Kruskal Wallis Test Calculator

Followed by post-hoc Dunn's test

Kruskal Wallis calculator with multiple comparisons, effect size, test power, outliers, and R syntax.

Significance level (α): 0.05

Outliers: Included

Effect size (offsets): 0.3

Correction Method: Bonferroni

Multiple comparisons method: Dunn's

Digits: 4

Step by step

- Enter raw data directly
- Enter raw data from excel

Compare	How to implement	List	Elaborate	How to design
3	3	5	3	3
3	3	5	3	4
2	3	5	5	3
4	4	5	5	5
2	3	3	3	3
2	3	3	4	5
4	3	2	5	2
2	2	2	4	3
3	4	2	3	2
2	2	3	5	3
3	2	3	5	4
2	2	2	5	3
4	4	2	4	4
3	2	3	4	3
2	3	3	5	4

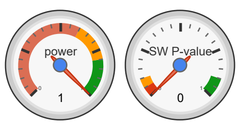
Header: you may rename 'Group1', 'Group2', etc.
Data: use Enter or , (comma) or space as delimiters.
The tool ignores empty cells, non-numeric cells, or empty columns.

Calculate | Insert column | Delete column | Clear | Load last run | Load example

Reporting results in APA style

The Kruskal-Wallis H test indicated that there is a significant difference in the *dependent variable* between the different *groups*, $\chi^2(4) = 22.79$, $p < .001$, with a mean rank score of 166.81 for Compare, 157.66 for How to implement, 190.92 for List, 224.48 for Elaborate, 225.77 for How to design.
The Post-Hoc Dunn's test using a Bonferroni corrected alpha of 0.005 indicated that the mean ranks of the following pairs are significantly different: **x_1-x_4 x_1-x_5 x_2-x_4 x_2-x_5**

How to do with R?



Groups:	Compare	How to implement	List	Elaborate	How to design
Skewness:	0.4202	0.2498	0.1819	-0.05692	-0.324
Skewness Shape:	Potentially	Potentially	Potentially	Potentially	Potentially
Excess kurtosis:	-0.1115	-0.4689	-0.6526	-0.7018	-0.6528
Tails Shape:	Potentially	Potentially	Potentially	Potentially	Potentially
Normality	0.00007311	0.000007183	0.000003383	1.255e-9	1.596e-8
Outliers:					
Median:	3	3	3	4	4
Sample size (n):	40	51	71	146	101
Rank sum (R):	6672.5	8040.5	13555	32774	22803
R^2/n:	1113056.406	1267640.005	2587859.507	7357089.562	5148285.238

Kruskal-Wallis-test, using Chi-Square(df:4) distribution (right-tailed)[Validation]

1. H_0 hypothesis

Since the p-value < α, H_0 is rejected.
Some of the groups' mean ranks consider to be not equal.
In other words, the difference between the mean ranks of some groups is big enough to be statistically significant.
When selecting a value from each of the groups, there are some groups with a higher probability of containing the highest value than others.

2. P-value

The p-value equals **0.0001397**, ($P(x \le 22.787) = 0.9999$). It means that the chance of type I error (rejecting a correct H_0) is small: 0.0001397 (0.014%). The smaller the p-value the more it supports H_1.

3. Test statistic

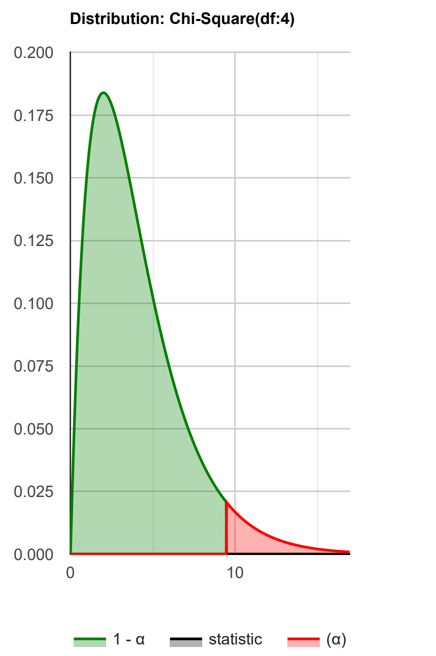
The test statistic **H** equals **22.787**, which is not in the 95% region of acceptance: [0, 9.4877].

