Imputation(test)

Xinyi Lin

8/6/2019

The purpose of following codes is to test imputation ability of two packages–Himsc and missForest.

Import data

Import "training.dta" as target dataset.

Creat the variable

Notation:

Variable name: student_r

Created from: occup1_r, occup2_r

Label: student_r — yes-is a student, no-not a student, NA-occup1_r and occup2_r are NAs

```
## # A tibble: 6 x 8
##   ageyrs_r sex   student_r area  educate_r currmarr_r sexp1yr SEScat
##      <dbl> <fct> <fct>     <fct> <fct>     <fct>        <dbl> <fct>
## 1       19 male  no        2     1         0                1 0
## 2       18 male  yes       0     1         0                1 1
## 3       16 male  no        0     1         0               NA 2
## 4       15 male  no        0     1         0               NA 1
## 5       16 male  no        0     1         0               NA 1
## 6       17 male  no        0     1         0                1 3
```

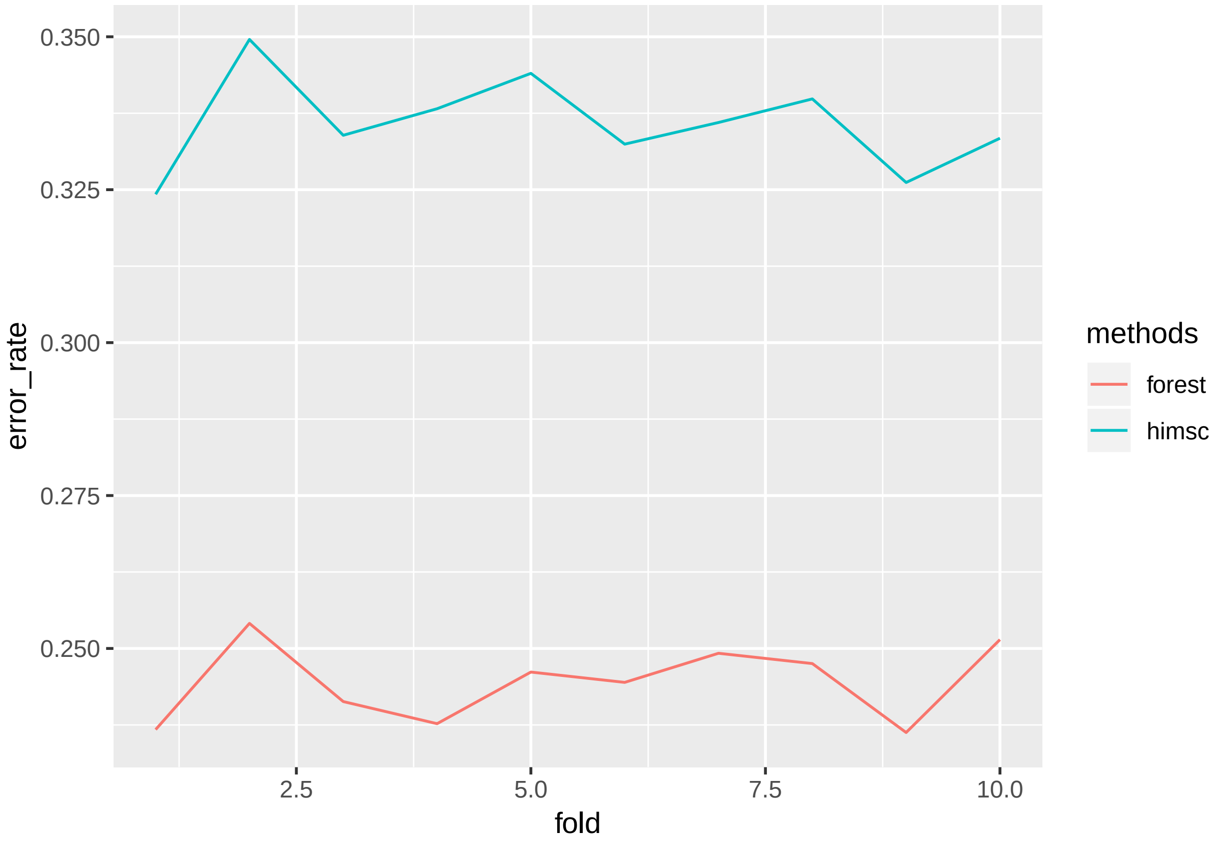
Test dataset

First, choose observations with known student_r as test dataset. There are 41482 observations in total.

