

Common statistical tests in R

Melbourne Statistical Consulting Platform

University of Melbourne

April 2024

Summary

These are some common statistical tests built into R. We assume you already know what these tests do and when to use them, and will show you how to do them in R.

Continuous data, two groups

`t.test()` for paired t-test, two sample t-test and one sample t-test.

`wilcox.test()` for Wilcoxon signed rank test (paired) and Wilcoxon rank sum test (two sample, a.k.a. Mann-Whitney test).

Continuous data, correlation

`cor.test()` for Pearson correlation, Spearman's rho and Kendall's tau.

Categorical data, association

`fisher.test()` for Fisher's Exact Test.

`chisq.test()` for Chi-squared Test.

Two sample t-test

```
epin <- read_csv("epin.csv")
```

```
glimpse(epin)
```

```
t.test(cholesterol ~ treatment,  
       data = epin,  
       paired = FALSE,  
       var.equal = FALSE)
```

`cholesterol ~ treatment` is a **model formula**. We'll see these again later. To the left of the `~` is the outcome variable (a.k.a. response, dependent variable). To the right of the `~` are the explanatory variables (a.k.a. predictors, independent variables).

`paired = FALSE` gives us the **unpaired, two-sample t-test**.

`var.equal = FALSE` gives us the **t-test not assuming equal variances**, a.k.a. Welch's t-test.

```
Rows: 24
```

```
Columns: 3
```

```
$ ID          <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, ...  
$ cholesterol <dbl> 178, 240, 210, 184, 190, 181, 156, 220, ...  
$ treatment   <chr> "placebo", "placebo", "placebo", "plac...
```

Welch Two Sample t-test

```
data:  cholesterol by treatment
```

```
t = 0.3855, df = 21.871, p-value = 0.7036
```

```
alternative hypothesis: true difference in means between groups
```

```
95 percent confidence interval:
```

```
-17.52587  25.52587
```

```
sample estimates:
```

mean in group epinephrine	mean in group placebo
198.6667	194.6667

You will note that R does not round the results to a sensible number of decimal places.

Wilcoxon rank sum test (Mann-Whitney test)

```
wilcox.test(cholesterol ~ treatment,  
            data = epin,  
            paired = FALSE)
```

Often referred to as the "non-parametric equivalent of the two sample t-test".

```
Warning in wilcox.test.default(x = DATA[[1L]], y =  
DATA[[2L]], ...): cannot compute exact p-value with ties
```

Wilcoxon rank sum test with continuity correction

```
data: cholesterol by treatment
```

```
W = 84, p-value = 0.5062
```

```
alternative hypothesis: true location shift is not equal to 0
```

Let's look a little closer at this data set

```
print(epin, n=24)
```

```
# A tibble: 24 × 3
```

	ID	cholesterol	treatment
	<dbl>	<dbl>	<chr>
1	1	178	placebo
2	2	240	placebo
3	3	210	placebo
4	4	184	placebo
5	5	190	placebo
6	6	181	placebo
7	7	156	placebo
8	8	220	placebo
9	9	210	placebo
10	10	165	placebo
11	11	188	placebo
12	12	214	placebo
13	1	184	epinephrine
14	2	243	epinephrine
15	3	210	epinephrine
16	4	189	epinephrine
17	5	200	epinephrine
18	6	191	epinephrine
19	7	150	epinephrine

```
epin_wide <-
```

```
  epin %>%
```

```
  pivot_wider(names_from = "treatment",
```

```
              values_from = "cholesterol")
```

```
print(epin_wide, n=12)
```

```
# A tibble: 12 × 3
```

	ID	placebo	epinephrine
	<dbl>	<dbl>	<dbl>
1	1	178	184
2	2	240	243
3	3	210	210
4	4	184	189
5	5	190	200
6	6	181	191
7	7	156	150
8	8	220	226
9	9	210	220
10	10	165	163
11	11	188	192
12	12	214	216

Paired t-test

```
t.test(epin_wide$epinephrine, epin_wide$placebo,  
       paired = TRUE)
```

`paired = TRUE` gives us the **paired t-test**.

Paired t-test

```
data: epin_wide$epinephrine and epin_wide$placebo  
t = 2.7763, df = 11, p-value = 0.01802  
alternative hypothesis: true mean difference is not equal to 0  
95 percent confidence interval:  
 0.8289329 7.1710671  
sample estimates:  
mean difference
```

4

Wilcoxon signed rank test

```
wilcox.test(epin_wide$epinephrine, epin_wide$placebo,  
            paired = TRUE)
```

Often referred to as the "nonparametric equivalent of the paired t-test".

```
Warning in wilcox.test.default(epin_wide$epinephrine,  
epin_wide$placebo, : cannot compute exact p-value with ties
```

```
Warning in wilcox.test.default(epin_wide$epinephrine,  
epin_wide$placebo, : cannot compute exact p-value with  
zeroes
```

Wilcoxon signed rank test with continuity correction

```
data: epin_wide$epinephrine and epin_wide$placebo
```

```
V = 57.5, p-value = 0.03208
```

```
alternative hypothesis: true location shift is not equal to 0
```

