




---

# Kruskal–Wallis Test

---

## When to use it

See the [Handbook](#) for information on this topic.

## Null hypothesis

This example shows just summary statistics, histograms by group, and the Kruskal–Wallis test. An example with plots, post-hoc tests, and alternative tests is shown in the “Example” section below.

### *Kruskal–Wallis test example*

```
### -----
### kruskal-wallis test, hypothetical example, p. 159
### -----
```

```
Input =(
```

```
"Group      value
Group.1      1
Group.1      2
Group.1      3
Group.1      4
Group.1      5
Group.1      6
Group.1      7
Group.1      8
Group.1      9
Group.1     46
Group.1     47
Group.1     48
Group.1     49
Group.1     50
Group.1     51
Group.1     52
Group.1     53
Group.1    342
Group.2     10
Group.2     11
Group.2     12
Group.2     13
Group.2     14
Group.2     15
Group.2     16
Group.2     17
Group.2     18
Group.2     37
Group.2     58
Group.2     59
Group.2     60
Group.2     61
Group.2     62
Group.2     63
Group.2     64
Group.2    193
Group.3     19
Group.3     20
Group.3     21
Group.3     22
```

```

Group.3 23
Group.3 24
Group.3 25
Group.3 26
Group.3 27
Group.3 28
Group.3 65
Group.3 66
Group.3 67
Group.3 68
Group.3 69
Group.3 70
Group.3 71
Group.3 72
")
Data = read.table(textConnection(Input),header=TRUE)

### Specify the order of factor levels
library(dplyr)

Data =
mutate(Data,
      Group = factor(Group, levels=unique(Group))
)

```

### Medians and descriptive statistics

As noted in the *Handbook*, each group has identical medians and means.

```

library(psych)
describeBy(Data$value,
            group=Data$Group,
            type=2)
# is type of skew and kurtosis

```

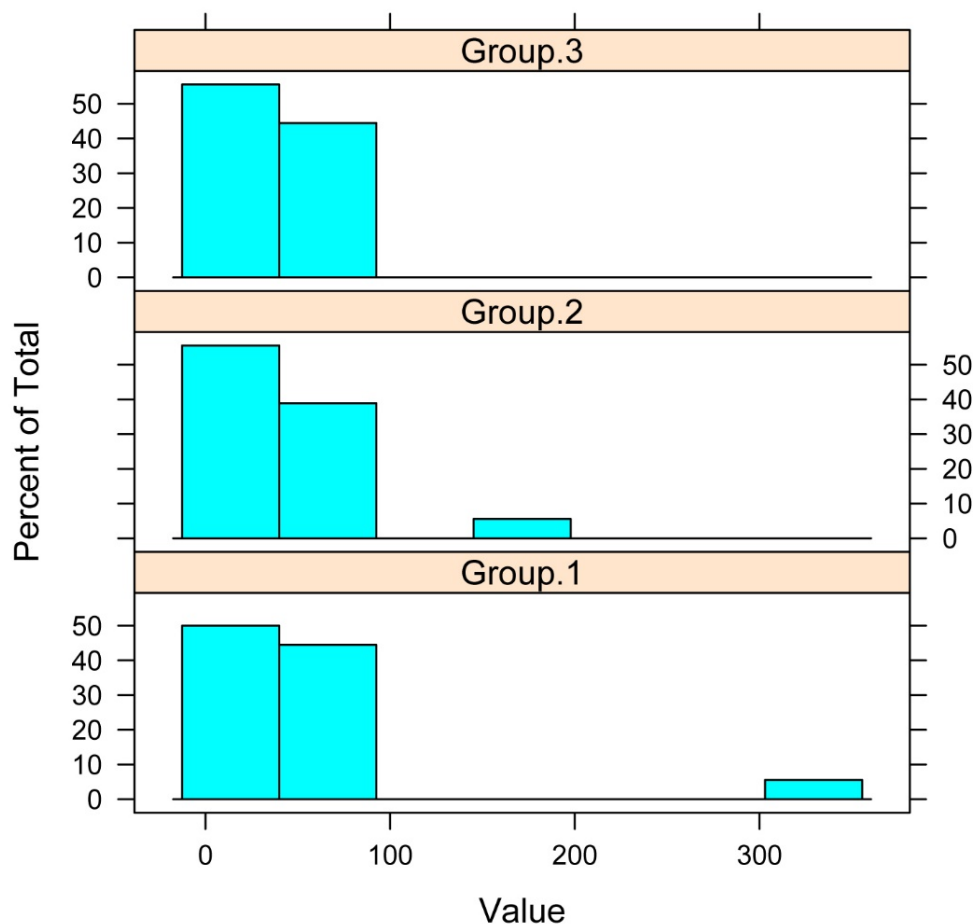
group:	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
1	1	18	43.5	77.78	27.5	27.5	33.36	1	342	341	3.67	14.62	18.33
-----													
group:	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
1	1	18	43.5	43.69	27.5	36.25	25.2	10	193	183	2.49	8.06	10.3
-----													
group:	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
1	1	18	43.5	23.17	27.5	43.25	11.86	19	72	53	0.23	-2.13	5.46

