Clinical Data Wrangling

Sebastian Quirarte | Nov 2024

- Overview
- Data
- Objective
- Loading Packages and Data
- Exploring Dataframes
- Missing Data
- Transforming Data
- Exporting Transformed Data

Overview

This R Markdown document aims to show basic data wrangling skills applied to fake clinical trial data. This serves as an example of real-world clinical data wrangling, based on a simplified dataset that has a similar structure to what can be found in data extraction files of EDC systems to implement the eCRF of clincial phase l-lV clinical trials.

Data

This example is based on an excel file ('fake-clinical-data.xlsx') that is made up of fake clinical data of 20 subjects and 40 varibles, structured across five excel sheets. In a real clinical trial, these files are significantly larger, include more subjects, variables, and sheets, in some cases including thousands of rows per sheet and dozens of sheets. They can also be found in .xlsx or .csv file formats, depending on the EDC platform implemented.

Objective

Every clinical trial will be different, depending on it's primary and secondary objectives, variables, treatments implemented, clinical tests, and other eCRF considerations.

In this example, the goal is to take the previously mentioned excel file, load the different sheets into R as dataframes, explore, and transform the data into a new dataframe that includes the variables of interest, simplifying statistical analysis, data review, and descriptive analysis of the data and it's use by technical and non-technical users involved in the clinical trial.

Loading Packages and Data

```
rm(list = ls()) # delete previously stored variables

# Packages
library(readxl) # read excel file
library(openxlsx) # write excel file

# Load excel sheets as dataframes
demographic <- read_excel('fake-clinical-data.xlsx', sheet = 'demographic')
baseline <- read_excel('fake-clinical-data.xlsx', sheet = 'baseline')
variables_l <- read_excel('fake-clinical-data.xlsx', sheet = 'variables_l')
variables_ll <- read_excel('fake-clinical-data.xlsx', sheet = 'variables_ll')
variables_lll <- read_excel('fake-clinical-data.xlsx', sheet = 'variables_lll')
```

Exploring Dataframes

demographic: dataframe of demographic variables of 20 subjects [20 x 7].

```
str(demographic)
```

```
## tibble [20 × 7] (S3: tbl_df/tbl/data.frame)
##  $ subject_ID: chr [1:20] "001" "002" "003" "004" ...
##  $ initials  : chr [1:20] "RQS" "LVF" "GGU" "AKY" ...
##  $ birth     : POSIXct[1:20], format: "1970-07-30" "1972-05-25" ...
##  $ age       : num [1:20] 54 52 36 38 45 39 41 63 35 54 ...
##  $ weight_kg : num [1:20] 68 62 87 69 91 96 57 77 69 88 ...
##  $ height_cm : num [1:20] 173 161 185 145 145 152 180 183 144 144 ...
##  $ sex       : chr [1:20] "F" "F" "F" "M" ...
```

baseline: dataframe of baseline variables of 20 subjects [20 x 7]. One row per study subject.

```
str(baseline)
```

```
## tibble [20 × 8] (S3: tbl_df/tbl/data.frame)
##  $ subject_ID: chr [1:20] "001" "002" "003" "004" ...
##  $ treat     : chr [1:20] "A" "B" "B" "B" ...
##  $ visit     : chr [1:20] "visit_1" "visit_1" "visit_1" "visit_1" ...
##  $ var_1     : num [1:20] 1 2 2 2 2 2 1 2 1 2 ...
##  $ var_2     : num [1:20] 12 13 18 7 6 19 12 5 18 5 ...
##  $ var_3     : num [1:20] 122 53 60 78 101 180 103 188 53 71 ...
##  $ var_4     : num [1:20] 57 42 47 54 98 61 80 68 49 51 ...
##  $ var_5     : num [1:20] 51 41 60 42 31 45 49 44 38 53 ...
```