Then, randomly split test dataset into ten subsets and set them as missing values to test error rates of each fold when using different methods.

We can find that error rates of Himsc package is around 33% and error rates of missForest is around 24%. Following is the plot of error rates.

```
## $himsc
##  [1] 0.3242527 0.3495661 0.3338959 0.3382353 0.3440212 0.3324494 0.3359846
##  [8] 0.3398409 0.3261813 0.3334137
## 
## $forest
##  [1] 0.2367406 0.2540984 0.2413211 0.2377049 0.2461427 0.2444552 0.2492167
##  [8] 0.2475295 0.2362584 0.2514465
```



```r
knitr::opts_chunk$set(echo = FALSE)
library(tidyverse)
library(missForest)
library(Hmisc)
library(caret)
library(haven)
training <- read_dta("training.dta")
student_data = training %>%
  as_data_frame() %>%
  mutate(student_r = ifelse(occup1_r == 8 | occup2_r == 8, "yes", "no"),
         student_r = ifelse(is.na(student_r) & occup1_r == 20, "no", student_r))
# for sexp1yr > 92, make them as NAs
impu_data = student_data %>%
  filter(ageyrs_r <= 19) %>%
  mutate(student_r = as.factor(student_r),
         visit = as.factor(visit),
         sex = ifelse(female==1, "female", "male")) %>%
  select(ageyrs_r, sex, student_r, area, educate_r, currmarr_r, sexp1yr, SEScat) %>%
  mutate(sex = as.factor(sex),
         area = as.factor(area),
         educate_r = as.factor(educate_r),
         currmarr_r = as.factor(currmarr_r),
         sexp1yr = ifelse(sexp1yr > 92, NA, sexp1yr),
         SEScat = as.factor(SEScat))
head(impu_data)
test_data = impu_data[!is.na(impu_data$student_r),]
flds <- createFolds(1:41482, k = 10, list = TRUE, returnTrain = FALSE)
```

```r
#flds
# error_rate = vector("list", 10)
set.seed(123)
test_data = as.data.frame(as.matrix(test_data))
error_himsc = rep(NA, 10)
error_forest = rep(NA, 10)
for (n in 1:10){
  na_data = test_data
  na_data[flds[[n]], 3] = NA
  # missForest
  impu_forest = missForest(na_data)
  impu_forest_df = impu_forest$ximp
  # Himsc
  impu_himsc = aregImpute(~ ageyrs_r + sex + student_r + area + educate_r + currmarr_r + SEScat, data =
  impu_himsc_l = impute.transcan(impu_himsc, data=na_data, imputation=1, list.out=TRUE, pr=FALSE, check=
  impu_himsc_df = as.data.frame(impu_himsc_l)
  error_himsc[n] = sum(abs(as.numeric(impu_himsc_df[flds[[n]], 3]) - as.numeric(test_data[flds[[n]], 3]
  error_forest[n] = sum(abs(as.numeric(impu_forest_df[flds[[n]], 3]) - as.numeric(test_data[flds[[n]], 3
}
res_error = list(himsc = error_himsc, forest = error_forest)
res_error
res_error %>%
  as.data.frame() %>%
  mutate(fold = 1:10) %>%
  gather(key = methods, value = error_rate, himsc:forest) %>%
  ggplot(aes(x = fold, y = error_rate, color = methods)) + geom_line()
```