STHDA

Statistical tools for high-throughput data analysis

Search...

Home	Basics	Data	Visualize	Analyze	Resources	Our Products

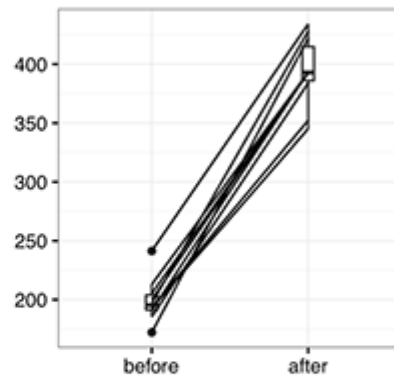| Support	About

⚙ Actions menu for module Wiki

📡 Paired Samples Wilcoxon Test in R

≡ Tools

- Visualize your data and compute paired samples Wilcoxon test in R
  - R function
  - Import your data into R
  - Check your data
  - Visualize your data using box plots
  - Compute paired-sample Wilcoxon test
- Online paired-sample Wilcoxon test calculator
- See also
- Infos

The **paired samples Wilcoxon test** (also known as **Wilcoxon signed-rank test**) is a **non-parametric** alternative to paired t-test used to compare paired data. It's used when your data are not normally distributed. This tutorial describes how to compute paired samples Wilcoxon test in **R**.

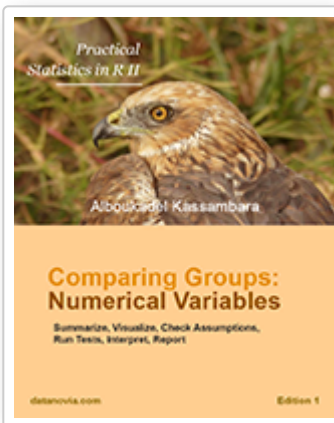❌ Differences between paired samples should be distributed symmetrically around the median.

Visualize your data and compute paired samples Wilcoxon test in R

R function

The R function **wilcox.test**() can be used as follow:

```
wilcox.test(x, y, paired = TRUE, alternative = "two.sided")
```

- **x,y**: numeric vectors
- **paired**: a logical value specifying that we want to compute a paired Wilcoxon test
- **alternative**: the alternative hypothesis. Allowed value is one of "two.sided" (default), "greater" or "less".

Import your data into R

1. **Prepare your data** as specified here: Best practices for preparing your data set for R

2. **Save your data** in an external .txt tab or .csv files

3. **Import your data into R** as follow:

```
# If .txt tab file, use this
my_data <- read.delim(file.choose())
# Or, if .csv file, use this
my_data <- read.csv(file.choose())
```

Here, we'll use an example data set, which contains the weight of 10 mice before and after the treatment.

```
# Data in two numeric vectors
# ++++++++++++++++++++++++++++
# Weight of the mice before treatment
before <-c(200.1, 190.9, 192.7, 213, 241.4, 196.9, 172.2, 185.5, 205.2, 193.7)
# Weight of the mice after treatment
after <-c(392.9, 393.2, 345.1, 393, 434, 427.9, 422, 383.9, 392.3, 352.2)
# Create a data frame
my_data <- data.frame(
                group = rep(c("before", "after"), each = 10),
                weight = c(before,  after)
                )
```

? We want to know, if there is any significant difference in the median weights before and after treatment?

Check your data

```r
# Print all data
print(my_data)
```

```
   group weight
1  before  200.1
2  before  190.9
3  before  192.7
4  before  213.0
5  before  241.4
6  before  196.9
7  before  172.2
8  before  185.5
9  before  205.2
10 before  193.7
11  after  392.9
12  after  393.2
13  after  345.1
14  after  393.0
15  after  434.0
16  after  427.9
17  after  422.0
18  after  383.9
19  after  392.3
20  after  352.2
```

Compute summary statistics (median and inter-quartile range (IQR)) by groups using the dplyr package can be used.

- Install **dplyr** package:

```r
install.packages("dplyr")
```

- Compute summary statistics by groups:

```r
library("dplyr")
group_by(my_data, group) %>%
  summarise(
    count = n(),
    median = median(weight, na.rm = TRUE),
    IQR = IQR(weight, na.rm = TRUE)
  )
```

```
Source: local data frame [2 x 4]
  group count median    IQR
  (fctr) (int)  (dbl)  (dbl)
```

```
1  after     10 392.95 28.800
2 before     10 195.30 12.575
```

Visualize your data using box plots

- To use R base graphs read this: R base graphs. Here, we'll use the **ggpubr** R package for an easy ggplot2-based data visualization.

