Missing values imputation in clinical trials (linear and quadratic models)

Valery Liamtsau

2025-08-09

In this tutorial the imputation of missing data will be performed

```
# upload packges
library(readxl)
library(tidyverse) # get for tibble
library(lubridate)
library(ggplot2)
library(data.table)
library(tidyr)
library(ggplot2)
library(dplyr)
library(huxtable) # printing pretty table
# temporarily turn off warnings
options(warn=0)
```

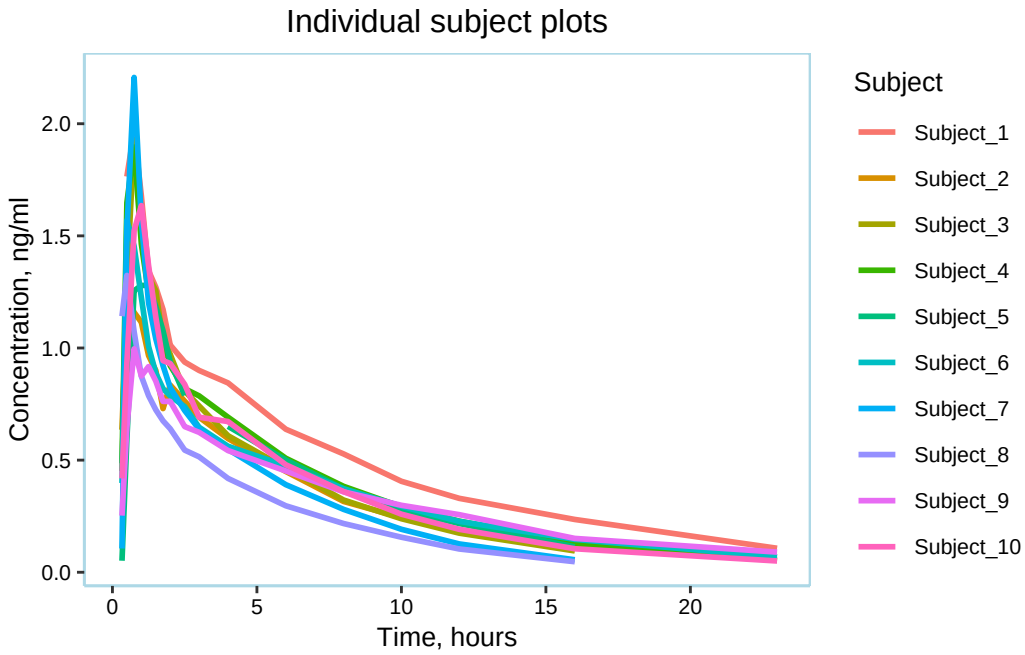
Import and data preprocessing

```
# set working directory
path_ <- "C:/Users/valer/Desktop/R_project/Project 5/data.xlsx"

# loading data containing test and reference datasheets
loading_data <- function(path_, sheet_) {

  # reading file
  data <- read_excel(path = path_, sheet = sheet_)

  # convert all types to numeric to enable further calculations
  data <- data |>
```

```
    mutate_all(as.numeric)
  sum(is.na(data))
  return (data)
}

# dataset for the test medicinal product
data_test <- loading_data(path_, "test_rand")

# values will not be imputed once all rows contain NAs,
# and corresponding sampling times will be dropped

# identify missing times that contain subjects with all missing data
col_subjects <- ncol(data_test) - 1

missing_times <-
  data_test[rowSums(is.na(data_test)) == col_subjects, ]$'Time'
df_non_missing <- data_test[data_test$"Time" != missing_times, ]

# data preprocessing from wide to long format
data_test_long <- melt(setDT(df_non_missing), id.vars = c("Time"),
             variable.name = "Subject", value.name = "Concentration")
```

