TableOne Documentation

Release 0.7.12

Tom J. Pollard, Alistair E. W. Johnson

Aug 22, 2022

Getting Started

`tableone` is a package for easily producing summary measures of a dataset for use in publications. See *the quickstart* to get started.

CHAPTER 1

Quickstart

1.1 Install

```
$ pip install tableone
```

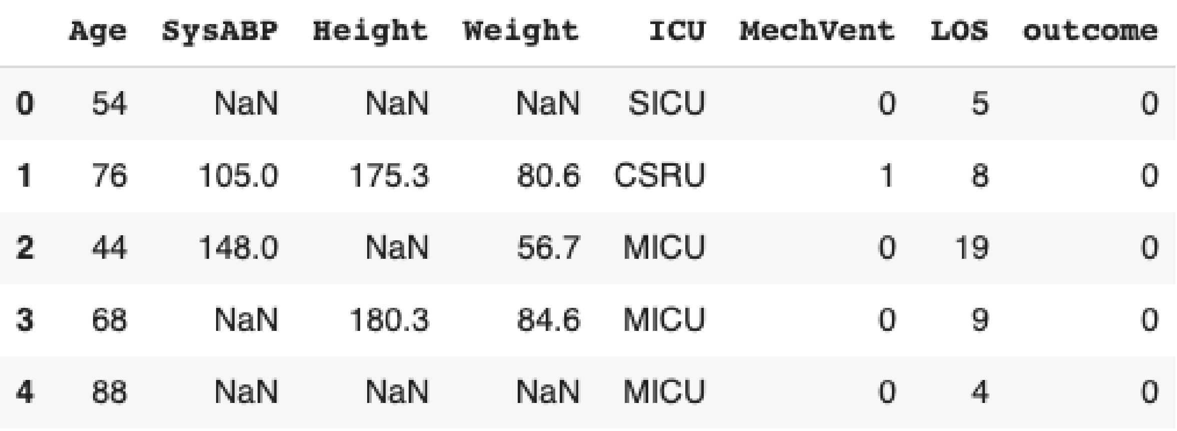
See *installation* document for more information.

1.2 Run demo

The easiest way to understand what this package does is to evaluate it on data.

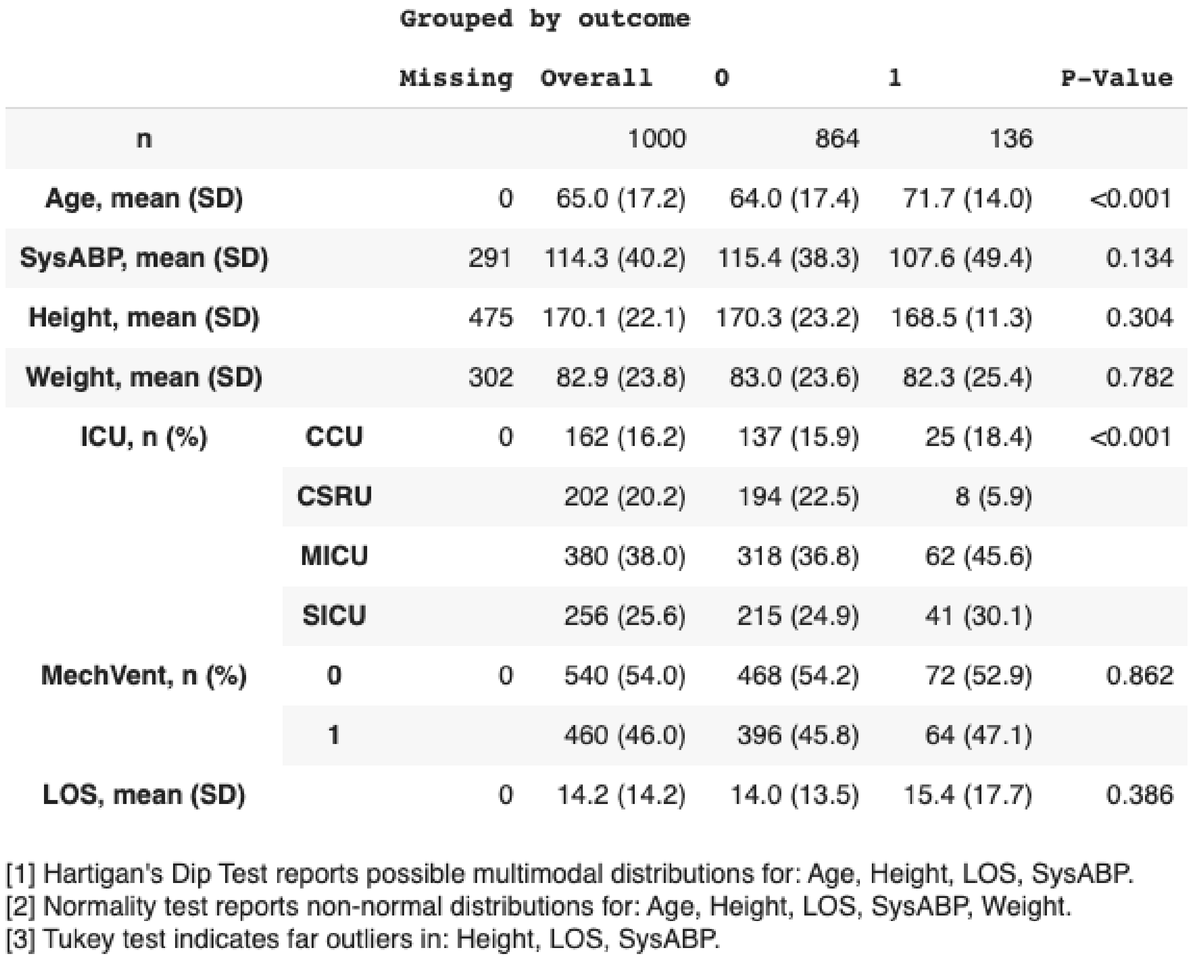
This Google Colaboratory Notebook is an executable document that allows you to explore *tableone*. At a high level, you can use the package as follows:

- Import the data into a pandas DataFrame



	Age	SysABP	Height	Weight	ICU	MechVent	LOS	outcome
0	54	NaN	NaN	NaN	SICU	0	5	0
1	76	105.0	175.3	80.6	CSRU	1	8	0
2	44	148.0	NaN	56.7	MICU	0	19	0
3	68	NaN	180.3	84.6	MICU	0	9	0
4	88	NaN	NaN	NaN	MICU	0	4	0

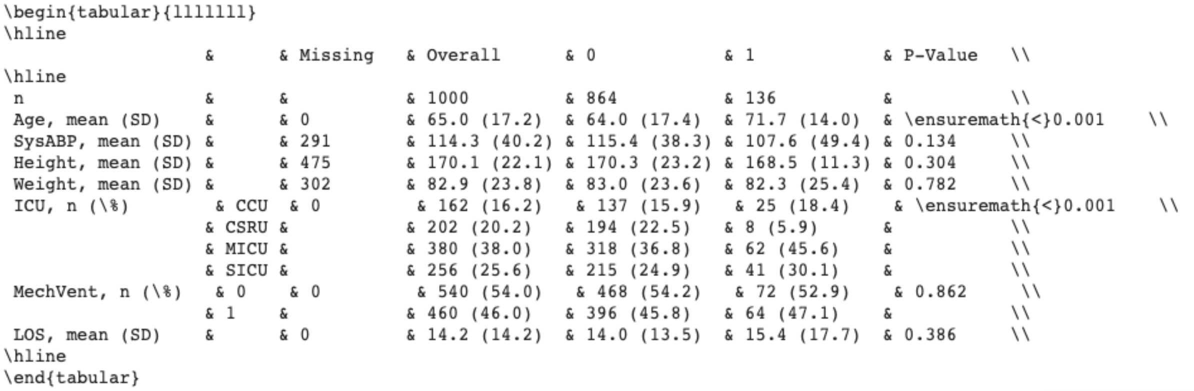
- Run tableone on this dataframe to output summary statistics



		Grouped by outcome				
		Missing	Overall	0	1	P-Value
n			1000	864	136	
Age, mean (SD)		0	65.0 (17.2)	64.0 (17.4)	71.7 (14.0)	<0.001
SysABP, mean (SD)		291	114.3 (40.2)	115.4 (38.3)	107.6 (49.4)	0.134
Height, mean (SD)		475	170.1 (22.1)	170.3 (23.2)	168.5 (11.3)	0.304
Weight, mean (SD)		302	82.9 (23.8)	83.0 (23.6)	82.3 (25.4)	0.782
ICU, n (%)	CCU	0	162 (16.2)	137 (15.9)	25 (18.4)	<0.001
	CSRU		202 (20.2)	194 (22.5)	8 (5.9)	
	MICU		380 (38.0)	318 (36.8)	62 (45.6)	
	SICU		256 (25.6)	215 (24.9)	41 (30.1)	
MechVent, n (%)	0	0	540 (54.0)	468 (54.2)	72 (52.9)	0.862
	1		460 (46.0)	396 (45.8)	64 (47.1)	
LOS, mean (SD)		0	14.2 (14.2)	14.0 (13.5)	15.4 (17.7)	0.386

[1] Hartigan's Dip Test reports possible multimodal distributions for: Age, Height, LOS, SysABP.
[2] Normality test reports non-normal distributions for: Age, Height, LOS, SysABP, Weight.
[3] Tukey test indicates far outliers in: Height, LOS, SysABP.