- Install the latest version of ggpubr from GitHub as follow (recommended):

```
# Install
if(!require(devtools)) install.packages("devtools")
devtools::install_github("kassambara/ggpubr")
```

- Or, install from CRAN as follow:

```
install.packages("ggpubr")
```

- Visualize your data:

```
# Plot weight by group and color by group
library("ggpubr")
ggboxplot(my_data, x = "group", y = "weight",
          color = "group", palette = c("#00AFBB", "#E7B800"),
          order = c("before", "after"),
          ylab = "Weight", xlab = "Groups")
```
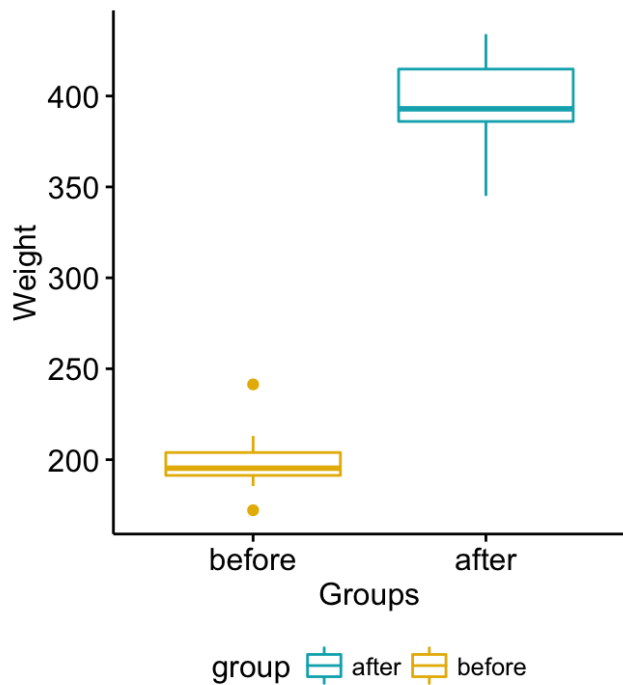
> ⚠ Box plots show you the increase, but lose the paired information. You can use the function **plot.paired**() [in **pairedData** package] to plot paired data ("before - after" plot).

- Install pairedData package:

```
install.packages("PairedData")
```

- Plot paired data:

```
# Subset weight data before treatment
before <- subset(my_data,  group == "before", weight,
                  drop = TRUE)
# subset weight data after treatment
after <- subset(my_data,  group == "after", weight,
                  drop = TRUE)
# Plot paired data
library(PairedData)
pd <- paired(before, after)
plot(pd, type = "profile") + theme_bw()
```
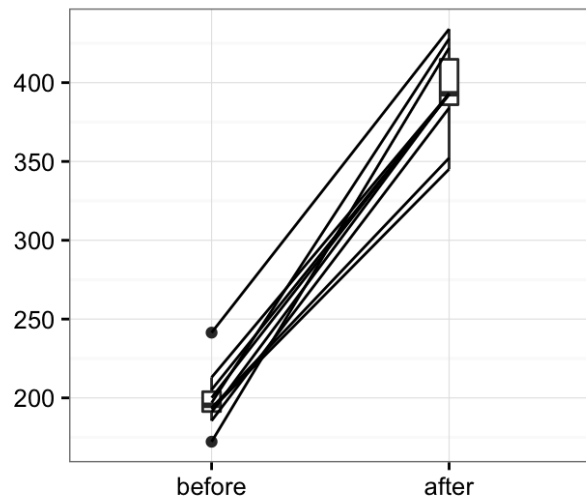
Compute paired-sample Wilcoxon test

> ❓ Question : Is there any significant changes in the weights of mice before after treatment?

1) Compute paired Wilcoxon test - Method 1: The data are saved in two different numeric vectors.

```
res <- wilcox.test(before, after, paired = TRUE)
res
```

```
	Wilcoxon signed rank test
data:  before and after
V = 0, p-value = 0.001953
alternative hypothesis: true location shift is not equal to 0
```

2) Compute paired Wilcoxon-test - Method 2: The data are saved in a data frame.

```
# Compute t-test
res <- wilcox.test(weight ~ group, data = my_data, paired = TRUE)
res
```

```
	Wilcoxon signed rank test
data:  weight by group
V = 55, p-value = 0.001953
alternative hypothesis: true location shift is not equal to 0
```

```
# print only the p-value
res$p.value
```

```
[1] 0.001953125
```

> **!** As you can see, the two methods give the same results.

