generate_and_ampute

Cem Kalender (2920734)

5/4/2022

```
# Setting working directory and loading libraries----
setwd("~/Applied Data Science/Thesis")

# import packages
library(lattice)
library(mice)
```

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

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

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

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.0
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## Warning: package 'ggplot2' was built under R version 4.2.0
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

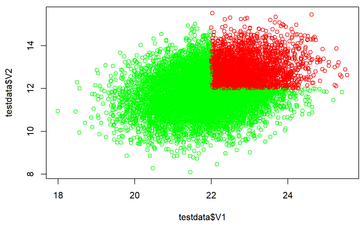
```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'stringr' was built under R version 4.2.0
```

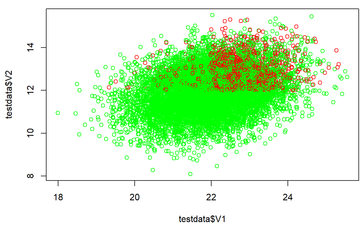
```
## Warning: package 'forcats' was built under R version 4.2.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks mice::filter(), stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
## GENERATE DATA ----

set.seed(12)
#options(scipen=999)

# randomly generate a dataset to be our complete dataset
testdata <- as.data.frame(MASS::mvrnorm(n = 10000,
                                        mu = c(22, 12, 0),      # V1   V2   V3
                                        Sigma = matrix(data = c(1.0, 0.3, 0.5,
                                                                0.3, 1.0, 0.5,
                                                                0.5, 0.5, 1.0),
                                                       nrow = 3,
                                                       byrow = T)))

# People falling in a certain region have a a higher probabilityof Y = 1
condition = testdata$V1>median(testdata$V1) & testdata$V2>median(testdata$V2)
condition2 = testdata$V1<median(testdata$V1) & testdata$V2>median(testdata$V2)
# plot
plot(testdata$V1, testdata$V2, col = ifelse(condition, 'red', 'green'))
```




```
# generate binomial variable for V3
testdata$V3 = ifelse(condition,rbinom(nrow(testdata[condition,]), size = 1, prob = 0.4),
                     ifelse(condition2,rbinom(nrow(testdata[condition2,]), size = 1, prob = 0.1), 0))
plot(testdata$V1, testdata$V2, col = ifelse(testdata$V3==1,'red', 'green'))
```



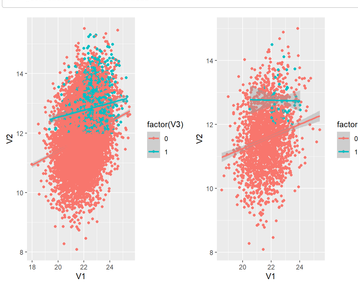

```
# descriptives
summary(testdata)
```

```
##        V1                V2                V3
##  Min.   :17.99   Min.   : 8.09   Min.   :0.0000
##  1st Qu.:21.33   1st Qu.:11.32   1st Qu.:0.0000
##  Median :22.00   Median :12.00   Median :0.0000
##  Mean   :22.00   Mean   :12.00   Mean   :0.1373
##  3rd Qu.:22.68   3rd Qu.:12.66   3rd Qu.:0.0000
##  Max.   :25.55   Max.   :15.51   Max.   :1.0000
```

```
# save as csv
## Save complete  dataframe as CSV file
write.csv(testdata,"C:\\Users\\surface\\Documents\\Applied Data Science\\Thesis\\complete_data.csv", row.names = FALSE)


## AMPUTE WITH MICE ----

# ampute the complete data once for every mechanism
ampdata1 <- ampute(testdata, patterns = c(0, 1, 1), prop = 0.8, mech = "MAR")$amp
ampdata2 <- ampute(testdata, patterns = c(1, 0, 1), prop = 0.8, mech = "MAR")$amp
ampdata3 <- ampute(testdata, patterns = c(1, 0, 1), prop = 0.8, mech = "MNAR")$amp
ampdata4 <- ampute(testdata, patterns = c(0, 0, 0), prop = 0.8, mech = "MCAR")$amp

# create a random allocation vector
# use the prob argument to specify how much of each mechanism should be created
# here, 0.5 of the missingness should be MAR and 0.5 should be MCAR
indices <- sample(x = c(1, 2, 3, 4), size = nrow(testdata),
                  replace = TRUE, prob = c(0.4, 0.4, 0.1, 0.1))

# create an empty data matrix
# fill this matrix with values from either of the two amputed datasets
ampdata <- matrix(NA, nrow = nrow(testdata), ncol = ncol(testdata))
ampdata[indices == 1, ] <- as.matrix(ampdata1[indices == 1, ])
ampdata[indices == 2, ] <- as.matrix(ampdata2[indices == 2, ])
ampdata[indices == 3, ] <- as.matrix(ampdata3[indices == 3, ])
ampdata[indices == 4, ] <- as.matrix(ampdata4[indices == 4, ])

# store as df
ampdata = data.frame(ampdata)
# change colnames
colnames(ampdata) = c('V1', 'V2', 'V3')


# Visualize
ggpubr::ggarrange(
    testdata %>% ggplot(aes(V1,V2, color = factor(V3))) + geom_point() + geom_smooth(method = 'lm'),
    na.omit(ampdata) %>% ggplot(aes(V1,V2, color = factor(V3))) + geom_point() + geom_smooth(method = 'lm')
)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
#save missing data
write.csv(ampdata, "C:\\Users\\surface\\Documents\\Applied Data Science\\Thesis\\missingdata.csv", row.names = FALSE)
```