찾지 못했기 때문이다. 여기서 만족의 기준은, MICE의 변수 형태별 imputation 방법이 잘 구현되지 않았다는 점이다. MICE는 변수가 연속형일 때, 범주형일 때, 순서형 범주형일 때 등등인 경우에 대해서 각각 다른 imputation 모형을 제시한다. 이러한 점을 파이썬 패키지에서는 클리어하게 발견하지 못했다. 또한 R 패키지를 만든 Stef van Buuren가 어떻게 사용해야 하는지 따로 논문도 발표하여서, R에서는 이를 쉽게 따라할 수 있었다. 이러한 이유로 MICE를 R에서 살펴보았다.

2. Toy Example

본격적으로 mice를 살펴보기 이전에, mice 함수가 어떤 과정을 거치는지 큰 그림을 그려보고 그 과정들을 간단하게 실행해본다.
아래는 MI의 각 과정과 이에 대응하는 mice 함수들을 함께 그려둔 그림이다. 아래 그림은 Rubin이 제시한 MI 과정과 동일하다.



첫 번째 스텝은 data frame에 있는 결측치에 대해서 그럴듯한 값으로 채워 넣는 과정이다. 이는 m번 반복되어 총 m개의 완전한 데이터 세트가 만들어진다. 이를 mice 패키지에서 mids class에 저장된다. 다음 스텝으로는, m개의 완전한 데이터 세트에서 관심있는 quantity, Q를 m개, 즉 $\hat{Q}^{(1)}, \cdots, \hat{Q}^{(m)}$으로 추정하는 것이다. mice 패키지에서는 with.mids()로 이를 제공한다. m개의 만들어진 완전한 데이터가 다를 것이므로 m개의 추정치 또한 다를 것이며 이 다른 정도가 우리가 impute 한 값에 대한

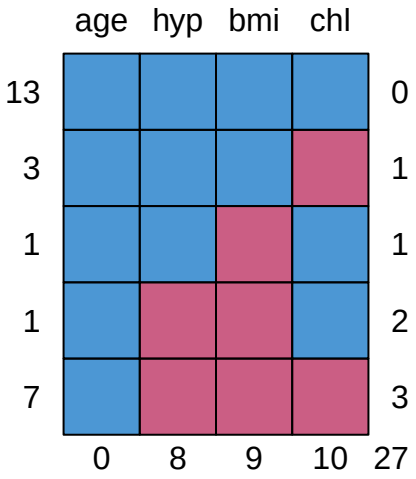
불확실성을 의미한다.
마지막 스텝은 $\hat{Q}^{(1)}, \cdots, \hat{Q}^{(m)}$ 을 하나의 추정치인 $\bar{Q}$로 pooling하는 것이다. 여기서 Rubin의 방법을 이용하여 mean으로 pooling하며 이에 대한 within, between-imputation variance는 pool() 함수를 통해 구할 수 있다. mice 패키지에서는 이를 mipo 클래스에 저장한다.

이제 간단하게, R에서 mice가 작동하는 과정을 살펴보자. 먼저 사용할 데이터는 mice 패키지 안에 있는 nhanes 데이터이다.

```
knitr::opts_chunk$set(comment=NA, fig.width=3, fig.height=3,fig.align='center',message=FALSE)
library(mice)
head(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
```

```
md.pattern(nhanes)
```




```
##    age hyp bmi chl
## 13   1   1   1   1  0
## 3    1   1   1   0  1
```

```
## 1    1   1   0   1  1
## 1    1   0   0   1  2
## 7    1   0   0   0  3
##      0   8   9  10 27
```

md.pattern() 함수를 통해 missing pattern을 살펴보았다. 파랑색과 빨간색은 각각 결측치가 있고 없음을 의미한다. 예를 들어, 결측치가 없는 완전한 데이터는 13개이고 chi 변수에만 결측치가 있는 데이터는 3개이다.
간단하게 mice() 함수를 적용해보자.

```
imp = mice(nhanes, seed=23109)
```

```
 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
```

```
print(imp)
```

```
Class: mids
Number of multiple imputations:  5
Imputation methods:
  age   bmi   hyp   chl
   "" "pmm" "pmm" "pmm"
PredictorMatrix:
    age bmi hyp chl
age   0   1   1   1
bmi   1   0   1   1
hyp   1   1   0   1
chl   1   1   1   0
```