> **✓** The **p-value** of the test is 0.001953, which is less than the significance level alpha = 0.05. We can conclude that the median weight of the mice before treatment is significantly different from the median weight after treatment with a **p-value** = 0.001953.

> **!** Note that:
>
> - if you want to test whether the median weight before treatment is less than the median weight after treatment, type this:
>
> ```
> wilcox.test(weight ~ group, data = my_data, paired = TRUE,
>         alternative = "less")
> ```
>
> - Or, if you want to test whether the median weight before treatment is greater than the median weight after treatment, type this
>
> ```
> wilcox.test(weight ~ group, data = my_data, paired = TRUE,
>         alternative = "greater")
> ```

Online paired-sample Wilcoxon test calculator

You can perform **paired-sample Wilcoxon test**, **online**, without any installation by clicking the following link:
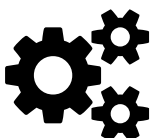
Online paired-sample Wilcoxon test calculator

See also

[Paired Samples T-test (parametric)](#)

Infos

> ⚠️ This analysis has been performed using **R software** (ver. 3.2.4).

> ✅ Enjoyed this article? I'd be very grateful if you'd help it spread by emailing it to a friend, or sharing it on Twitter, Facebook or Linked In.
>
> Show me some love with the like buttons below... Thank you and please don't forget to share and comment below!!
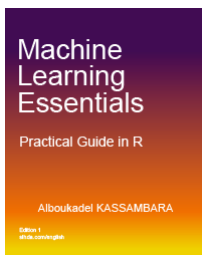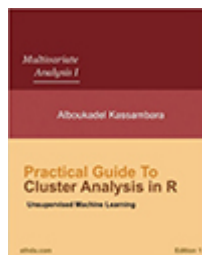
Recommended for You!

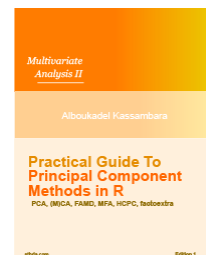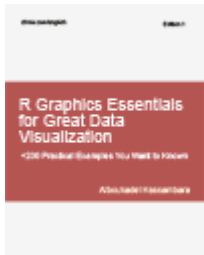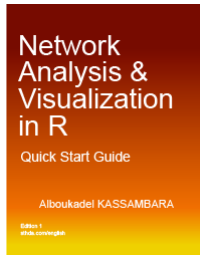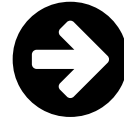|		
Machine Learning Essentials: Practical Guide in R	Practical Guide to Cluster Analysis in R	Practical Guide to Principal Component Methods in R

[R Graphics Essentials for Great Data Visualization](#)

[Network Analysis and Visualization in R](#)

[More books on R and data science](#)

Recommended for you

✔️ This section contains best data science and self-development resources to help you on your path.

Coursera - Online Courses and Specialization
Data science

- [Course: Machine Learning: Master the Fundamentals](#) by Standford
- [Specialization: Data Science](#) by Johns Hopkins University
- [Specialization: Python for Everybody](#) by University of Michigan
- [Courses: Build Skills for a Top Job in any Industry](#) by Coursera
- [Specialization: Master Machine Learning Fundamentals](#) by University of Washington
- [Specialization: Statistics with R](#) by Duke University
- [Specialization: Software Development in R](#) by Johns Hopkins University
- [Specialization: Genomic Data Science](#) by Johns Hopkins University

Popular Courses Launched in 2020

- [Google IT Automation with Python](#) by Google
- [AI for Medicine](#) by deeplearning.ai
- [Epidemiology in Public Health Practice](#) by Johns Hopkins University
- [AWS Fundamentals](#) by Amazon Web Services

Trending Courses

- [The Science of Well-Being](#) by Yale University
- [Google IT Support Professional](#) by Google
- [Python for Everybody](#) by University of Michigan
- [IBM Data Science Professional Certificate](#) by IBM
- [Business Foundations](#) by University of Pennsylvania
- [Introduction to Psychology](#) by Yale University
- [Excel Skills for Business](#) by Macquarie University
- [Psychological First Aid](#) by Johns Hopkins University
- [Graphic Design](#) by Cal Arts

Books - Data Science
Our Books

- [Practical Guide to Cluster Analysis in R](#) by A. Kassambara (Datanovia)
- [Practical Guide To Principal Component Methods in R](#) by A. Kassambara (Datanovia)