### Histograms for each group

```

library(lattice)
histogram(~ Value | Group,
          data=Data,
          layout=c(1,3)
)
# columns and rows of individual plots

```



### Kruskal–Wallis test

In this case, there is a significant difference in the distributions of values among groups, as is evident both from the histograms and from the significant Kruskal–Wallis test. Only in cases where the distributions are similar can a significant Kruskal–Wallis test be interpreted as a difference in medians.

```
kruskal.test(Value ~ Group,
             data = Data)
```

```
kruskal-wallis chi-squared = 7.3553, df = 2, p-value = 0.02528
```

```
# # #
```

## How the test works

### Assumptions

See the *Handbook* for information on these topics.

### Example

The Kruskal–Wallis test is performed on a data frame with the *kruskal.test* function in the native *stats* package. Shown first is a complete example with plots, post-hoc tests, and alternative methods, for the example used in R help. It is data measuring if the mucociliary efficiency in the rate of dust removal is different among normal subjects, subjects with obstructive airway disease, and subjects with asbestosis. For the original citation, use the *?kruskal.test* command. For both the submissive dog example and the oyster DNA example from the *Handbook*, a Kruskal–Wallis test is shown later in this chapter.

### *Kruskal–Wallis test example*

```

### -----
### kruskal-wallis test, asbestosis example from R help for
### kruskal.test
### -----

Input =(
"obs Health      Efficiency
1  Normal      2.9
2  Normal      3.0
3  Normal      2.5
4  Normal      2.6
5  Normal      3.2
6  OAD         3.8
7  OAD         2.7
8  OAD         4.0
9  OAD         2.4
10 Asbestosis  2.8
11 Asbestosis  3.4
12 Asbestosis  3.7
13 Asbestosis  2.2
14 Asbestosis  2.0
")

Data = read.table(textConnection(Input),header=TRUE)

### Specify the order of factor levels

library(dplyr)
Data =
mutate(Data,
      Health = factor(Health, levels=unique(Health))
)

```

### Medians and descriptive statistics

```

library(psych)
describeBy(Data$Efficiency,
           group=Data$Health,
           type=2)

```

# is type of skew and kurtosis

```

group: Normal
  vars n mean    sd median trimmed  mad min max range  skew kurtosis  se
1    1 5 2.84 0.29   2.9   2.84 0.44 2.5 3.2  0.7 -0.04   -1.8 0.13
-----
group: OAD
  vars n mean    sd median trimmed  mad min max range  skew kurtosis  se
1    1 4 3.23 0.79   3.25   3.23 0.96 2.4  4  1.6 -0.07   -5 0.4
-----
group: Asbestosis
  vars n mean    sd median trimmed  mad min max range  skew kurtosis  se
1    1 5 2.82 0.74   2.8   2.82 0.89  2 3.7  1.7 0.09   -2.41 0.33