먼저 변수별로 어떤 방법을 사용했는지 나온다. 여기서는 numeric 변수에 대한 기본값인 pmm이 사용되었다. iteration은 기본값인 $m = 5$가 사용되었다.

```
imp$imp$bmi
```

```
      1    2    3    4    5
1  20.4 33.2 27.2 22.0 29.6
3  26.3 29.6 30.1 29.6 29.6
4  24.9 27.4 21.7 22.7 27.2
6  20.4 21.7 24.9 27.4 21.7
10 21.7 20.4 26.3 27.4 26.3
11 26.3 27.5 22.0 25.5 35.3
12 21.7 27.5 22.5 28.7 22.7
16 28.7 35.3 26.3 28.7 30.1
21 29.6 30.1 22.0 33.2 26.3
```

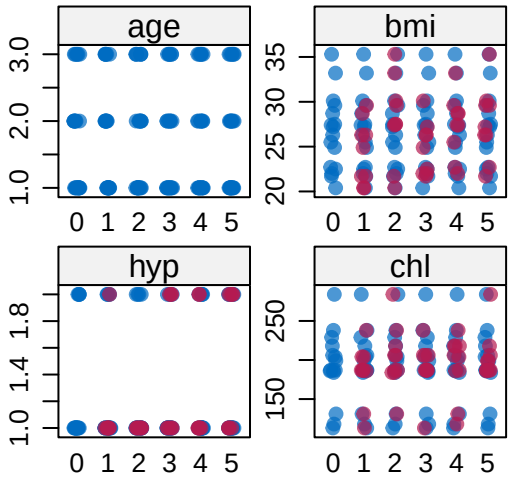
bmi 변수에 대해서, 다섯 번의 imputation 한 값들이다. 왼쪽의 숫자들은 행 인덱스이다. 즉, 첫 행을 해석해보면 첫 번째 데이터가 bmi변수에 대해 결측값을 가지고 있는데, 이에 대해서 5번 imputation을 했다는 뜻이다.

```
head(complete(imp))
```

```
  age  bmi hyp chl
1   1 20.4   1 113
2   2 22.7   1 187
3   1 26.3   1 187
4   3 24.9   2 186
5   1 20.4   1 113
6   3 20.4   1 184
```

총 다섯 번의 iteration 중, 첫 번째 iteration에 대한 complete data set이다.
아래와 같이 이상하게 impute된 값은 없는지, 그림으로 각 변수에 대한 분포를, 관측 데이터와 imputed 데이터로 나누어서 살펴볼 수 있다. 파랑색, 빨간색이 각각 완전, imputed 데이터이다.

```
stripplot(imp, pch = 20, cex = 1.2)
```



예를 들어, 완전한 데이터에서 분석 목적이 regression of chl on age, bmi라고 하자. 이를 위해 with.mids()를 사용할 수 있는데, 이는 각 imputed 데이터 세트에 대해서 해당 모형을 적용하는 함수이다.

```
fit = with(imp, lm(chl ~ age + bmi))
summary(pool(fit))
```

```
             estimate std.error  statistic       df     p.value
```

```
(Intercept) -19.985920 59.797186 -0.3342284 13.67191 0.743278782
age          33.943887  9.810254  3.4600417 11.43139 0.005049049
bmi           5.757804  1.924088  2.9924839 13.44479 0.010064370
```

3. Imputation Models

3.1 Things to Consider

1. MAR 가정을 할 수 있는지 확인해야 한다. mice는 MAR, MNAR 가정을 모두 다룰 수 있고 MNAR인 경우, 추가적인 모델링 가정이 필요하기 때문에 이를 확인해야 한다. 6.2에 어떻게 확인하는지 설명한다.

2. 각 변수별로 imputation model을 지정해야 한다. mice는 연속형, 범주형, 순서형 범주형 변수에 대한 결측치 모형을 따로 설정하기 때문에 이를 명시해야 한다.