- [Machine Learning Essentials: Practical Guide in R](#) by A. Kassambara (Datanovia)
- [R Graphics Essentials for Great Data Visualization](#) by A. Kassambara (Datanovia)
- [GGPlot2 Essentials for Great Data Visualization in R](#) by A. Kassambara (Datanovia)
- [Network Analysis and Visualization in R](#) by A. Kassambara (Datanovia)
- [Practical Statistics in R for Comparing Groups: Numerical Variables](#) by A. Kassambara (Datanovia)
- [Inter-Rater Reliability Essentials: Practical Guide in R](#) by A. Kassambara (Datanovia)

Others

- [R for Data Science: Import, Tidy, Transform, Visualize, and Model Data](#) by Hadley Wickham & Garrett Grolemund
- [Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems](#) by Aurelien Géron
- [Practical Statistics for Data Scientists: 50 Essential Concepts](#) by Peter Bruce & Andrew Bruce
- [Hands-On Programming with R: Write Your Own Functions And Simulations](#) by Garrett Grolemund & Hadley Wickham
- [An Introduction to Statistical Learning: with Applications in R](#) by Gareth James et al.
- [Deep Learning with R](#) by François Chollet & J.J. Allaire
- [Deep Learning with Python](#) by François Chollet

Want to Learn More on R Programming and Data Science?

Follow us [by Email](#)

Subscribe
by [FeedBurner](#)

On Social Networks:
on Social Networks

Get involved :

Click to **follow us** on [Facebook](#) and [Google+](#) :

Comment this article by clicking on "Discussion" button (top-right position of this page)

Sign in

Login

Login

Password

Password

Auto connect

☑

Sign in

Welcome!

Want to Learn More on R Programming and Data Science?
Follow us by Email

Subscribe
by FeedBurner

on Social Networks

Click to see our collection of resources to help you on your path...

Course & Specialization

Recommended for You (on Coursera):

- Course: Machine Learning: Master the Fundamentals
- Specialization: Data Science
- Specialization: Python for Everybody
- Course: Build Skills for a Top Job in any Industry
- Specialization: Master Machine Learning Fundamentals
- Specialization: Statistics with R
- Specialization: Software Development in R
- Specialization: Genomic Data Science

See More Resources

- 📋 **factoextra**
- 📋 **survminer**
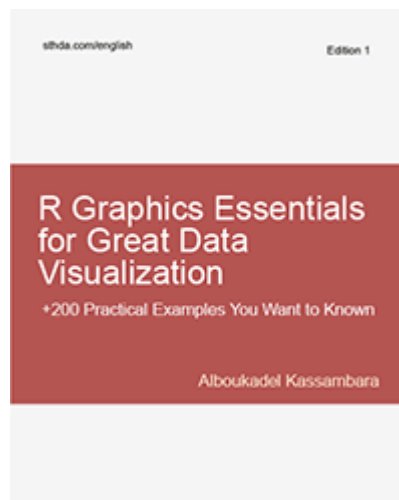- 📋 **ggpubr**
- 📋 **ggcorrplot**
- 📋 **fastqcr**

Our Books

3D Plots in R

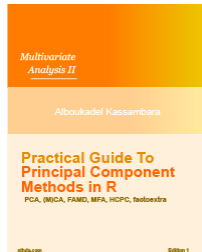### R Graphics Essentials for Great Data Visualization: 200 Practical Examples You Want to Know for Data Science
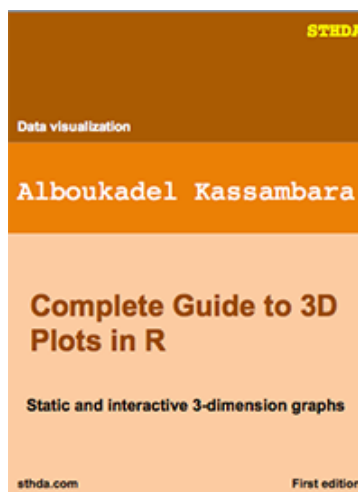⭐ NEW!!

[Practical Guide to Cluster Analysis in R](#)

[Practical Guide to Principal Component Methods in R](#)

🌐 **Datanovia: Online Data Science Courses**

🌐 **R-Bloggers**

Newsletter Email ✉

Recommended for you

ggplot2 title : main, axis and legend titles - Easy Guides - Wiki - STHDA

www.sthda.com

ggplot2 colors : How to change colors automatically and ma...

www.sthda.com

Reading Data From Excel Files (xls|xlsx) into R - Easy Guides - Wiki...

www.sthda.com

AddThis