Lazy Data Exploration

Paula Reyes for the MJFF DCoP

When you frequently work with epidemiological data, it becomes a repetitive activity to explore datasets and perform routine statistics. Today, I'd like to share with you some libraries and functions in R that make this task easier.

Let's employ a readly available dataset as example data. In this case, I've decided to use a dataset from the package PBImisc, creates by Przemyslaw Biecek and which includes many datasets. Particularly, I loaded the dataset "eden" which is an artificial dataset based on an original study of European day hospital evaluation

```
library(PBImisc)
data(eden)
```

glimpse()

Function from the dplyr library This function provides a quick overview of a dataframe, showing the structure of the data in a compact format. It displays the number of rows, columns, and a preview of the data types and values. It is especially useful for large or complex datasets. Also it saves a lot of time when you are using this dataframe as an input for other function (yes, it turns out your variable was an integer while you needed a double and that is why your code is not running)

```
library(dplyr)
```

```
glimpse(eden)
```

```
## Rows: 642
## Columns: 12
## $ mdid               <fct> 12, 11, 10, 12, 11, 13, 12, 13, 11, 11, 12, 12, 13,~
## $ center             <fct> Dresden, Dresden, Dresden, Dresden, Dresden, Dresde~
## $ BPRS.Maniac        <dbl> 1.8, 1.8, 1.0, 1.3, 1.2, 1.3, 1.4, 1.5, 1.0, 1.1, 1~
## $ BPRS.Negative      <dbl> 2.3, 1.3, 2.0, 1.1, 1.9, 1.9, 1.4, 1.5, 1.0, 2.0, 1~
## $ BPRS.Positive      <dbl> 2.1, 1.1, 1.3, 1.1, 1.4, 1.3, 1.6, 1.2, 1.0, 1.1, 1~
## $ BPRS.Depression    <dbl> 3.4, 2.1, 3.0, 1.5, 3.4, 4.0, 3.1, 2.0, 1.9, 1.0, 1~
## $ BPRS.Average       <dbl> 2.400, 1.575, 1.825, 1.250, 1.975, 2.125, 1.875, 1.~
## $ MANSA              <dbl> 3.1, 3.0, 4.1, 4.0, 4.1, 4.0, 4.2, 4.1, 5.1, 5.1, 5~
## $ sex                <fct> man, woman, woman, woman, woman, woman, woman, man,~
## $ children           <int> 2, 1, 3, 0, 2, 1, 2, 2, 0, 0, 0, 2, 0, 0, 0, 2, 2, ~
## $ years.of.education <int> 12, 17, 10, 12, 14, 15, 18, 18, 10, 10, 10, 12, 14,~
## $ day                <dbl> 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, ~
```

For context this is what each variable means (according to the webpage):

mdid: Medical doctor id, there are 24 different MDs which examine patients

center: City in which the examination takes place

BPRS.Maniac, BPRS.Negative, BPRS.Positive, BPRS.Depression: BPRS stands for Brief Psychiatric Rating Scale, scores are averaged in four subscales

BPRS.Average: Average from 24 questions

*MANSA*Scale which measures Quality of Life (Manchester Short Assessment of Quality of Life)

sex: Sex

children: Number of children

years.of.education: Number of years of education

day: Hospitalization mode, day or stationary

tabyl()

Function from the janitor library You can use this to create frequency tables and cross-tabulations. It shows counts and proportions of categorical data, which helps in understanding the distribution of your data.

```
library(janitor)
```

For instance, we can check how much data we have per Center

```
eden%>%
  tabyl(center)
```

```
##      center   n    percent
##     Dresden 147 0.22897196
##      London  62 0.09657321
##  Michalovce 114 0.17757009
##      Prague 153 0.23831776
##     Wroclaw 166 0.25856698
```

Or what is the frequency of men and women having 0,1,2,3,4 or 5 children

```
eden%>%
  tabyl(sex,children)
```

```
##    sex   0  1   2  3  4 5
##    man 125 40  47 16  7 4
##  woman 127 95 139 30 10 2
```

