Analyze Your Data with R: A complete guide.

Contents

1 Introduction

1.1 What is `tableone` package in *R*?

- `tableone` is an *R* package that helps you create "Table 1" *(making summary statistics)*, as well as performing the basic needed statistical tests.

1.2 Features of `tableone` Package:

- Handles both continuous and categorical variables.
- Categorical variables are summarized as counts (how many) and/or percentages.
- Continuous variables can be summarized in two ways:
  - Normal way: Shows the mean (average) and standard deviation (how much values vary).
  - Non-normal way: Shows the median (middle value) and interquartile range (range of the middle 50% of values, excluding first and last quartiles).

2 Getting into it

2.1 What Will You Need?

- To effectively utilize the *tableone* package in *R* for data analysis, the following tools and software are recommended:
  - **RStudio**: An integrated development environment (IDE) for *R* that facilitates code writing, package usage, and data visualization.
  - ***tableone* Package** itself.
  - **Spreadsheet Software**: Such as **Microsoft Excel** or any equivalent alternative, to export, review, and refine the generated tables.
  - **Word Processing Software**: Such as **Microsoft Word** or any similar application, to compile and finalize the analysis report, including formatted tables and interpretations.

2.2 How to Setup the Environment?

- Follow these steps to set up your environment:
  - **Install RStudio**: If you haven't already, download and install RStudio.
  - **Install Required Packages**: Once RStudio is installed and running, you'll need to install the necessary packages:
    - The *tableone* package is essential for creating summary tables and finding p-values.
    - It is recommended to also install the *tidyverse* library.

```r
install.packages("tableone") # Installing the `tableone` package
install.packages("tidyverse") # Installing the `tidyverse` library

# Activating the packages
library(tableone)
library(tidyverse)
```

2.3 What about the Workflow?

1. **Import Your Data**: Usually from Excel file.
2. **Create Summary Tables**: Utilize the *tableone* package to perform descriptive and inferential statistics.
3. **Export Tables**: Once your tables are ready, export each one to an Excel file.
4. **Integrate with Your Document**: Finally, copy the tables from Excel and paste them into your Word document.

Detailed instructions for each step will be provided in the next sections.

3 Usage

3.1 Importing the Raw Data from Excel File

- You can import the data file using RStudio interface:
  - Import Dataset *in the upper right corner* => From Excel
- You can import it using r code:

```
my_data <- readxl::read_excel("my_data.xlsx")
# Storing the data in a variable called "my_data"



# NOTE: Some concepts you need to know
#
#   1. A variable: you can think of a variable as the pocket that will contain data
# during any process.
#
#   2. A function: you can think of a function as the machine that will take your data
# and operate on it, giving you an output based on the given data. The syntax of a
# function is: my_function(parameter, another_parameter, ..., another_parameters)
#
#   3. A parameter: It is a piece of information you pass to a function, it can be the
# data that the function will operate on, or it can be some instruction.
```

Here is a preview of the data we will work on during this tutorial:

```
## # A tibble: 62 x 5
##       ID Group Weight Height    Hb
##    <dbl> <dbl>  <dbl>  <dbl> <dbl>
##  1     1     1     87    160    16
##  2     2     1     90    165    15
##  3     3     1     85    166    14
##  4     4     1     90    172    13
##  5     5     2     68    177     7
##  6     6     2     56    172    10
##  7     7     1     90    183    12
##  8     8     2     71    185     6
##  9     9     2     71    157     7
## 10    10     1     94    166    13
## # i 52 more rows
```

3.2 Something to Put in Mind: *categorical variable conversion*

- Usually, categorical variables are coded numerically *(when working with R, it is better to leave it uncoded).*
- Our package can't recognize this, So you have to include a list *(vector)* with all the categorical variables of your data in a parameter *(instruction)* while using the package. *Will be clarified in the next section*