3. imputation model에 포함할 predictors을 선택해야 한다. 가능한 관련있는 변수를 많이 넣는 것이 좋은데, 자세한 사항은 2.9에서 살펴본다.

4. 다른 불완전 변수의 함수로 구성된 변수에 대해서 impute를 할지 결정해야 한다. 변수로 평균, 비 등등이 있을 수 있다. 자세한 사항은 2.9에서 살펴본다.

5. imputation 순서이다. 여러 방법이 있고 장/단점이 있다. 이를 3.6에서 살펴본다.

6. starting imputations와 iteration의 횟수이다. 이에 대해 2.9에서 살펴본다.

7. 마지막으로는 imputed data sets 갯수인 m이다. m을 작게 정하는 것은 simulation error을 낳을 수 있고, 특히 fraction of missing information이 클 경우에 주의해야 한다.

3.2 Univariate imputation methods

Method	Description	Scale type	Default
`pmm`	Predictive mean matching	numeric	Y
`norm`	Bayesian linear regression	numeric	
`norm.nob`	Linear regression, non-Bayesian	numeric	
`mean`	Unconditional mean imputation	numeric	
`2L.norm`	Two-level linear model	numeric	
`logreg`	Logistic regression	factor, 2 levels	Y
`polyreg`	Multinomial logit model	factor, >2 levels	Y
`polr`	Ordered logit model	ordered, >2 levels	Y
`lda`	Linear discriminant analysis	factor	
`sample`	Random sample from the observed data	any	

Table 1: Built-in univariate imputation techniques. The techniques are coded as functions named `mice.impute.pmm()`, and so on.

univariate imputation method는 complete predictors을 변수로 받고 타겟 변수의 결측치에 대해서 single imputation을 제공한다. 위 표는 mice에서 제공하는 univariate imputation models이다.
만약에, 아래와 같이 method를 하나로 지정한다면 모든 변수에 대해서 해당 method가 적용된다.
또는 변수의 갯수만큼 방법을 지정하면 각 변수에 대해서 다른 방법이 적용된다.

```
imp <- mice(nhanes, method = "norm")
```

```
 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
```

```
imp <- mice(nhanes, meth = c("", "norm", "pmm", "mean"))
```

```
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
```

위 코드를 보면, 첫 번째 변수의 method는 “”로 지정되었다. mice에서는 이와 같이 지정된 변수에 대해서는 imputation을 생략한다.

Overview of Imputation Methods

- mice.imput.pmm()
  predictive mean matching을 지원한다. semi-parametric imputation method로써, non-linear 관계도 잘 잡아낸다고 한다. 대체적으로 성능이 좋다고 나온다고 하는데, 디폴트로 지정된 이유 같다.
- mice.impute.norm() & mice.impute.norm.nob() & mice.impute.mean()
  linear imputation model을 지원하고 만약 normal 가정이 맞다면 빠르고 효율적이다. nob()은

매우 큰 표본일대 적절하다. mean()은 단순한 mean imputation이고 보통 좋지 않은 전략으로 알려져 있다.

- mice.impute.2L.norm()
  heteroscedastic linear two level model by a gibbs sampler을 사용하여 impute한다. 데이터의 clustering 구조를 반영하면 상당히 좋은 결과를 낼 수 있다.
- 나머지는 위 표를 참고하자.

3.3 Predictor Selection

MICE의 가장 큰 장점 중 하나는 각 불완전 변수에 대한 predictors를 지정할 수 있다는 점이다. predictorMatrix를 통해 변수 갯수 크기의 정사각 행렬이 나오는데, 행 이름의 변수를 impute 하기 위해 어떤 변수들이 사용되는지 0,1 값으로 표시되어 있다.

```
imp = mice(nhanes, print=FALSE)
imp$predictorMatrix
```

```
    age bmi hyp chl
age   0   1   1   1
bmi   1   0   1   1
hyp   1   1   0   1
chl   1   1   1   0
```

예를 들어 위 행렬에서 bmi 변수의 결측치를 impute하기 위해, age, hyp, chl의 변수가 사용되었음을 알 수 있다.