Visualizing the data

```
# create time series plots for all subjects
p_all_subjects <-
  ggplot(data_test_long, aes(x=Time, y=Concentration)) +
  geom_line(aes(colour=Subject), size = 1) +
  xlab("Time, hours")+
  ylab("Concentration, ng/ml") +
  ggtitle('Individual subject plots') +
  theme(axis.line = element_line(color = "lightblue"), axis.text = element_text(color = "bla
        plot.title = element_text(hjust = .5),
        panel.grid.minor = element_blank(), panel.grid.major = element_blank(),
        panel.border = element_rect(color = "lightblue", fill = NA),
        panel.background = element_rect(fill = NA),
        text = element_text(size=10))

plot(p_all_subjects)
```



Identification of missing values

```
# create time series plots for subjects with missing data
missing_data_subjects <- c("Subject_1", "Subject_5")
df_missing <- df_non_missing |>
  select('Time', missing_data_subjects)
df_missing
```

Two data points missing from the entire dataset. First missing data point is for Subject 1 at time 0.330 hours; subject is selected at random using R. Second missing data point is for Subject 5 at time 3 and time 23 hours; subject is selected at random using R.

Identification of absorption and elimination phases. Dataframe subsetting for models fit based on the corresponding phases. Missingness will be computed by subsetting the dataset by two datasets: absorption and elimination phases.

```
# get the list of subjects to iterate over them later
subjects_list <- colnames(data_test)[2:ncol(data_test)]

# store Tmax values for all subjects
Tmax_values <- NULL

# iterate over subjects
```

```r
for (subject in subjects_list) {

  data_temp <- data_test |>
    select(1, subject)
  # find Cmax values
  Cmax <- max(data_temp[subject], na.rm = TRUE)
  # find Tmax values corresponding to Cmax
  Tmax <- filter(data_temp, data_temp[subject] == Cmax)$Time
  # store Tmax in a vector
  Tmax_values <- c(Tmax_values, Tmax)
}

# cut off value which will define absorption and elimination phase
Tmax_median <- median(Tmax_values)
# identify the absorption phase dataset
df_abs <- data_test[data_test['Time'] <= Tmax_median, ]
# data preprocessing from wide to long format
df_abs_long <- melt(setDT(df_abs), id.vars = c("Time"),
                        variable.name = "Subject",
                      value.name = "Concentration")

# identify the elimination phase
df_elim <- data_test[data_test['Time'] >= Tmax_median, ]
# data preprocessing from wide to long format
df_elim_long <- melt(setDT(df_elim), id.vars = c("Time"),
                        variable.name = "Subject",
                     value.name = "Concentration")

# identify subset for quadratic imputation
# the cutoff will be 2 hours timepoint (base on the observation of all plots)
data_quadr <- data_test[data_test['Time'] <= 2, ]
# data preprocessing from wide to long format
data_quadr_long <- melt(setDT(data_quadr), id.vars = c("Time"),
                        variable.name = "Subject",
                        value.name = "Concentration")
```