- Specify your desired output format: text, latex, markdown, etc.



```
\begin{tabular}{lllllll}
\hline
                  &       & Missing   & Overall      & 0            & 1            & P-Value   \\
\hline
 n                &       &           & 1000         & 864          & 136          &           \\
 Age, mean (SD)   &       & 0         & 65.0 (17.2)  & 64.0 (17.4)  & 71.7 (14.0)  & \ensuremath{<}0.001    \\
 SysABP, mean (SD) &      & 291       & 114.3 (40.2) & 115.4 (38.3) & 107.6 (49.4) & 0.134     \\
 Height, mean (SD) &      & 475       & 170.1 (22.1) & 170.3 (23.2) & 168.5 (11.3) & 0.304     \\
 Weight, mean (SD) &      & 302       & 82.9 (23.8)  & 83.0 (23.6)  & 82.3 (25.4)  & 0.782     \\
 ICU, n (\%)       & CCU  & 0          & 162 (16.2)   & 137 (15.9)   & 25 (18.4)    & \ensuremath{<}0.001    \\
                  & CSRU &           & 202 (20.2)   & 194 (22.5)   & 8 (5.9)      &           \\
                  & MICU &           & 380 (38.0)   & 318 (36.8)   & 62 (45.6)    &           \\
                  & SICU &           & 256 (25.6)   & 215 (24.9)   & 41 (30.1)    &           \\
 MechVent, n (\%)  & 0    & 0          & 540 (54.0)   & 468 (54.2)   & 72 (52.9)    & 0.862     \\
                  & 1    &           & 460 (46.0)   & 396 (45.8)   & 64 (47.1)    &           \\
 LOS, mean (SD)   &       & 0         & 14.2 (14.2)  & 14.0 (13.5)  & 15.4 (17.7)  & 0.386     \\
\hline
\end{tabular}
```

Additional options include:

- Select a subset of columns.
- Specify the data type (e.g. *categorical*, *numerical*, *nonnormal*).
- Compute p-values, and adjust for multiple testing (e.g. with the Bonferroni correction).

- Compute standardized mean differences (SMDs).
- Provide a list of alternative labels for variables
- Limit the output of categorical variables to the top N rows.
- Display remarks relating to the appopriateness of summary measures (for example, computing tests for multimodality and normality).

1.3 Suggested citation

If you use tableone in your study, please cite the following paper:

```
Tom J Pollard, Alistair E W Johnson, Jesse D Raffa, Roger G Mark;
tableone: An open source Python package for producing summary statistics
for research papers, JAMIA Open, Volume 1, Issue 1, 1 July 2018, Pages 26–31,
https://doi.org/10.1093/jamiaopen/ooy012
```

Download the BibTex file from: https://academic.oup.com/jamiaopen/downloadcitation/5001910?format=bibtex

1.4 Example

1. Import libraries:

```
from tableone import TableOne
import pandas as pd
```

2. Load sample data into a pandas dataframe:

```
url="https://raw.githubusercontent.com/tompollard/data/master/primary-biliary-
↪cirrhosis/pbc.csv"
data=pd.read_csv(url)
```

3. Optionally, a list of columns to be included in Table 1:

```
columns = ['age','bili','albumin','ast','platelet','protime',
       'ascites','hepato','spiders','edema','sex', 'trt']
```

4. Optionally, a list of columns containing categorical variables:

```
categorical = ['ascites','hepato','edema','sex','spiders','trt']
```

5. Optionally, a categorical variable for stratification and a list of non-normal variables:

```
groupby = 'trt'
nonnormal = ['bili']
```

6. Create an instance of TableOne with the input arguments:

```
mytable = TableOne(data, columns, categorical, groupby, nonnormal)
```

7. Display the table using the `tabulate` method. The `tablefmt` argument allows the table to be displayed in multiple formats, including “github”, “grid”, “fancy_grid”, “rst”, “html”, and “latex”.:

```
print(mytable.tabulate(tablefmt="github"))
```