variables_l: dataframe of additional variables of 20 subjects [100 x 12]. Five visits (rows) per study subject.

```
str(variables_l)
```

```
## tibble [100 × 12] (S3: tbl_df/tbl/data.frame)
##  $ subject_ID: chr [1:100] "001" "001" "001" "001" ...
##  $ visit     : chr [1:100] "visit_1" "visit_2" "visit_3" "visit_4" ...
##  $ var_16    : num [1:100] 40 46 NA NA NA 76 87 25 29 NA ...
##  $ var_17    : num [1:100] 90 53 NA NA NA 52 33 96 1 NA ...
##  $ var_18    : num [1:100] 29 3 NA NA NA 1 18 87 45 NA ...
##  $ var_19    : num [1:100] 74 9 NA NA NA 20 92 30 43 NA ...
##  $ var_20    : num [1:100] 44 95 NA NA NA 26 50 12 67 NA ...
##  $ var_21    : num [1:100] 1 25 NA NA NA 2 79 50 7 NA ...
##  $ var_22    : num [1:100] 17 28 NA NA NA 32 46 67 25 NA ...
##  $ var_23    : num [1:100] 65 80 NA NA NA 14 100 52 78 NA ...
##  $ var_24    : num [1:100] 73 81 NA NA NA 27 93 40 4 NA ...
##  $ var_25    : num [1:100] 48 72 NA NA NA 91 65 66 8 NA ...
```

variables_ll: dataframe of additional variables of 20 subjects [100 x 8]. Five visits (rows) per study subject.

```
str(variables_ll)
```

```
## tibble [100 × 7] (S3: tbl_df/tbl/data.frame)
##  $ subject_ID: chr [1:100] "001" "001" "001" "001" ...
##  $ visit     : chr [1:100] "visit_1" "visit_2" "visit_3" "visit_4" ...
##  $ var_26    : num [1:100] 12 30 NA NA NA 53 62 31 10 NA ...
##  $ var_27    : num [1:100] 4 66 NA NA NA 65 35 32 62 NA ...
##  $ var_28    : num [1:100] 6 45 NA NA NA 16 16 74 55 NA ...
##  $ var_29    : num [1:100] 40 36 NA NA NA 39 38 57 77 NA ...
##  $ var_30    : num [1:100] 21 58 NA NA NA 25 29 4 87 NA ...
```

variables_lll: dataframe of additional variables of 20 subjects [100 x 12]. Five visits (rows) per study subject.

```
str(variables_lll)
```

```
## tibble [100 × 12] (S3: tbl_df/tbl/data.frame)
##  $ subject_ID: chr [1:100] "001" "001" "001" "001" ...
##  $ visit     : chr [1:100] "visit_1" "visit_2" "visit_3" "visit_4" ...
##  $ var_31    : num [1:100] 43 31 NA NA NA 27 46 7 18 NA ...
##  $ var_32    : num [1:100] 37 9 NA NA NA 82 98 38 89 NA ...
##  $ var_33    : num [1:100] 8 84 NA NA NA 48 66 45 32 NA ...
##  $ var_34    : num [1:100] 68 100 NA NA NA 1 35 61 7 NA ...
##  $ var_35    : num [1:100] 96 54 NA NA NA 76 94 56 64 NA ...
##  $ var_36    : num [1:100] 48 49 NA NA NA 79 86 96 53 NA ...
##  $ var_37    : num [1:100] 66 74 NA NA NA 26 74 35 97 NA ...
##  $ var_38    : num [1:100] 60 11 NA NA NA 58 66 30 5 NA ...
##  $ var_39    : num [1:100] 12 7 NA NA NA 48 27 19 14 NA ...
##  $ var_40    : num [1:100] 31 50 NA NA NA 69 18 34 65 NA ...
```

