Homework 7

Tauqeer Kasam Rumaney

December 5, 2022

Download this R Markdown file, save it on your computer, and perform all the below tasks by inserting your answer in text or by inserting R chunks below. After you are done, upload this file with your solutions on Moodle.

Preparations

Load KiGGS data:

```
dat_link <- url("https://www.dropbox.com/s/pd0z829pv2otzqt/KiGGS03_06.RData?dl=1")
load(dat_link)
kiggs <- KiGGS03_06
```

Exercise 1: Multiple imputation

a) In the KiGGS dataset, choose 2 metric variables of your choice. #PPoint and KreaUrin
b) Compute the correlation between these two variables

```
res0 <- cor.test(kiggs$PPoint, kiggs$KreaUrin, method = "pearson")
res0$estimate
```

```
##         cor
## 0.006408265
```

```
res0$conf.int
```

```
## [1] -0.009541319  0.022354588
## attr(,"conf.level")
## [1] 0.95
```

```
res0$p.value
```

```
## [1] 0.4309998
```

c) Perform a multiple imputation using the mice function with the 'pmm' method and otherwise default settings.

```
library(mice)
```

```
##
## Attaching package: 'mice'

## The following object is masked from 'package:stats':
##
##     filter

## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

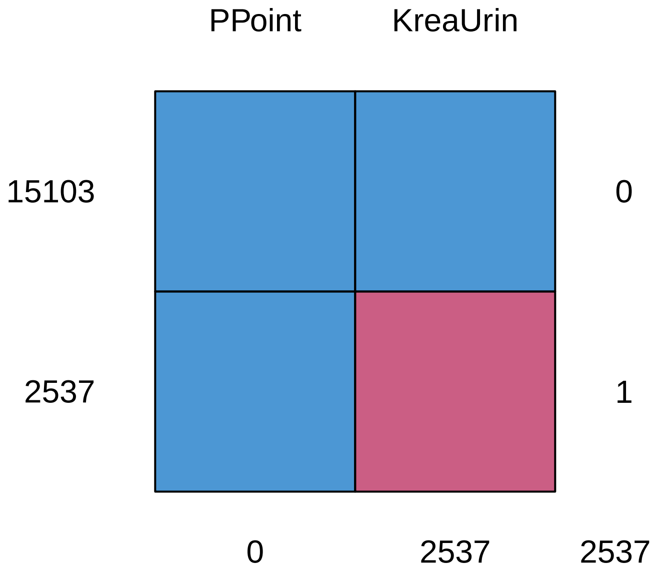
```
dat <- data.frame(PPoint = as.numeric(as.character(kiggs$PPoint)),
                  KreaUrin = as.numeric(as.character(kiggs$KreaUrin)))
table(!is.na(dat$PPoint))
```

```
##
##  TRUE
## 17640
```

```
table(!is.na(dat$KreaUrin))
```

```
##
## FALSE  TRUE
##  2537 15103
```

```
# Pattern of missing values
md.pattern(dat)
```

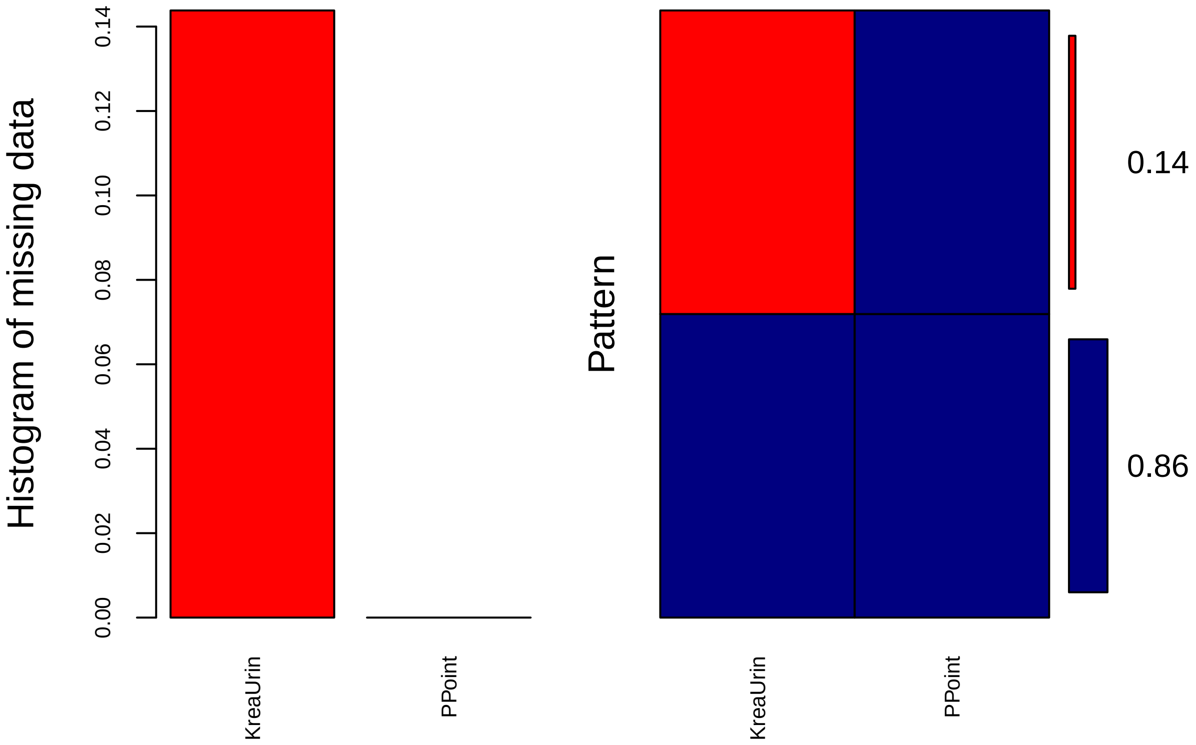