3.3 Making Summary Statistics *(Descriptive Statistics)*

- Use the function ***'CreateTableOne()'***

3.3.1 Simplest Use Case: treating the whole data as one group

- In this example, all columns of your data will be included.

```
tableone::CreateTableOne(data = my_data)
```

```
##
##                       Overall
##   n                      62
##   ID (mean (SD))      31.50 (18.04)
##   Group (mean (SD))    1.60 (0.49)
##   Weight (mean (SD))  74.50 (12.10)
##   Height (mean (SD)) 172.85 (10.27)
##   Hb (mean (SD))      10.82 (3.42)
```

```
# Here, we are using the function `CreateTableOne()` with the parameter "data" to
# give it our data, which is stored in the variable "my_data"
```

3.3.1.1 Choosing Only Some Variables to Be Included

- Here, we will add the parameter "vars":

```
## Vector of variables to summarize
included_vars <- c("Group", "Weight", "Height", "Hb") # Excluded the ID variable

## Create a TableOne
my_summary_table <- tableone::CreateTableOne(data = my_data, vars = included_vars)
# "vars" parameter determines the variables included in the summary.

print(my_summary_table) # The function "print" shows the table
```

```
##
##                       Overall
##   n                      62
##   Group (mean (SD))    1.60 (0.49)
##   Weight (mean (SD))  74.50 (12.10)
##   Height (mean (SD)) 172.85 (10.27)
##   Hb (mean (SD))      10.82 (3.42)
```

| 3.3.1.2 Categorical Variable Conversion: Define categorical variables

- Here, we will add the parameter "factorVars" to determine which variables are categorical:

```
## Vector of variables to summarize
included_vars <- c("Group", "Weight", "Height", "Hb") # Excluded the ID variable

## Vector of categorical variables that need transformation
categorical_vars <- c("Group")

## Create a TableOne
my_summary_table <- tableone::CreateTableOne(
  data = my_data,
  vars = included_vars,
  factorVars = categorical_vars
  )
# "factorVars" parameter determines the variables that should be treated as categorical.

print(my_summary_table) # The function "print" shows the table
```

```
##
##                      Overall
##   n                     62
##   Group = 2 (%)         37 (59.7)
##   Weight (mean (SD))  74.50 (12.10)
##   Height (mean (SD)) 172.85 (10.27)
##   Hb (mean (SD))      10.82 (3.42)
```

3.3.2 Showing all levels for categorical variables

- If you want to show all levels, you can use "showAllLevels" argument to the print() method in order to show all categories of the included categorical variables:

```
## Vector of variables to summarize
included_vars <- c("Group", "Weight", "Height", "Hb") # Excluded the ID variable

## Vector of categorical variables that need transformation
categorical_vars <- c("Group")

## Create a TableOne
my_summary_table <- tableone::CreateTableOne(
  data = my_data,
  vars = included_vars,
  factorVars = categorical_vars
  )

# Now we are using some new parameters for the print function
print(my_summary_table, showAllLevels = TRUE, formatOptions = list(big.mark = ","))
```

```
##
##                       level Overall
##   n                             62
##   Group (%)           1         25 (40.3)
##                       2         37 (59.7)
##   Weight (mean (SD))          74.50 (12.10)
##   Height (mean (SD))         172.85 (10.27)
##   Hb (mean (SD))              10.82 (3.42)
```

3.3.3 Grouping the Data: dividing the data into groups according to given categorical variable

- Here, we will add the parameter "strata" and assign to it the categorical variable (grouping factor) by which the data will be grouped.
- You can also see that the package had perfomed the proper statistical test to find p-value of difference between groups of the given grouping factor.

```
## Vector of variables to summarize
included_vars <- c("Weight", "Height", "Hb") # Excluded the ID variable

## Vector of categorical variables that need transformation
categorical_vars <- c("Group")

## Create a TableOne
my_summary_table <- tableone::CreateTableOne(
  data = my_data,
  vars = included_vars,
  factorVars = categorical_vars,
  strata = "Group"
  )

# Now we are using some new parameters for the print function
print(my_summary_table, showAllLevels = TRUE, formatOptions = list(big.mark = ","))
```

```
##                      Stratified by Group
##                       level 1             2              p      test
##   n                                 25            37
##   Weight (mean (SD))             87.04 (5.70)  66.03 (6.59)  <0.001
##   Height (mean (SD))            171.52 (9.01) 173.76 (11.07)  0.405
##   Hb (mean (SD))                 14.56 (1.58)   8.30 (1.37)  <0.001
```