Removing a predictor

아래와 같이 bmi 변수를 각 변수의 predictor에서 제거할 수 있다.

```
pred = imp$predictorMatrix
pred[,'bmi'] = 0
pred
```

```
    age bmi hyp chl
age   0   0   1   1
bmi   1   0   1   1
hyp   1   0   0   1
chl   1   0   1   0
```

```
imp = mice(nhanes, pred=pred, print=FALSE)
```

Multilevel imputation

데이터에 계층 구조가 있을 때 이를 이용하는 imputation 방법이다. random effects와 class variable을 각각 2, -2로 코딩해서 GLMM 모형을 적합한다. 예시는 아래와 같다. popmis 데이터는 popular 변수만 missing 값을 가지며 class 변수로는 school을 사용하였다. 참고로 level은 하나의 변수만 지정할 수 있다.

```
popmis[1:3,]
```

```
  pupil school popular sex texp const teachpop
1     1      1      NA   1   24     1        7
2     2      1      NA   0   24     1        7
3     3      1       7   1   24     1        6
```

```
ini = mice(popmis, maxit=0)
```

```
Warning: Number of logged events: 1
```

```
pred = ini$pred
pred['popular',] = c(0,-2,0,2,1,2,0)
imp = mice(popmis, meth = c('', '', '2l.norm', '', '', '', ''),
                        pred = pred, maxit = 1, seed = 71152)
```

```
 iter imp variable
  1   1  popular
  1   2  popular
  1   3  popular
  1   4  popular
  1   5  popular
```

```
Warning: Number of logged events: 1
```

Advice on predictor selection

imputation을 할 때, 최대한 많은 변수를 사용하는 것이 MAR 가정과 근접해질 수 있으며 bias를 줄이는 방법임이 알려져 있다. 하지만 변수가 매우 많을 경우 이를 모두 포함하는 것이 쉽지 않을 것이다. 저자는 15개에서 25개 정도 변수를 추천하는데, 구체적으로 아래의 전략을 제시한다.

- imputation 이후에 나오는 complete-data 모델을 모두 사용하자.
- 결측치에 영향을 주는 변수를 포함하자. 예를 들어 어떤 변수에 대해서, 결측과 그렇지 않은 그룹으로 나누고 다른 변수에 대해서 t-test를 진행할 수도 있다. 이렇게 함으로써 두 그룹 분포의 차이에 관련성이 있는 변수를 알아낸다. 또는 상관계수를 통해서도 알아낼 수 있다.
- 또한 분산의 상당부분을 설명하는 변수를 포함하자.
- 너무 많은 결측치를 가진 변수는 포함하지 말자.

Quic predictor selections

usable cases의 비율은 타겟 변수와 설명 변수가 얼마나 같이 등장했는지에 대한 지표이다. 둘 모두 동시에 결측됐다면 비율이 낮을 것이고 그 predictor는 타겟 변수를 설명하는데 적은 정보를 가질 것이다. proportion of usable cases는 아래와 같이 계산된다.

```
p = md.pairs(nhanes)
round(p$mr/(p$mr+p$mm),3)
```

```
    age bmi   hyp   chl
age NaN NaN   NaN   NaN
bmi   1 0.0 0.111 0.222
hyp   1 0.0 0.000 0.125
chl   1 0.3 0.300 0.000
```

위 표를 보면 hyp를 impute하기 위해서, 8번 중 1번만 chl이 관측되었다 (타겟 변수는 행 기준이다) 따라서 hyp를 impute하기 위해 사용할 수 있는 chl의 정보는 이 둘의 상관계수가 큼에도 적을 것이라고 예상할 수 있다.
mice에서는 quickpred() 라는 함수를 제공하는데, 상관계수 및 usable cases의 비율을 자동으로 계산해서 디폴트 최소 상관계수인 0.1에 의해서 변수를 선택해주고 최종 predictorMatrix까지 뽑아준다.

```
quickpred(nhanes)
```

```
    age bmi hyp chl
age   0   0   0   0
bmi   1   0   1   1
hyp   1   0   0   1
chl   1   1   1   0
```