Missing Data

The datasets ***variables_l***, ***variables_ll***, and ***variables_lll*** contain empty cells.

```
head(variables_l)
```

```
## # A tibble: 6 × 12
##   subject_ID visit   var_16 var_17 var_18 var_19 var_20 var_21 var_22 var_23
##   <chr>      <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 001        visit_1     40     90     29     74     44      1     17     65
## 2 001        visit_2     46     53      3      9     95     25     28     80
## 3 001        visit_3     NA     NA     NA     NA     NA     NA     NA     NA
## 4 001        visit_4     NA     NA     NA     NA     NA     NA     NA     NA
## 5 001        visit_5     NA     NA     NA     NA     NA     NA     NA     NA
## 6 002        visit_1     76     52      1     20     26      2     32     14
## # ℹ 2 more variables: var_24 <dbl>, var_25 <dbl>
```

```
head(variables_ll)
```

```
## # A tibble: 6 × 7
##   subject_ID visit   var_26 var_27 var_28 var_29 var_30
##   <chr>      <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 001        visit_1     12      4      6     40     21
## 2 001        visit_2     30     66     45     36     58
## 3 001        visit_3     NA     NA     NA     NA     NA
## 4 001        visit_4     NA     NA     NA     NA     NA
## 5 001        visit_5     NA     NA     NA     NA     NA
## 6 002        visit_1     53     65     16     39     25
```

```
head(variables_lll)
```

```
## # A tibble: 6 × 12
##   subject_ID visit   var_31 var_32 var_33 var_34 var_35 var_36 var_37 var_38
##   <chr>      <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 001        visit_1     43     37      8     68     96     48     66     60
## 2 001        visit_2     31      9     84    100     54     49     74     11
## 3 001        visit_3     NA     NA     NA     NA     NA     NA     NA     NA
## 4 001        visit_4     NA     NA     NA     NA     NA     NA     NA     NA
## 5 001        visit_5     NA     NA     NA     NA     NA     NA     NA     NA
## 6 002        visit_1     27     82     48      1     76     79     26     58
## # ℹ 2 more variables: var_39 <dbl>, var_40 <dbl>
```

In a clinical setting it is important to determine why there are missing values, in this case they are not due to data capture error but instead due to the fact that in this example the EDC system creates all five visits (rows) for each subject, even if they still haven't completed all of their visits.

Every scenario is different, in some cases data imputation is the best course of action, in other cases it is best to modify the missing values to a string value, but in this case we identified the reason for missing data and no modifications are necessary at this point.

Transforming Data

For this example, let's suppose we are interested in creating a dataframe that includes the following variables:

- ***subject_ID***
- ***visit***
- ***sex***
- ***treat***
- ***var_1***
- ***var_17***
- ***var_24***
- ***var_28***
- ***var_40***

These variables are found in different sheets of the original excel file, which we have loaded as individual dataframes. Some dataframes have only one row per subject and others have multiple rows per subject, one for each visit.

Dataframes can be combined using the ***merge()*** function. We'll only select the variables of interest from each dataframe. It is important to note that it is necessary to first add the variables found in the dataframes with repeated rows for each subject, so that all visits are loaded into the dataframe.

```
# dataframes with multiple rows per subject
df <- merge(variables_l[c('subject_ID', 'visit', 'var_17', 'var_24')],
            variables_ll[c('subject_ID', 'visit', 'var_28')])
df <- merge(df,
            variables_lll[c('subject_ID', 'visit', 'var_40')])

# display first 10 rows of dataframe
head(df, 10)
```

```
##    subject_ID   visit var_17 var_24 var_28 var_40
## 1         001 visit_1     90     73      6     31
## 2         001 visit_2     53     81     45     50
## 3         001 visit_3     NA     NA     NA     NA
## 4         001 visit_4     NA     NA     NA     NA
## 5         001 visit_5     NA     NA     NA     NA
## 6         002 visit_1     52     27     16     69
## 7         002 visit_2     33     93     16     18
## 8         002 visit_3     96     40     74     34
## 9         002 visit_4      1      4     55     65
## 10        002 visit_5     NA     NA     NA     NA
```

```
# dataframes with one row per subject
df <- merge(df,
            demographic[c('subject_ID', 'sex')])
df <- merge(df,
            baseline[c('subject_ID', 'treat', 'var_1')])

# display first 10 rows of dataframe
head(df, 10)
```

```
##    subject_ID   visit var_17 var_24 var_28 var_40 sex treat var_1
## 1         001 visit_1     90     73      6     31   F     A     1
## 2         001 visit_2     53     81     45     50   F     A     1
## 3         001 visit_3     NA     NA     NA     NA   F     A     1
## 4         001 visit_4     NA     NA     NA     NA   F     A     1
## 5         001 visit_5     NA     NA     NA     NA   F     A     1
## 6         002 visit_1     52     27     16     69   F     B     2
## 7         002 visit_2     33     93     16     18   F     B     2
## 8         002 visit_3     96     40     74     34   F     B     2
## 9         002 visit_4      1      4     55     65   F     B     2
## 10        002 visit_5     NA     NA     NA     NA   F     B     2
```