CreateTableOne()

Function from the tableone library This function allows you to easily create a summary statistics of a study, which is common in clinical research. It provides descriptive statistics for continuous and categorical variables, and it allows for stratification by groups. Personally I wouldn't use this one for my final table on a manuscript, specially I recommend double checking if the test for p-values is the one you want, but it is incredibly useful for checking all data all at once.

```
library(tableone)
```

For instance, here we can see a table that displays summary statistics for all variables in our dataset, broken down by the levels of the "sex" variable.

```
CreateTableOne(data = eden, strata = "sex")
```

```
##                                  Stratified by sex
##                                   man           woman          p      test
##   n                                 239           403
##   mdid (%)                                                     0.188
##      10                               8 ( 3.3)     12 (  3.0)
##      11                               9 ( 3.8)     30 (  7.4)
##      12                              11 ( 4.6)     28 (  6.9)
##      13                              12 ( 5.0)     25 (  6.2)
##      14                               4 ( 1.7)      8 (  2.0)
##      20                               5 ( 2.1)      2 (  0.5)
##      21                              16 ( 6.7)     17 (  4.2)
##      22                               6 ( 2.5)     12 (  3.0)
##      23                               2 ( 0.8)      2 (  0.5)
##      30                              17 ( 7.1)      9 (  2.2)
##      31                              19 ( 7.9)     20 (  5.0)
##      32                              12 ( 5.0)     19 (  4.7)
##      33                               9 ( 3.8)      9 (  2.2)
##      40                               4 ( 1.7)      9 (  2.2)
##      41                              16 ( 6.7)     27 (  6.7)
##      42                              14 ( 5.9)     22 (  5.5)
##      43                              17 ( 7.1)     24 (  6.0)
##      44                               6 ( 2.5)     14 (  3.5)
##      50                               7 ( 2.9)     14 (  3.5)
##      51                              12 ( 5.0)     19 (  4.7)
##      52                              11 ( 4.6)     27 (  6.7)
##      53                               7 ( 2.9)     19 (  4.7)
##      54                               8 ( 3.3)     24 (  6.0)
##      55                               7 ( 2.9)     11 (  2.7)
##   center (%)                                                   0.003
##      Dresden                         44 (18.4)    103 ( 25.6)
##      London                          29 (12.1)     33 (  8.2)
##      Michalovce                      57 (23.8)     57 ( 14.1)
##      Prague                          57 (23.8)     96 ( 23.8)
##      Wroclaw                         52 (21.8)    114 ( 28.3)
##   BPRS.Maniac (mean (SD))          1.43 (0.52)   1.32 (0.36)    0.001
##   BPRS.Negative (mean (SD))        1.76 (0.75)   1.59 (0.63)    0.003
##   BPRS.Positive (mean (SD))        1.50 (0.74)   1.37 (0.58)    0.019
##   BPRS.Depression (mean (SD))      2.18 (0.93)   2.35 (1.03)    0.037
##   BPRS.Average (mean (SD))         1.71 (0.50)   1.66 (0.47)    0.137
##   MANSA (mean (SD))                4.48 (0.96)   4.53 (0.93)    0.542
##   sex = woman (%)                     0 ( 0.0)    403 (100.0)  <0.001
##   children (mean (SD))             0.96 (1.23)   1.27 (1.09)    0.001
##   years.of.education (mean (SD)) 12.59 (2.93)  12.17 (2.72)    0.067
##   day (mean (SD))                  0.48 (0.50)   0.53 (0.50)    0.230
```

summ()

Function from the epiDisplay library Overall, epiDisplay is a package for data exploration and result presentation of epidemiologcal data. It contains the full Epicalc package which provides a variety of functions for data management, descriptive statistics, and result presentation, as well as functions for calculating descriptive statistics, creating contingency tables, and running common epidemiological tests such as relative risk, odds ratios, and logistic regression models.

This package is very extensive and includes their own datasets. (It has many functions that can be covered with R-base tho). I highly suggest checking the documentation because there's too much to cover!