이 행렬은 대칭이 아니다. 예를 들어, bmi는 hyp에 대한 predictor가 아니지만 hyp는 bmi에 대한 predictor가 될 수 있다. 이는 hyp와 response indicator of bmi의 correlation이 0.139, 즉 threshold을 넘기 때문이다.

```
round(cor(y = nhanes, x = !is.na(nhanes), use = "pair"),3)
```

```
Warning in cor(y = nhanes, x = !is.na(nhanes), use = "pair"): 표준편차가 0입니다
```

```
       age    bmi    hyp   chl
age     NA     NA     NA    NA
bmi  0.086     NA  0.139 0.053
hyp  0.008     NA     NA 0.045
chl -0.040 -0.012 -0.107    NA
```

quickpred() 함수는 threshold correlation을 지정할 수도 있고 변수별로 다르게 설정할 수도 있다. 또한 include 옵션을 통해서 predictor에 항상 포함하고 싶은 변수를 지정할 수도 있다.

```
imp = mice(nhanes, pred = quickpred(nhanes, minpuc = 0.25, include = "age"))
```

3.4 Passive imputation

imputation을 하다보면 변수를 변환해야할 때가 있다. 예를 들어, numeric 변수를 impute할 때, log 변환이 정규성을 더 만족할 때가 있을 것이다. 이럴 때는 log 변환을 한 변수를 추가해야 하고 변환하지 않은 변수와 동시에 mice에서 돌아가지 않도록 해야 한다. 이러한 귀찮음을 해결해주는 방법을 mice에서는 제공한다.

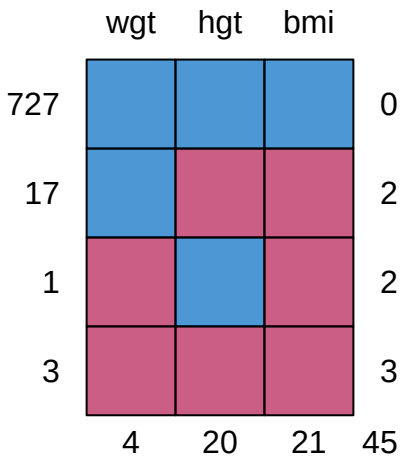
예를 들어 chl보다는 log(chl)을 통해서 bmi가 더 잘 예측된다고 생각해보면, log(chl) 칼럼을 추가하고 싶을 것이다. chl의 결측치는 log(chl)에서도 결측치이다.

```
nhanes2.ext = cbind(nhanes2, lchl = log(nhanes$chl))
ini = mice(nhanes2.ext, max = 0, print=FALSE)
meth = ini$meth
meth['lchl'] = '~log(chl)'
pred = ini$pred
pred['age',] = 0 # age variable does not have missing values
pred[c('hyp','chl','age'),'lchl'] = 0 # do not use log(chl) to impute hyp, chl
pred['bmi','chl'] = 0 # use log(chl) to impute bmi not chl
```

index of two variables

예를 들어 weight, height, bmi의 변수를 가지는 데이터를 생각해보자. bmi는 weight와 height로 계산되므로 둘 중 하나만 결측치라면 bmi도 결측치가 된다. 이러한 상황에서, weight와 height의 결측치를 모두 채운다면 bmi의 결측치를 채울 수 있다. 이를 아래와 같이 I를 이용하여 수행한다.

```
md.pattern(boys[, c('hgt','wgt','bmi')])
```




```
    wgt hgt bmi
727   1   1   1  0
17    1   0   0  2
1     0   1   0  2
3     0   0   0  3
      4  20  21 45
```

```
ini = mice(boys, max=0, print=FALSE)
meth = ini$meth
meth['bmi'] = '~I(wgt/(hgt/100)^2)'
pred = ini$predictorMatrix
pred[c('wgt','hgt','hc','reg'), 'bmi'] = 0
pred[c('gen','phb','tv'), c('hgt', 'wgt', 'hc')] = 0
pred
```

```
    age hgt wgt bmi hc gen phb tv reg
age   0   1   1   1  1   1   1  1   1
hgt   1   0   1   0  1   1   1  1   1
wgt   1   1   0   0  1   1   1  1   1
bmi   1   1   1   0  1   1   1  1   1
hc    1   1   1   0  0   1   1  1   1
gen   1   0   0   1  0   0   1  1   1
```

```
phb   1   0   0   1  0   1   0  1   1
tv    1   0   0   1  0   1   1  0   1
reg   1   1   1   0  1   1   1  1   0
```

```
imp.idx = mice(boys, pred=pred, meth=meth, maxit=20, seed=9212, print=FALSE)
head(complete(imp.idx)[is.na(boys$bmi), ], 3)
```

```
      age  hgt   wgt      bmi   hc gen phb tv   reg
18  0.087 53.5  4.54 15.86165 39.0  G1  P1  1  west
52  0.177 57.5  4.82 14.57845 40.4  G1  P1  3  west
174 1.481 85.5 12.04 16.47002 47.5  G1  P1  2 north
```