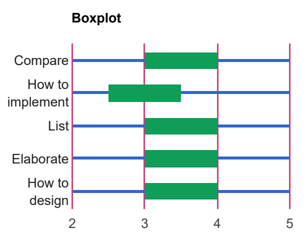
4. Effect size

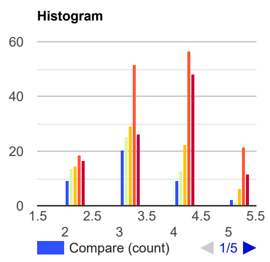
The observed effect size η^2 is **small**, **0.047**. This indicates that the magnitude of the difference between the average is small.

5. Multiple comparisons

The mean ranks of the following pairs are significantly different: x_1-x_4 x_1-x_5 x_2-x_4 x_2-x_5







Validation

Test power

The test priori power is strong **1**

Normality

The normality is **not** an assumption for the Kruskal-Wallis Test! We only check the normality to know if you could use a better test.
The normality was checked based on the Shapiro-Wilk Test. (α=0.05)
When running the SW test on the residuals, the p-value is 2.009e-9.
It is assumed that all the groups distribute normally or have a big sample size, at least 30.
It is valid to use the KW test on a normal data! But in this case you should consider the more powerfull ANOVA test calculator

Multiple comparisons

Compares any pair of groups using the Kruskal Wallis test. In this case, the test is identical to the Mann-Whitney U test with normal approximation.
If you won't correct the significance level (α) and all the comparisons are independent, then the type I error may get to 0.4013. $1 - (1- 0.05)^{10} = 0.4013$.
This is the worse case, since usually some aspects of the multiple tests are in common, and the type I error will be lower.
In this case the Bonferroni correction would be an over correction and would reduce the test power.
The corrected α using **Bonferroni** correction method is **0.005**.
m - the number of tests / pairs.
Corrected α = α / m = 0.05 / 10 = 0.005.

Pair	Mean Rank difference	Z	SE	Critical value	p-value	p-value/2
x_1-x_2	9.1556	0.3871	23.6495	66.3849	0.6987	0.3493

x_1-x_3	-24.103	1.0888	22.137	62.1392	0.2762	0.1381
x_1-x_4	-57.667	2.8858	19.9832	56.0936	0.003905	0.001952
x_1-x_5	-58.9598	2.8185	20.9187	58.7195	0.004825	0.002412
x_2-x_3	-33.2586	1.6182	20.5533	57.6938	0.1056	0.05281
x_2-x_4	-66.8226	3.6689	18.2133	51.1252	0.0002436	0.0001218
x_2-x_5	-68.1154	3.5412	19.235	53.9933	0.0003983	0.0001991
x_3-x_4	-33.564	2.0717	16.2009	45.4766	0.03829	0.01915
x_3-x_5	-34.8568	2.01	17.3416	48.6786	0.04443	0.02222
x_4-x_5	-1.2928	0.08921	14.492	40.6794	0.9289	0.4645

Group	How to implement	List	Elaborate	How to design
Compare	**9.16**	**-24.1**	**-57.67**	**-58.96**
How to implement	0	**-33.26**	**-66.82**	**-68.12**
List	**-33.26**	0	**-33.56**	**-34.86**
Elaborate	**-66.82**	**-33.56**	0	**-1.29**