8. Compute p values by setting the `pval` argument to *True*:

```
mytable = TableOne(data, columns, categorical, groupby, nonnormal, pval=True)
```

9. Tables can be exported to file in various formats, including LaTeX, CSV, and HTML. Files are exported by calling the `to_format` method on the DataFrame. For example, mytable can be exported to an Excel spreadsheet named 'mytable.tex' with the following command:

```
mytable.to_latex('mytable.tex')
```

CHAPTER 2

Best Practice

We recommend seeking guidance from a statistician when using `tableone` for a research study, especially prior to submitting the study for publication. It is beyond the scope of this documentation to provide detailed guidance on summary statistics, but as a primer we provide some considerations for choosing parameters when creating a summary table.

2.1 Data visualization

Plotting the distribution of each variable by group level via histograms, kernel density estimates and boxplots is a crucial component to data analysis pipelines. Vizualisation is often is the only way to detect problematic variables in many real-life scenarios. Some example plots are provided in the tableone notebook.

2.2 Normally distributed variables

Variables not listed in the *nonnormal* argument will be summarised by their mean and standard deviation. The mean and standard deviation are often poor estimates of the center or dispersion of a variable's distribution when the distribution: is asymmetric, has 'fat' tails and/or outliers, contains only a very small finite set of values or is multi-modal. Although formal statistical tests are available to detect most of these features, they often are not very useful in small sample sizes[1].

For normally distributed variables, both estimation and hypothesis testing (provided the standard deviations of each group are the same) are more efficient when the variable is not set in the *nonnormal* argument[2][3] . This may also hold in some circumstances where the data are clearly not normally distributed, provided the sample sizes are large enough. In other situations, assuming normality when the data is not normally distributed can lead to inefficient or spurious inference.

[1] Mohd Razali, Nornadiah & Yap, Bee. (2011). "Power Comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling Tests". Journal of statistical modeling and analytics, volume 2, pp21-33

[2] Zimmerman, D. (1987). "Comparative Power of Student T Test and Mann-Whitney U Test for Unequal Sample Sizes and Variances". The Journal of Experimental Education, 55(3), 171-174.

[3] Hodges, J., & Lehmann, E. (1956). "The Efficiency of Some Nonparametric Competitors of the t-Test". The Annals of Mathematical Statistics, 27(2), 324-335.

2.3 Non-normally distributed variables

For numeric variables, including integer and floating point values in addition to some ordered discrete variables, the *nonnormal* argument of TableOne merits some discussion. The practical consequence of including a variable in the *nonnormal* argument is to rely on rank based methods for estimation of the center and variability of the distribution for the relevant variable, along with non-parametric methods to conduct hypothesis testing evaluating if the distributions of all the groups are the same[4][5]. Median and interquartile range may offer a more robust summary than mean and standard deviation for skewed distributions or in the presence of outliers, but may be misleading in cases such as multimodality.

2.4 Comparison of estimates

To supplement data visualization, you may choose to compare two `TableOne` tables created from the same dataset: firstly with all numeric variables in the *nonnormal* argument, and subsequently with none of the variables in *nonnormal* argument. Then one can focus on situations where:

- substantial differences exist between the mean and median estimates
- the median or mean is not well centered between the first and third quartiles[6]
- large differences exist between the absolute differences in the first and third quartile and the standard deviation, understanding that the interquartile range will be about 35% larger than the standard deviation under normality

A particular situation to note is when the number of groups specified in the *groupby* argument are three or more and the group variances differ to a large degree. Under such a situation it may be preferable to consider the data as non-normal, even if each group's data were generated from a normal distribution[7], particularly when the group sizes are unequal or the sample sizes are small.

When the number of groups are limited to two, this is addressed using Welch's two sample t-test which is generally both efficient and robust under unequal variances between two groups[8]. A similar type of test exists for one-way ANOVA[9], but is currently not implemented.

2.5 Alternatives to consider

Thus far we have suggested methods which vary estimation and hypothesis testing techniques when a normality assumption is not appropriate. Alternatives do exist which may be more practical to your situation. In many circumstances transforming the variable can reduce the influence of asymmetry or other features of the distribution. Under monotone transformations (e.g., logarithm or square root for strictly positive number) this should have little impact on any variable which is included in the *nonnormal* argument, as these methods will typically be invariant to this class of transformation.

[4] Lehmann, Erich L and D'Abrera, Howard JM (1975). "Nonparametrics: statistical methods based on ranks". Oxford, England: Holden-Day.