결측치였던 행의 bmi를 살펴보면, hgt, wgt를 채워 넣고 bmi 공식에 따라 채워진 결과를 확인할 수 있다.

Squeeze

imputation 값이 불가능한 값으로 나올 때, mice() 함수는 에러를 반환한다. 예를 들어, chl을 normal 분포로부터 impute하기 때문에 음수가 나올 수 있고 이는 log(chl)에서 에러를 반환할 것이다. 이를 아래 squeeze()를 통해 방지할 수 있다.

```
# squeeze
meth['lchl'] = '~log(squeeze(chl, bound=c(100,300)))'
imp = mice(nhanes2.ext, meth=meth, pred=pred, seed=1, maxit=1)
```

3.5 Post-processing imputations

어떤 범위 사이의 imputation을 원한다든지, 불가능한 값을 imputation 후보에서 제외한다든지 등을 원할 때의 imputation을 뜻한다. mice() function에서는 post 옵션을 통해서 이를 수행할 수 있다.

3.6 Visiting scheme

mice() 함수는 디폴트로 왼쪽에서 오른쪽으로 칼럼을 채워 넣는다. 이론상으로 iteration이 많으면 큰 상관이 없지만 결측치 갯수가 많은 순으로 mice() 함수가 돌아간다면 수렴이 더 빨리된다고 한다. 이를 vis의 옵션으로 명시할 수 있다.

```
imp = mice(nhanes2.ext, meth=meth, pred=pred, vis = 'monotone', print=FALSE)
```

4. Running MICE

4.1 Dry run

maxit = 0으로 옵션을 주고 mice() 함수를 돌리는 것을 뜻한다. 사실, 아무것도 안하는 것과 동일한데 유저가 method, predictorMatrix을 직접 넣고 싶을 때 이런 식으로 많이 시작한다.

4.2 Assessing convergence

아래는 수렴이 잘 안된 mice 결과이다.

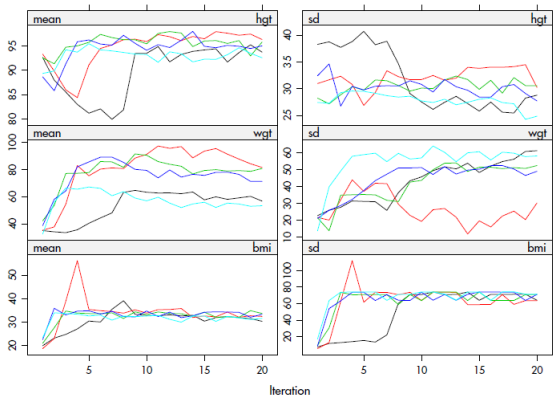


Figure 8: Non-convergence of the MICE algorithm. Imputations for `hgt`, `wgt` and `bmi` hardly mix and resolve into a steady state.

아래는 수렴이 잘 된 mice 결과이다.