We now have a transformed dataframe that includes the variables of interest previously mentioned.

```
# display first and last rows of dataframe
head(df, 10)
```

```
##    subject_ID   visit var_17 var_24 var_28 var_40 sex treat var_1
## 1         001 visit_1     90     73      6     31   F     A     1
## 2         001 visit_2     53     81     45     50   F     A     1
## 3         001 visit_3     NA     NA     NA     NA   F     A     1
## 4         001 visit_4     NA     NA     NA     NA   F     A     1
## 5         001 visit_5     NA     NA     NA     NA   F     A     1
## 6         002 visit_1     52     27     16     69   F     B     2
## 7         002 visit_2     33     93     16     18   F     B     2
## 8         002 visit_3     96     40     74     34   F     B     2
## 9         002 visit_4      1      4     55     65   F     B     2
## 10        002 visit_5     NA     NA     NA     NA   F     B     2
```

```
tail(df, 10)
```

```
##     subject_ID   visit var_17 var_24 var_28 var_40 sex treat var_1
## 91         019 visit_1     66     60     36     29   M     A     1
## 92         019 visit_2     76     31      4    100   M     A     1
## 93         019 visit_3     21     66     77     20   M     A     1
## 94         019 visit_4     56     21     58      2   M     A     1
## 95         019 visit_5     NA     NA     NA     NA   M     A     1
## 96         020 visit_1     91     43     21     32   M     A     2
## 97         020 visit_2     41     49     32     71   M     A     2
## 98         020 visit_3     52     91     27     43   M     A     2
## 99         020 visit_4     81     91     42     27   M     A     2
## 100        020 visit_5     NA     NA     NA     NA   M     A     2
```

Exporting Transformed Data

Lastly, we'll export the transformed dataframe as an excel file.

Depending on the coding of the variables, it is sometimes useful to change the column names before exporting, especially if this data is intended for non-technical users sucha as doctors and clinical personnel involved in the clinical trial. In this case variable coding has been very simple but in real-world clinical data extractions variables can be coded in complex and abstarct manners due to the amount of variables involved and often require the use of a codebook that contains variable names and descriptions.

```
# change dataframe column names
colnames(df) <- c('ID', 'Visit', 'Var17', 'Var24', 'Var28', 'Var40', 'Sex', 'Trea
tment', 'Var1')

# display first and last rows of dataframe
head(df, 10)
```

```
##     ID   Visit Var17 Var24 Var28 Var40 Sex Treatment Var1
## 1  001 visit_1    90    73     6    31   F         A    1
## 2  001 visit_2    53    81    45    50   F         A    1
## 3  001 visit_3    NA    NA    NA    NA   F         A    1
## 4  001 visit_4    NA    NA    NA    NA   F         A    1
## 5  001 visit_5    NA    NA    NA    NA   F         A    1
## 6  002 visit_1    52    27    16    69   F         B    2
## 7  002 visit_2    33    93    16    18   F         B    2
## 8  002 visit_3    96    40    74    34   F         B    2
## 9  002 visit_4     1     4    55    65   F         B    2
## 10 002 visit_5    NA    NA    NA    NA   F         B    2
```

```
tail(df, 10)
```

```
##      ID   Visit Var17 Var24 Var28 Var40 Sex Treatment Var1
## 91  019 visit_1    66    60    36    29   M         A    1
## 92  019 visit_2    76    31     4   100   M         A    1
## 93  019 visit_3    21    66    77    20   M         A    1
## 94  019 visit_4    56    21    58     2   M         A    1
## 95  019 visit_5    NA    NA    NA    NA   M         A    1
## 96  020 visit_1    91    43    21    32   M         A    2
## 97  020 visit_2    41    49    32    71   M         A    2
## 98  020 visit_3    52    91    27    43   M         A    2
## 99  020 visit_4    81    91    42    27   M         A    2
## 100 020 visit_5    NA    NA    NA    NA   M         A    2
```

Now we can export our dataframe as an excel file using the ***write.xlsx()*** function from the ***openxlsx*** package.

```
write.xlsx(df, file = 'transformed-df.xlsx')
```