[5] Conover, W., & Iman, R. (1981). "Rank Transformations as a Bridge Between Parametric and Nonparametric Statistics". The American Statistician, 35(3), 124-129. doi:10.2307/2683975

[6] Altman, D., & Bland, J. (1996). "Detecting Skewness From Summary Information. BMJ: British Medical Journal". 313(7066), 1200-1200.

[7] Boneau, C. A. (1960). "The effects of violations of assumptions underlying the t test". Psychological Bulletin, 57(1), 49-64. http://dx.doi.org/10.1037/h0041412

[8] Welch Bernard L (1947). "The generalization of 'Student's' problem when several different population varlances are involved". Biometrika, Volume 34, Issue 1-2, 1 January 1947, Pages 28–35, https://doi.org/10.1093/biomet/34.1-2.28

[9] Weerahandi, Samaradasa (1995). "ANOVA under Unequal Error Variances." Biometrics, vol. 51, no. 2, 1995, pp. 589–599.

2.6 Multiple testing

If multiple hypotheses are tested, as is commonly the case when numerous variables are summarised in a table, the chance of a rare event increases. As a result, the likelihood of incorrectly rejecting a null hypothesis (i.e., making a Type I error) increases. By default, `tableone` computes the Bonferroni correction to account for multiple testing. This correction addresses the problem of multiple comparisons in a simple way, by dividing the prespecified significance level (Type I error rate,) by the number of hypothesis tests conducted.

The Bonferroni correction is known to over-correct, effectively reducing the statistical power of the tests, particularly when the number of hypotheses are large or when the tests are positively correlated. There are many alternatives which may be more suitable and also widely used, and which should be considered in situations that would be adversely affected by the conservative nature of the Bonferroni correction[10][11][12].

2.7 Summary

It should be noted that while we have tried to use best practices, automation of even basic statistical tasks can be unsound if done without supervision. We encourage use of `tableone` alongside other methods of descriptive statistics and, in particular, visualization to ensure appropriate data handling.

[10] Benjamini, Yoav & Hochberg, Yosef (1995). “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. Journal of the Royal Statistical Society, Series B. 57 (1): 125–133.

[11] Holm, S. (1979). “A simple sequentially rejective multiple test procedure”. Scandinavian Journal of Statistics. 6 (2): 65–70.

[12] Šidák, Z. K. (1967). “Rectangular Confidence Regions for the Means of Multivariate Normal Distributions”. Journal of the American Statistical Association. 62 (318): 626–633.

CHAPTER 3

Install TableOne

You can install tableone with `conda`, with `pip`, or by installing from source.

3.1 Conda

To install the latest version of tableone from the conda-forge repository using conda:

```
conda install tableone -c conda-forge
```

3.2 Pip

Or install tableone with `pip`:

```
pip install tableone
```

3.3 Source

To install distributed from source, clone the repository from github:

```
git clone https://github.com/tompollard/tableone.git
cd tableone
python setup.py install
```

CHAPTER 4

Contributing to TableOne

We welcome all contributions to the package!

Table of contents:

4.1 Where to start?

Bug reports, bug fixes, documentation improvements, and other contributions are welcome. For reporting bugs or suggesting improvements, please use the GitHub issues tab.

4.2 Bug reports

Bug reports are core to ensuring the package remains useful for all users. A complete bug report greatly improves the ability of others to understand and fix it. For information on how to make a complete bug report, we recommend you review this helpful StackOverflow article.

4.3 Contributing improvements

Bug fixes or other enhancements are welcome via pull requests. You can read more about pull requests on GitHub's website.

4.4 Contributing to the documentation

Rewriting small pieces of the documentation as you read through it is a surefire way of improving them for the next user.

4.4.1 About the documentation

The documentation is written in *reStructuredText*, and subsequently built using the Python package Sphinx. The Sphinx documentation provides a gentle introduction to reStructuredText.

The documentation follows the NumPy Docstring Standard, which are parsed using the *napolean extension for sphinx <http://www.sphinx-doc.org/en/1.5.1/ext/napoleon.html>*.

4.4.2 How to build the documentation

Requirements

To build the documentation you will need to additionally install `sphinx`. Furthermore, you'll also need to install the readthedocs theme. This is easily done using pip:

```
pip install sphinx sphinx_rtd_theme
```

Building the documentation

Navigate to the `docs` subfolder and run:

```
sphinx-build -b html . _build
```

Which will build the documentation in the subfolder `_build`. Alternatively, you can run the Makefile provided:

```
make html
```

CHAPTER 5

tableone

5.1 TableOne