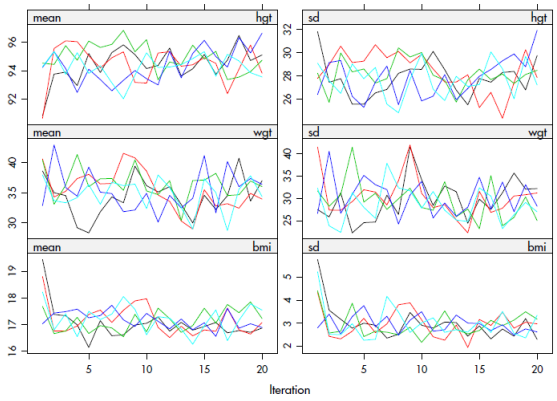


Figure 9: Healthy convergence of the MICE algorithm for `hgt`, `wgt` and `bmi`, where feedback loop of `bmi` into `hgt` and `wgt` is broken (solution `imp.idx`).

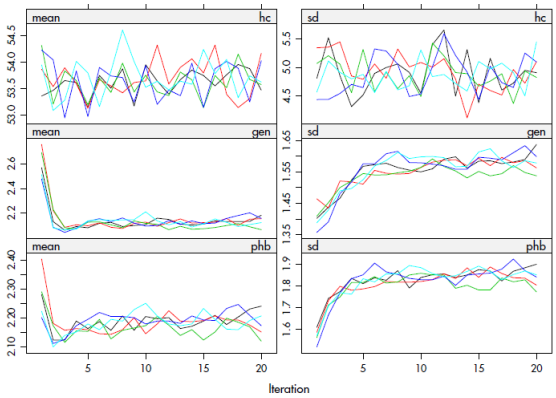
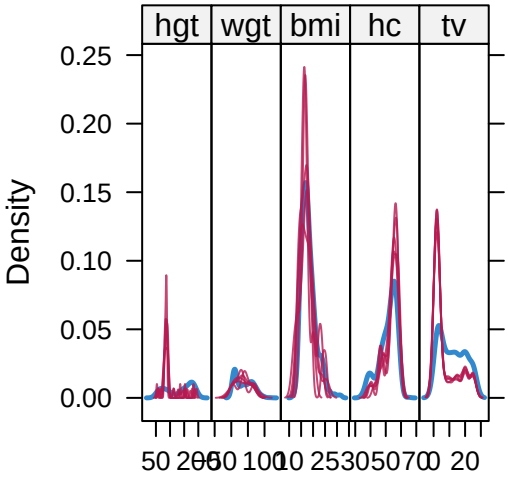


Figure 10: Healthy convergence of the MICE algorithm for `hc`, `gen` and `phb` showing a strong initial trend for the latter two (solution `imp.idx`).

4.3 Checking imputations

모든 변수의 관찰된 값과 impute된 값을 비교하여 imputations이 합리적인지 확인하는 것은 중요한 단계이다.

```
densityplot(imp.idx, scales = list(x = list(relation = "free")), layout = c(5, 1))
```



예를 들어, 위 분포에서 hgt 변수를 보면, impute된 hgt의 값들의 분포가 90 근처에서 크다. 이는 hgt가 결측치인 관측치들의 특성을 살펴보면 이유를 유추할 수 있다.

```
boys[is.na(boys$hgt),]$age
```

```
 [1]  0.087  1.481  1.494  1.530  1.585  1.675  1.697  1.839  1.848  1.867
[11]  1.911  1.938  1.957  1.960  1.973  1.979  1.990  5.820 11.696 19.526
```

hgt가 결측인 관측치의 나이는 상당히 어리므로 impute 된 hgt도 작게 나오는 것이다. 이를 통해 그럴듯한 값이 채워졌음을 확인할 수 있다.

5. Conclusion

이상으로 R에서 구현된 MICE 패키지를 뜯어보았다. 생각보다 매우 다양한 기능을 제공하는 듯하여 놀라웠다. 변수별 imputation model이 다르다는 것, 독립변수로 사용할 predictor를 직접 고른다는 것, imputation을 할 때 주의해야할 점, 마지막으로 imputation 진단까지 다양한 과정을 직접 해보았다. 결측치는 대부분의 데이터에 나타나기 때문에 앞으로 MICE를 잘 활용해야겠다.