3.3.4 Summarizing Nonnormal Variables

- Do it with the nonnormal argument to the print() method:

```
## Vector of variables to summarize
included_vars <- c("Weight", "Height", "Hb") # Excluded the ID variable

## Vector of categorical variables that need transformation
categorical_vars <- c("Group")

# Lets assume that the height is not normally distributed
nonnormal_vars <- c("Height")

## Create a TableOne
my_summary_table <- tableone::CreateTableOne(
  data = my_data,
  vars = included_vars,
  factorVars = categorical_vars,
  strata = "Group"
  )

# Now we are using some new parameters for the print function
print(
  my_summary_table,
  nonnormal = nonnormal_vars,
  showAllLevels = TRUE,
  formatOptions = list(big.mark = ","))
```

```
##                         Stratified by Group
##                          level 1                       2
##   n                                25                      37
##   Weight (mean (SD))            87.04 (5.70)            66.03 (6.59)
##   Height (median [IQR])        170.00 [165.00, 175.00] 175.00 [164.00, 184.00]
##   Hb (mean (SD))                14.56 (1.58)             8.30 (1.37)
##                         Stratified by Group
##                          p      test
##   n
##   Weight (mean (SD))     <0.001
##   Height (median [IQR])   0.351 nonnorm
##   Hb (mean (SD))         <0.001
```

3.3.5 Detailed information including missingness

- If you need more detailed information including the number/proportion missing. Use the summary() method on the result object.
- It also shows the p-values in both cases of normality.
- The continuous variables are shown first, and the categorical variables are shown second.
- Unfortunately, our data doesn't contain categorical variables other than the grouping one, but if you try it with another data, it will work.

```
summary(my_summary_table)
```

```
##
##      ### Summary of continuous variables ###
##
## Group: 1
##         n miss p.miss mean sd median p25 p75 min max  skew kurt
## Weight 25    0      0   87  6     89  84  91  76  94 -0.72 -0.7
## Height 25    0      0  172  9    170 165 175 158 189  0.58 -0.6
## Hb     25    0      0   15  2     15  13  16  12  17 -0.09 -1.0
## ------------------------------------------------------------
## Group: 2
##         n miss p.miss mean sd median p25 p75 min max skew kurt
## Weight 37    0      0   66  7     67  60  71  55  78 -0.2   -1
## Height 37    0      0  174 11    175 164 184 155 190 -0.3   -1
## Hb     37    0      0    8  1      9   7   9   6  10 -0.4   -1
##
## p-values
##             pNormal   pNonNormal
## Weight 4.335796e-19 4.312439e-11
## Height 4.047534e-01 3.505207e-01
## Hb     4.612746e-24 2.390141e-11
##
## Standardize mean differences
##           1 vs 2
## Weight 3.4116154
## Height 0.2216052
## Hb     4.2280597
```

3.3.6 Categorical or Continuous Variables Only

- To get the categorical part only of the previous methods, you can use the following code before using "print":

```
# We don't have categorical varaibles, but let's try the syntax!
my_summary_table <- my_summary_table$CatTable
```

- To get the continuous part only of the previous methods, you can use the following code before using "print":

```
my_summary_table <- my_summary_table$ContTable
```