```

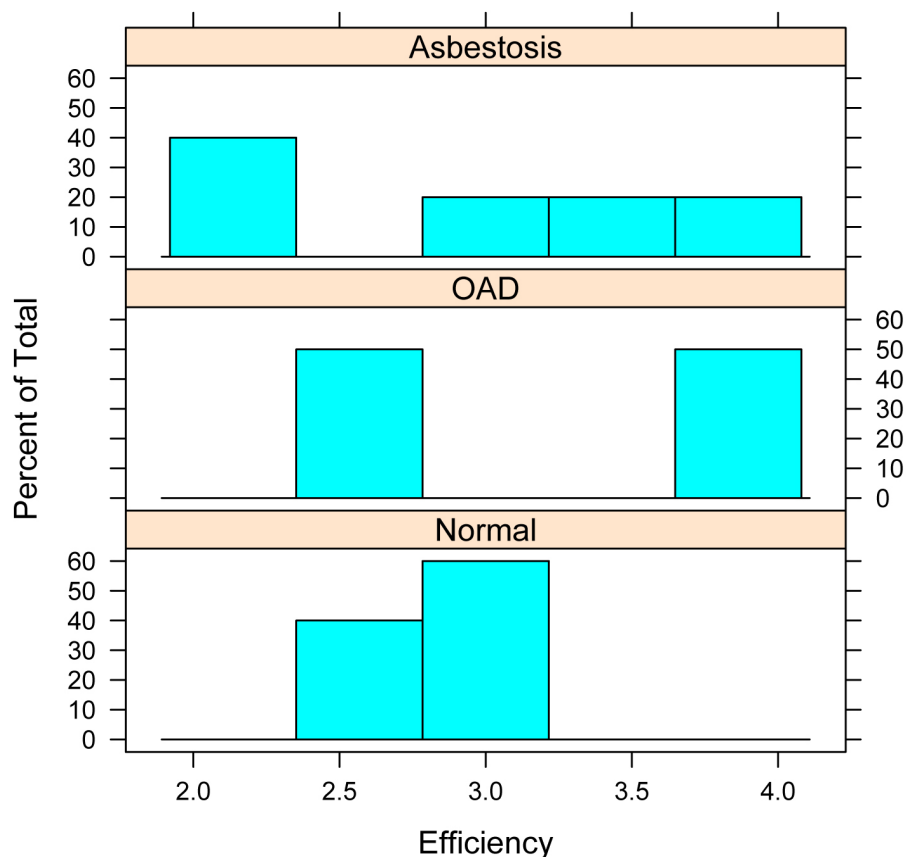
### ***Graphing the results***

#### Stacked histograms of values across groups

```

library(lattice)
histogram(~ Efficiency | Health,
         data=Data,
         layout=c(1,3)      # columns and rows of individual plots
)

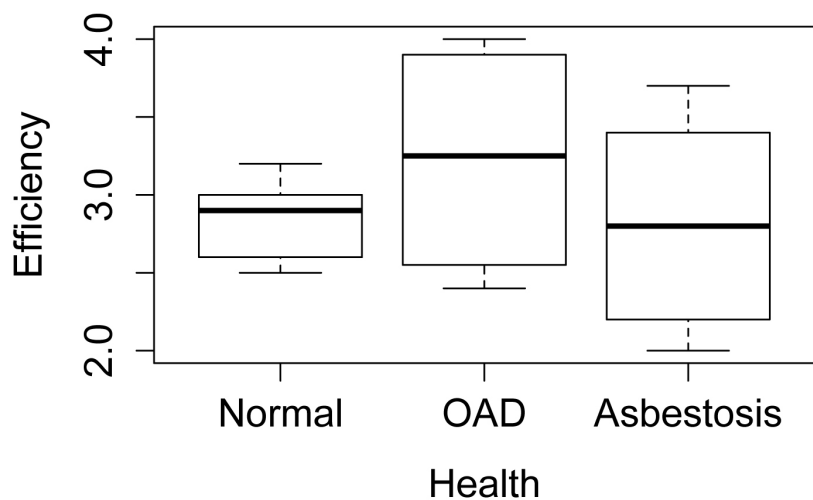
```



Stacked histograms for each group in a Kruskal–Wallis test. If the distributions are similar, then the Kruskal–Wallis test will test for a difference in medians.