Calculation

Ranks

Group	Value	Rank
Compare	2	**35.5**
Compare	2	**35.5**
Compare	2	**35.5**
Compare	2	**35.5**
Compare	2	**35.5**
Compare	2	**35.5**
Compare	2	**35.5**
Compare	2	**35.5**
Compare	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
How to implement	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
List	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**

Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
Elaborate	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
How to design	2	**35.5**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
Compare	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
How to implement	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**

List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
List	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**

Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
Elaborate	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
How to design	3	**146**
Compare	4	**295**
Compare	4	**295**
Compare	4	**295**
Compare	4	**295**
Compare	4	**295**
Compare	4	**295**
Compare	4	**295**
Compare	4	**295**
Compare	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
How to implement	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**

List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
List	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
Elaborate	4	**295**
How to design	4	**295**
How to design	4	**295**

How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
How to design	4	**295**
Compare	5	**389**
Compare	5	**389**
How to implement	5	**389**
List	5	**389**
List	5	**389**
List	5	**389**
List	5	**389**
List	5	**389**
List	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**

Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
Elaborate	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**
How to design	5	**389**

R_1 = 35.5+35.5+35.5+...+295+389+389= 6672.5.
R_2 = 35.5+35.5+35.5+...+295+295+389= 8040.5.
R_3 = 35.5+35.5+35.5+...+389+389+389= 13555.
R_4 = 35.5+35.5+35.5+...+389+389+389= 32774.
R_5 = 35.5+35.5+35.5+...+389+389+389= 22803.
$n = n_1 + n_2 + ... + n_k = 409$

$$H' = \frac{12}{n(n+1)}\left(\frac{R_1^2}{n_1}+\frac{R_2^2}{n_2}+...+\frac{R_k^2}{n_k}\right) - 3(n+1)$$

$$H' = \frac{12}{409(409+1)}\left(\frac{6672.5^2}{40}+\frac{8040.5^2}{51}+\frac{13555^2}{71}+\frac{32774^2}{146}+\frac{22803^2}{101}\right) - 3(409+1) = 20.4453$$

$$H = \frac{H'}{1 - 0.1028} = 22.787$$

$MeanRank_1$ = 6672.5 / 40 = 166.8125.
$MeanRank_2$ = 8040.5 / 51 = 157.6569.
$MeanRank_3$ = 13555 / 71 = 190.9155.
$MeanRank_4$ = 32774 / 146 = 224.4795.
$MeanRank_5$ = 22803 / 101 = 225.7723.

Kruskal Wallis Test

The Kruskal-Wallis test also called one-way ANOVA on ranks is a non-parametric test.
Use the Kruskal-Wallis test calculator when your data doesn't meet the assumptions of the [one-way ANOVA test calculator].

The kruskal wallis test online checks the null assumption that when selecting a value from each of 'n' groups, each of these groups will have an equal probability of containing the highest value.

Target: To check if the difference between the ranks of two or more groups is significant, using a sample data
When the groups have a similar distribution shape, the null assumption is stronger and states that the medians of the groups are equal. When performing the Kruskal Wallis test, we try to determine, if the difference between the ranks reflects a significant difference between the groups, or is due to the random noise inside each group. The Chi-square statistic is an approximation for the exact calculation.

Right-tailed the Kruskal Wallis test can use only the right tail. Why?

Hypotheses

$H_0: MR_1 = .. = MR_k$
$H_1: not(MR_1 = .. = MR_k)$

MR - Mean rank.

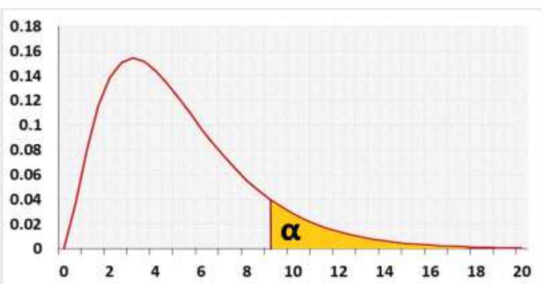
Test statistic

$$H' = \frac{12}{n(n+1)}\sum\left(\frac{R_j^2}{n_j}\right) - 3(n+1)$$

$$H = \frac{H'}{1 - correction}$$

R_j - the rank sum of group j.
n_j - the sample size of group j.
n - the total sample size across all groups, $n = n_1 + ... + n_j$.