```
##       PPoint KreaUrin
## 15103      1        1    0
## 2537       1        0    1
##            0     2537 2537
```

```
# Nice visualization of missing values in the VIM package, can be useful to identify patterns of missin
library(VIM)
```

```
## Loading required package: colorspace

## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```r
aggr(dat, col = c('navyblue','red'), numbers = TRUE, sortVars = TRUE, labels = names(dat), cex.axis = 0
```



```
##
##  Variables sorted by number of missings:
##  Variable     Count
##  KreaUrin 0.1438209
##    PPoint 0.0000000
```

```r
# Step 1: Do imputation of missing values, generate 5 imputed datasets
tempData <- mice(dat, m = 5, maxit = 5, meth = 'pmm', seed = 500)
```

```
##
##  iter imp variable
##   1   1  KreaUrin
##   1   2  KreaUrin
##   1   3  KreaUrin
##   1   4  KreaUrin
##   1   5  KreaUrin
##   2   1  KreaUrin
##   2   2  KreaUrin
##   2   3  KreaUrin
##   2   4  KreaUrin
##   2   5  KreaUrin
##   3   1  KreaUrin
##   3   2  KreaUrin
##   3   3  KreaUrin
##   3   4  KreaUrin
##   3   5  KreaUrin
##   4   1  KreaUrin
##   4   2  KreaUrin
##   4   3  KreaUrin
##   4   4  KreaUrin
##   4   5  KreaUrin
##   5   1  KreaUrin
##   5   2  KreaUrin
##   5   3  KreaUrin
##   5   4  KreaUrin
##   5   5  KreaUrin
```

```
summary(tempData)
```

```
## Class: mids
## Number of multiple imputations:  5
## Imputation methods:
##   PPoint KreaUrin
##       ""    "pmm"
## PredictorMatrix:
##          PPoint KreaUrin
## PPoint        0        1
## KreaUrin      1        0
```

```
# The 5 imputed datasets can be extracted as follows:
completedData1 <- complete(tempData,1)
completedData2 <- complete(tempData,2)
completedData3 <- complete(tempData,3)
completedData4 <- complete(tempData,4)
completedData5 <- complete(tempData,5)
```

d) Compute the correlation again on the multiple imputed datasets.

```
# Step 2: Do the statistical analysis using the "with" function
modelFit1 <- with(tempData, cor(PPoint, KreaUrin, method = "pearson"))
mean(unlist(modelFit1$analyses))
```

```
## [1] 0.01576219
```

```
with(tempData, cor.test(PPoint, KreaUrin, method = "pearson"))
```

```
## call :
## with.mids(data = tempData, expr = cor.test(PPoint, KreaUrin,
##     method = "pearson"))
##
## call1 :
## mice(data = dat, m = 5, method = "pmm", maxit = 5, seed = 500)
##
## nmis :
##   PPoint KreaUrin
##        0     2537
##
## analyses :
## [[1]]
##
##  Pearson's product-moment correlation
##
## data:  PPoint and KreaUrin
## t = 0.50611, df = 17638, p-value = 0.6128
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.01094700  0.01856699
## sample estimates:
##         cor
## 0.003810827
##
##
## [[2]]
##
##  Pearson's product-moment correlation
##
## data:  PPoint and KreaUrin
## t = 0.0099473, df = 17638, p-value = 0.9921
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.01468232  0.01483209
## sample estimates:
##          cor
## 7.489944e-05
##
##
## [[3]]
##
##  Pearson's product-moment correlation
##
## data:  PPoint and KreaUrin
## t = 0.89743, df = 17638, p-value = 0.3695
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.008000854  0.021512215
## sample estimates:
##         cor
## 0.006757152
##
##
## [[4]]
##
##  Pearson's product-moment correlation
##
## data:  PPoint and KreaUrin
## t = 4.0299, df = 17638, p-value = 5.603e-05
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.01557962 0.04506689
## sample estimates:
##        cor
## 0.03032985
##
##
## [[5]]
##
##  Pearson's product-moment correlation
##
## data:  PPoint and KreaUrin
## t = 5.0288, df = 17638, p-value = 4.983e-07
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.02309390 0.05256607
## sample estimates:
##        cor
## 0.03783821
```

```
modelFit1 <- with(tempData, lm(PPoint ~ KreaUrin))
summary(pool(modelFit1), dfcom = 17638)
```

```
##          term    estimate  std.error  statistic       df      p.value
## 1 (Intercept) 83.07732597 1.47136242 56.4628569 6.058792 1.767292e-09
## 2    KreaUrin  0.01120885 0.01436333  0.7803798 5.372882 4.681392e-01
```