### Simple boxplots of values across groups

```
boxplot(Efficiency ~ Health,
        data = Data,
        ylab="Efficiency",
        xlab="Health")
```



### Kruskal–Wallis test

```
kruskal.test(Efficiency ~ Health,
             data = Data)
```

kruskal-wallis chi-squared = 0.7714, df = 2, p-value = 0.68

### ***Dunn test for multiple comparisons***

If the Kruskal–Wallis test is significant, a post-hoc analysis can be performed to determine which levels of the independent variable differ from each other level. Probably the most popular test for this is the Dunn test, which is performed with the *dunnTest* function in the *FSA* package, or with the *DunnTest* function in *DescTools*. Adjustments to the p-values could be made using the *method* option to control the familywise error rate or to control the false discovery rate. See *?p.adjust* for details.

Be cautious: at the time of writing, *DunnTest* in *DescTools* reports one-sided p-values, which are usually not what is desired. The *dunnTest* function in *FSA* is therefore preferred.

Zar (2010) states that the Dunn test is appropriate for groups with unequal numbers of observations.

```
library(FSA)
dunnTest(Efficiency ~ Health,
         data=Data,
         method="fdr")      # Can adjust p-values;
                           # See ?p.adjust for options

Dunn (1964) Kruskal-wallis multiple comparison
p-values adjusted with the False Discovery Rate method.
```

	Comparison	Z	P.unadj	P.adj
1	OAD-Normal=0	0.6414270	0.5212453	0.7818680
2	Asbestosis-Normal=0	-0.2267787	0.8205958	0.8205958
3	Asbestosis-OAD=0	-0.8552360	0.3924205	0.7818680

```
library(DescTools)
DunnTest(x = Data$Efficiency,
         g = Data$Health,
         method="fdr")      # Can adjust p-values;
                           # See ?p.adjust for options

Dunn's test of multiple comparisons using rank sums : fdr
```

	mean.rank.diff	pval
OAD-Normal	1.8	0.3909
Asbestosis-Normal	-0.6	0.4103
Asbestosis-OAD	-2.4	0.3909

```
### Note that these p-values are one-sided values.
### For two-sided p-values, they would need to be doubled.
```

### ***Nemenyi test for multiple comparisons***

Zar (2010) suggests that the Nemenyi test is not appropriate for groups with unequal numbers of observations.

```
library(DescTools)
NemenyiTest(x = Data$Efficiency,
            g = Data$Health,
            dist="tukey")

Nemenyi's test of multiple comparisons for independent samples (tukey)
```

	mean.rank.diff	pval
OAD-Normal	1.8	0.7972
Asbestosis-Normal	-0.6	0.9720
Asbestosis-OAD	-2.4	0.6686

```
library(PMCMR)
posthoc.kruskal.nemenyi.test(Data$Efficiency,
                             Data$Health,
                             method = "Tukey")
```

Pairwise comparisons using Tukey and Kramer (Nemenyi) test  
with Tukey-Dist approximation for independent samples

	Normal	OAD
OAD	0.80	-
Asbestosis	0.97	0.67

P value adjustment method: none      #    Is original Tukey-Kramer method  
#    that controls family-wise error

### Pairwise Mann–Whitney U-tests

Another post-hoc approach is to use pairwise Mann–Whitney U-tests. To prevent the inflation of type I error rates, adjustments to the p-values can be made using the *p.adjust.method* option to control the familywise error rate or to control the false discovery rate. See *?p.adjust* for details.

```
pairwise.wilcox.test(Data$Efficiency,
                    Data$Health,
                    p.adjust.method="none")
# Can adjust p-values;
# See ?p.adjust for options
```

Pairwise comparisons using wilcoxon rank sum test

	Normal	OAD
OAD	0.73	-
Asbestosis	1.00	0.41

### Compact letter display from lower triangle results

It is common for pairwise methods in R to display the results as a table of p-values, with the lower triangle of p-values filled in, as in the case of the *pairwise.wilcox.test* function. If there are several values to compare, it can be beneficial to have R convert this table to a compact letter display for you. The *multcompLetters* function in the *multcompView* package can do this, but first the table of p-values must be converted to a full table.