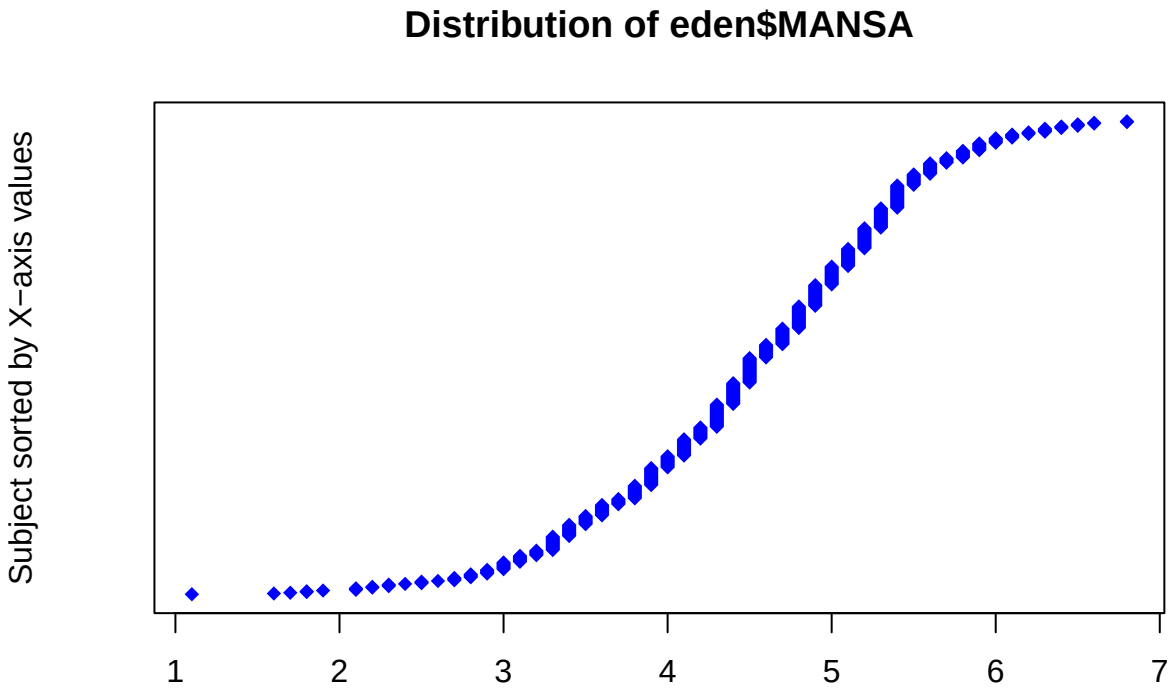
However, summ() creates a summary of data frame in a convenient table. And can also create statistics and a graph for a certain variable

```
library(epiDisplay)
```

```
summ(eden)
```

```
##
## No. of observations = 642
##
##    Var. name          obs. mean   median  s.d.   min.   max.
## 1  mdid               642  12.754 13      7.015  1      24
## 2  center             642  3.201  3       1.499  1      5
## 3  BPRS.Maniac        642  1.36   1.3     0.43   1      5
## 4  BPRS.Negative      642  1.65   1.5     0.68   1      5.4
## 5  BPRS.Positive      642  1.42   1.2     0.65   1      5.5
## 6  BPRS.Depression    642  2.28   2.1     1      1      6
## 7  BPRS.Average       642  1.68   1.58    0.48   1      3.92
## 8  MANSA              642  4.51   4.55    0.94   1.1    6.8
## 9  sex                642  1.628  2       0.484  1      2
## 10 children           642  1.16   1       1.16   0      5
## 11 years.of.education 642  12.32  12      2.81   5      20
## 12 day                642  0.51   1       0.5    0      1
```

```
summ(eden$MANSA)
```




```
##  obs. mean   median  s.d.   min.   max.
##  642  4.508  4.55    0.939  1.1    6.8
```