Getting test results in a data frame

The `tidy()` function in the `broom` package takes the output from a statistical test and saves it in a data frame. It works for most statistical tests available in R.

```
library(broom)
epin_test <- t.test(epin_wide$epinephrine,
                   epin_wide$placebo, paired = TRUE) %>%
  tidy()
glimpse(epin_test)
```

```
Rows: 1
Columns: 8
$ estimate      <dbl> 4
$ statistic     <dbl> 2.776334
$ p.value       <dbl> 0.01802169
$ parameter     <dbl> 11
$ conf.low      <dbl> 0.8289329
$ conf.high     <dbl> 7.171067
$ method        <chr> "Paired t-test"
$ alternative    <chr> "two.sided"
```


Getting test results in a data frame

The `tidy()` function in the `broom` package takes the output from a statistical test and saves it in a data frame. It works for most statistical tests available in R.

```
library(broom)
epin_test <- t.test(epin_wide$epinephrine,
                   epin_wide$placebo, paired = TRUE) %>%
  tidy()
glimpse(epin_test)
```

```
t.test(epin_wide$epinephrine,
       epin_wide$placebo, paired = TRUE) %>%
  tidy() %>%
  select(estimate, conf.low, conf.high, p.value) %>%
  gt() %>%
  fmt_number(c(estimate, conf.low, conf.high),
             decimals = 1) %>%
  fmt_number(p.value, decimals = 3) %>%
  cols_merge_range(conf.low, conf.high, sep = ", ") %>%
  cols_align("center", everything()) %>%
  cols_label(estimate = "Estimate", conf.low = "95% CI",
             p.value = "P-value") %>%
  tab_spanner("Mean difference (epinephrine - placebo)",
             estimate:conf.high)
```

```
Rows: 1
Columns: 8
$ estimate      <dbl> 4
$ statistic     <dbl> 2.776334
$ p.value       <dbl> 0.01802169
$ parameter     <dbl> 11
$ conf.low      <dbl> 0.8289329
$ conf.high     <dbl> 7.171067
$ method        <chr> "Paired t-test"
$ alternative    <chr> "two.sided"
```

Mean difference (epinephrine - placebo)		
Estimate	95% CI	P-value
4.0	0.8, 7.2	0.018

Correlation between continuous variables

Pearson correlation is the traditional measure of linear association between continuous variables.

```
cor.test(epin_wide$epinephrine, epin_wide$placebo,  
         method = "pearson")
```

Or equivalently (but arguably a bit neater):

```
cor.test(~ epinephrine + placebo,  
         data = epin_wide,  
         method = "pearson")
```

Do you expect this correlation to be positive or negative?

Respond at [PollEv.com/rrr](https://poll-ev.com/rrr)

Pearson's product-moment correlation

data: epinephrine and placebo

t = 17.254, df = 10, p-value = 9.043e-09

alternative hypothesis: true correlation is not equal to 0

95 percent confidence interval:

0.9407921 0.9955380

sample estimates:

cor

0.9836171

Correlation between continuous variables

Spearman's rho is a rank-based measure of association which may be more appropriate when dealing with ordinal variables or non-linear relationships.

```
cor.test(~ epinephrine + placebo,  
        data = epin_wide,  
        method = "spearman")
```

```
Warning in cor.test.default(x = mf[[1L]], y = mf[[2L]],  
...): Cannot compute exact p-value with ties
```

Spearman's rank correlation rho

```
data: epinephrine and placebo  
S = 5.5092, p-value = 2.022e-08  
alternative hypothesis: true rho is not equal to 0  
sample estimates:  
rho  
0.9807371
```

Fisher's Exact Test

This is the recommended test for association between categorical variables in most circumstances.

```
dental_decay <-  
  read_csv("dentaldecay.csv")
```

```
dental_decay %>%  
  tabyl(Brush, Caries) %>%  
  adorn_percentages("row") %>%  
  adorn_pct_formatting(digits = 0) %>%  
  adorn_ns(position = "front") %>%  
  gt()
```

Brush	No	Yes
Once or less a day	83 (72%)	33 (28%)
Twice or more a day	80 (77%)	24 (23%)

Fisher's Exact Test

This is the recommended test for association between categorical variables in most circumstances.

```
fisher.test(dental_decay$Brush, dental_decay$Caries)
```

Odds ratios and confidence intervals are displayed when both variables are dichotomous (binary).

Brush	No	Yes
Once or less a day	83 (72%)	33 (28%)
Twice or more a day	80 (77%)	24 (23%)

Fisher's Exact Test for Count Data

```
data: dental_decay$Brush and dental_decay$Caries
p-value = 0.4412
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 0.390300 1.447092
sample estimates:
odds ratio
 0.7555144
```

Chi-squared Test

This is the traditional test for association between categorical variables. It is based on a large-sample approximation.

```
chisq.test(dental_decay$Brush, dental_decay$Caries)
```

```
Pearson's Chi-squared test with Yates' continuity  
correction
```

```
data: dental_decay$Brush and dental_decay$Caries  
X-squared = 0.56811, df = 1, p-value = 0.451
```

The `janitor` package enables the use of `chisq.test` with a `tbl` as the input.

Extension

`corrr` package: pairwise correlations (highly recommended).

<https://cran.r-project.org/web/packages/corrr/vignettes/using-corrr.html>

`pairwise.t.test()`: t-tests between all pairs of groups.

`pairwise.wilcox.test()`: Wilcoxon tests between all pairs of groups.

`pairwise.prop.test()`: tests for differences in proportion between all pairs of groups.

Exercise 4.2.