PT is the p-value table output for some test

```
PT = pairwise.wilcox.test(Data$Efficiency,
                        Data$Health,
                        p.adjust.method="none")$p.value
# Can adjust p-values;
# See ?p.adjust for options
```

PT

	Normal	OAD
OAD	0.7301587	NA
Asbestosis	1.0000000	0.4126984

Convert PT to a full table and call it PT1

```
source("http://rcompanion.org/r_script/full.p.table.r")
PT1 = full.p.table(PT)
PT1
```

	Normal	OAD	Asbestosis
Normal	1.0000000	0.7301587	1.0000000
OAD	0.7301587	1.0000000	0.4126984
Asbestosis	1.0000000	0.4126984	1.0000000

Produce compact letter display

```
library(multcompview)
multcompLetters(PT1,
                compare="<",
                ...)
```

```

threshold=0.05,
Letters=letters,
reversed = FALSE)

Normal      OAD Asbestos
"a"         "a"  "a"

### values sharing the same letter are not significantly different

#      #      #

```

### ***Kruskal–Wallis test example***

```

### -----
### kruskal-wallis test, submissive dog example, pp. 161–162
### -----

Input =(

"Dog      Sex      Rank
Merlino   Male      1
Gastone   Male      2
Pippo     Male      3
Leon     Male      4
Golia     Male      5
Lancillotto Male      6
Mamy       Female     7
Nana       Female     8
Isotta     Female     9
Diana      Female    10
Simba      Male      11
Pongo      Male      12
Semola     Male      13
Kimba      Male      14
Morgana    Female    15
Stella     Female    16
Hansel     Male      17
Cucciola   Male      18
Mammolo    Male      19
Dotto      Male      20
Gongolo    Male      21
Gretel     Female    22
Brontolo   Female    23
Eolo       Female    24
Mag        Female    25
Emy        Female    26
Pisola     Female    27

")

Data = read.table(textConnection(Input),header=TRUE)
kruskal.test(Rank ~ Sex,
              data = Data)

kruskal-wallis chi-squared = 4.6095, df = 1, p-value = 0.03179

#      #      #

```

## **Graphing the results**

Graphing of the results is shown above in the “Example” section.

## **Similar tests**

*One-way anova* is presented elsewhere in this book.

## **How to do the test**

### ***Kruskal–Wallis test example***

```

### -----
### kruskal-wallis test, oyster DNA example, pp. 163–164

```



###

Input =(

```
"Markername  Markertype  fst
CVB1         DNA         -0.005
CVB2m        DNA          0.116
CVJ5         DNA         -0.006
CVJ6         DNA          0.095
CVL1         DNA          0.053
CVL3         DNA          0.003
6Pgd         protein       -0.005
Aat-2        protein        0.016
Acp-3        protein        0.041
Adk-1        protein        0.016
Ap-1         protein        0.066
Est-1        protein        0.163
Est-3        protein        0.004
Lap-1        protein        0.049
Lap-2        protein        0.006
Mpi-2        protein        0.058
Pgi          protein       -0.002
Pgm-1        protein        0.015
Pgm-2        protein        0.044
Sdh          protein        0.024
```

")

```
Data = read.table(textConnection(Input),header=TRUE)
```

```
kruskal.test(fst ~ Markertype,
             data = Data)
```

```
kruskal-wallis chi-squared = 0.0426, df = 1, p-value = 0.8365
```

```
#      #      #
```

## Power Analysis

See the *Handbook* for information on this topic.

## References

Zar, J.H. 2010. Biostatistical Analysis, 5th ed. Pearson Prentice Hall: Upper Saddle River, NJ.

©2015 by Salvatore S. Mangiafico, except for organization of statistical tests and selection of examples for these tests  
©2014 by John H. McDonald. Used with permission.

Non-commercial reproduction of this content, with attribution, is permitted. For-profit reproduction without permission is prohibited.

If you use the code or information in this site in a published work, please cite it as a source. Also, if you are an instructor and use this book in your course, please let me know. My contact information is on the *About the Author* page.

Mangiafico, S.S. 2015. *An R Companion for the Handbook of Biological Statistics*, version 1.09.  
[rcompanion.org/rcompanion/](http://rcompanion.org/rcompanion/). (Pdf version: [rcompanion.org/documents/RCompanionBioStatistics.pdf](http://rcompanion.org/documents/RCompanionBioStatistics.pdf).)