3.4 Testing for Difference *(Finding the p-Value)*

- As you can see in the previous table, when there are two or more groups group comparison p-values are printed along with the table.
- The hypothesis test functions used by default are:
  - chisq.test() for categorical variables.
  - oneway.test() for continous variables (with equal variance assumption, i.e., regular ANOVA).
- You may be worried about the nonnormal variables and small cell counts in the stage variable.
- In such a situation, you can use the nonnormal argument like before as well as the exact (test) argument in the print() method.
  - Now kruskal.test() is used for the nonnormal continuous variables and fisher.test() is used for categorical variables specified in the exact argument.

```
## Vector of variables to summarize
included_vars <- c("Weight", "Height", "Hb") # Excluded the ID variable

## Vector of categorical variables that need transformation
categorical_vars <- c("Group")

# Lets assume that the height is not normally distributed
nonnormal_vars <- c("Height")

## Create a TableOne
my_summary_table <- tableone::CreateTableOne(
  data = my_data,
  vars = included_vars,
  factorVars = categorical_vars,
  strata = "Group"
  )

# You test for the Standardize mean differences using the "smd" parameter
print(my_summary_table, nonnormal = nonnormal_vars, smd = TRUE)
```

```
##                         Stratified by Group
##                          1                        2                        p
##   n                          25                       37
##   Weight (mean (SD))      87.04 (5.70)             66.03 (6.59)            <0.001
##   Height (median [IQR]) 170.00 [165.00, 175.00] 175.00 [164.00, 184.00]  0.351
##   Hb (mean (SD))          14.56 (1.58)              8.30 (1.37)            <0.001
##                         Stratified by Group
##                          test    SMD
##   n
##   Weight (mean (SD))             3.412
##   Height (median [IQR]) nonnorm  0.222
##   Hb (mean (SD))                 4.228
```

3.5 Exporting

- The following code exports the final table, which you made using the previous methods, into CSV file *(A type of data files, it is also supported by Microsoft Excel)*:

```
write.csv(tab, file = "myTable.csv")
# Save to a CSV file
```

4 Summary

In this tutorial, we explored the process of generating and customizing summary statistics tables using the *tableone* package, with particular attention to the "`CreateTableOne()`" function. Below is a brief summary of the key aspects and modifications discussed:

- **Variable Selection and Stratification**:
  - You can specify variables to include using the "`vars`" argument.
  - You can stratify the table by categorical variables with the "`strata`" argument.
- **Showing All Levels for Categorical Variables**:
  - By adding 'showAllLevels = TRUE' to the print method.
- **Handling Non-Normal Distributions**:
  - The "`nonnormal`" argument allows the presentation of median and interquartile ranges (IQR) for non-normally distributed continuous variables.
- **Defining Factor Variables**:
  - The "`factorVars`" argument ensures proper handling of categorical data.
- **P-Value Calculation and Customization**:
  - p-values are automatically calculated using the proper tests when stratifying the data.

This summary encapsulates the key modifications and parameters available in the "`CreateTableOne()`" function, offering a concise reference for efficiently generating and tailoring summary tables in your research work.

5 Example: Place your parameters and run this code

```r
# Ultimate Guide: analyze your data in simple steps
# ======================================================

# Importing the file
my_data <- readxl::read_excel("my file destination/my file.xlsx")

# Determining the included variables
included_vars <- c("included variable name 1", "included variable name n")
# Determining the nonnormal variables
nonnormal_vars <- c("nonnormal variable name 1", "nonnormal variable name n")
# Determining the categorical variables
categorical_vars <- c("categorical variable name 1", "categorical variable name n")
# Determining the grouping variable
grouping_variable <- "grouping variable name"

# Making the analysis
analysis_table <- CreateTableOne(
  data = my_data,
  vars = included_vars,
  factorVars = categorical_vars,
  strata = grouping_variable
)
final_table <- print(analysis_table,
      showAllLevels = TRUE,
      nonnormal = nonnormal_vars,
      formatOptions = list(big.mark = ","))

# Exporting the final_table to a CSV file
write.csv(final_table, file = "Data Analysis Results.csv")

# ======================================================
# Congrats! You've finished the analysis.
```