χ^2 distribution



Assumptions

- Independent samples from independent groups. One subject can't be in more than one group.
- The dependent variable is ordinal variable or continuous variable
- Two or more groups (the independent variable is categorical variable with two or more values)

Required Sample Data

- Sample data from all compared groups.

Multiple comparisons

Even if we know that not all the ranks are equal, we don't know which groups are not equal, hence we run a Multiple comparisons test to compare all the pairs.
We support two methods for the Multiple comparisons:

- **Dunn's test calculator** - takes into consideration the total number of groups (k) even when comparing only two groups
- **Mann Whitney U test** - the calculator uses the normal approximation of the Mann Whitney U test and supports the same results as the Kruskal Wallis test with two groups.

R Code

The following R code should produce the same results

```
if(!"MultNonParam" %in% installed.packages()){install.packages("MultNonParam")}
library(MultNonParam)
if(!"dunn.test" %in% installed.packages()){install.packages("dunn.test")}
library(dunn.test)
x1 <- c(3,3,2,4,2,2,4,2,3,2,3,2,4,3,2,3,4,3,4,5,3,3,4,3,3,4,3,3,2,3,2,5,3,3,4,3,4,3,3,3)
x2 <- c(3,3,3,4,3,3,3,2,4,2,2,2,4,2,3,3,3,2,2,3,3,4,4,3,2,3,2,3,3,3,3,2,2,3,3,4,5,4,3,3,3,4,4,2,3,3,2,4,4,4,3)
x3 <- c(5,5,5,5,3,3,2,2,2,3,3,2,2,3,3,3,4,4,3,3,2,3,3,3,3,2,2,2,4,3,4,4,3,4,4,3,4,4,4,4,4,4,4,3,3,4,4,4,3,2,4,4,5,3,3,3,4,3,3,2,2,3,3,4,3,5,3,3,2,2,4)
x4 <-
c(3,3,5,5,3,4,5,4,3,5,5,5,4,4,5,4,2,3,4,3,4,3,5,2,3,4,4,3,3,4,2,2,3,3,2,5,4,4,4,3,3,4,4,4,4,3,2,3,2,4,3,4,3,3,3,4,3,4,4,4,4,4,2,2,4,3,4,4,4,4,3,3,5,3,3,4,5,4,3,3,4,3,2,3,4,4,4,4,3,4,4,3,4,2,2,4,3,3,4,4,4,4,3,5,5,4,4,2,4,3,3,4,5,3,2,2,2,3,4,3,3,3,3,3,4,3,
x5 <-
c(3,4,3,5,3,5,2,3,2,3,4,3,4,3,4,3,3,2,2,3,4,4,5,5,3,2,4,2,2,4,4,3,4,3,4,4,4,4,3,3,4,4,4,5,4,4,4,4,4,3,4,4,2,4,4,4,3,3,2,4,4,3,4,4,3,3,5,4,5,3,4,4,3,3,2,2,4,4,4,2,4,5,2,4,4,4,4,3,4,3,5,4,2,2,5,4,2,4,4,5,4)
list1=list(x1,x2,x3,x4,x5)
kruskal.test(list1)
dunn.test(list1)
nreps <- c(40,51,71,146,101)
shifts <- c(0,0.3,0.6,0.8999999999999999,1.2)
kwpower(nreps,shifts,'normal')
```

The dunn.test results show half of the p-values, that you should compare to half alpha.
This calculator results show full p-values.
When using corrections, the dunn.test corrects the p-value while this calculator corrects the alpha.