Fitting the data

```r
# Models description (absorption phase)
# Population model with time
# fit linear regression model using 'x' as predictor and 'y'
# as response variable
```

```r
model_abs <- lm(Concentration ~ Time, data=df_abs_long)
summary(model_abs)
```

```
Call:
lm(formula = Concentration ~ Time, data = df_abs_long)

Residuals:
     Min       1Q   Median       3Q      Max
-0.65111 -0.35709 -0.09898  0.30571  0.75164

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -0.2562     0.2320  -1.104    0.279
Time          2.5379     0.4205   6.036 1.66e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4168 on 28 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.5654,    Adjusted R-squared:  0.5499
F-statistic: 36.43 on 1 and 28 DF,  p-value: 1.663e-06
```

```r
# Multiple R-squared:  0.5654

# Quadratic model
#create a new variable for hours2
data_quadr_long$Time2 <- data_quadr_long$Time^2
#fit quadratic regression model
quadraticModel <- lm(Concentration ~ Time + Time2, data=data_quadr_long)
#view model summary
summary(quadraticModel)
```

```
Call:
lm(formula = Concentration ~ Time + Time2, data = data_quadr_long)

Residuals:
     Min       1Q   Median       3Q      Max
-0.71903 -0.29303 -0.04027  0.24574  0.99828
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2092     0.1755   1.192    0.237
Time          1.9905     0.3518   5.657 2.51e-07 ***
Time2        -0.8781     0.1502  -5.847 1.14e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3606 on 77 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.3082,    Adjusted R-squared:  0.2902
F-statistic: 17.15 on 2 and 77 DF,  p-value: 6.917e-07
```

```r
# Multiple R-squared:  0.3082
# R-squared is larger for the linear model rather than quadratic

# Population model with time
# fit linear regression model using 'x' as predictor and 'y'
# as response variable
model_elim <- lm(Concentration ~ Time, data=df_elim_long)
summary(model_elim)
```

```
Call:
lm(formula = Concentration ~ Time, data = df_elim_long)

Residuals:
     Min       1Q   Median       3Q      Max
-0.39962 -0.19148 -0.08736  0.10341  1.17802

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.076856   0.033763   31.89   <2e-16 ***
Time        -0.064718   0.004253  -15.21   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.286 on 142 degrees of freedom
  (16 observations deleted due to missingness)
Multiple R-squared:  0.6198,    Adjusted R-squared:  0.6171
F-statistic: 231.5 on 1 and 142 DF,  p-value: < 2.2e-16
```

```r
# Multiple R-squared:  0.6198

# Quadratic model
#create a new variable for hours2
data_quadr_long$Time2 <- data_quadr_long$Time^2
#fit quadratic regression model
quadraticModel <- lm(Concentration ~ Time + Time2, data=data_quadr_long)
#view model summary
summary(quadraticModel)
```

```
Call:
lm(formula = Concentration ~ Time + Time2, data = data_quadr_long)

Residuals:
     Min       1Q   Median       3Q      Max
-0.71903 -0.29303 -0.04027  0.24574  0.99828

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2092     0.1755   1.192    0.237
Time          1.9905     0.3518   5.657 2.51e-07 ***
Time2        -0.8781     0.1502  -5.847 1.14e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3606 on 77 degrees of freedom
  (20 observations deleted due to missingness)
Multiple R-squared:  0.3082,    Adjusted R-squared:  0.2902
F-statistic: 17.15 on 2 and 77 DF,  p-value: 6.917e-07
```

```r
# Multiple R-squared:  0.3082
# R-squared is larger for the linear model rather than quadratic
```

```r
# imputation of missing values

# Subject_1 should be imputed by the absorption linear model
# use model to predict points value
selected_subject <- c('Subject_1')
df_subject1 <- df_non_missing |>
  select(1, selected_subject)
predict(model_abs, df_subject1)
```

```
        1         2         3         4         5         6         7
0.1676638 0.5813463 1.0127943 1.6472767 2.2817591 2.9162415 3.5507239
        8         9        10        11        12        13        14
4.1852062 4.8196886 6.0886534 7.3576182 9.8955478 14.9714069 20.0472660
       15        16        17        18
25.1231251 30.1989842 40.3507025 58.1162094
```

```
# Subject_5 should be imputed by the elimination linear model
# use model to predict points value
selected_subject <- c('Subject_5')
df_subject5 <- df_non_missing |>
  select(1, selected_subject)
predict(model_elim, df_subject5)
```

```
         1          2          3          4          5          6
1.06604774 1.05549877 1.04449677 1.02831735 1.01213794 0.99595853
         7          8          9         10         11         12
0.97977912 0.96359971 0.94742030 0.91506147 0.88270265 0.81798500
        13         14         15         16         17         18
0.68854971 0.55911442 0.42967912 0.30024383 0.04137325 -0.41165028
```

Time	Subject_1	Subject_5
0.167	0.0653	
0.33		0.0514
0.5	1.76	0.596
0.75	1.97	1.26
1	1.68	1.28
1.25	1.34	1.28
1.5	1.27	1.18
1.75	1.17	1.08
2	1.01	0.938
2.5	0.937	0.789
3	0.9	
4	0.844	0.651
6	0.638	0.501
8	0.528	0.371
10	0.406	0.277
12	0.33	0.217
16	0.235	0.113
23	0.107	