T-Test

Background

Used to compare the mean of a sample to a known value or between two groups to determine if their means are significantly different.

- One-Sample T-Test: Compares the sample mean to a known or hypothesized population mean.
- Independent T-Test: Compares the means of two independent groups.
- Paired T-Test: Compares means from the same group at two different times (before/after) or under two different conditions.

Requirements

To perform a valid **t-test**, certain assumptions and requirements must be met depending on the type of t-test (one-sample, independent, or paired). The general requirements for any t-test include the following:

1. **Scale of Measurement**

- The dependent variable should be **continuous** (interval or ratio scale). The t-test compares means, so the data must be numerical.

2. **Random Sampling**

- The sample should be drawn from the population **randomly** to ensure that the results are representative of the population and that biases are minimized.

3. **Independence**

- Observations must be **independent** of each other.
    - For an **independent t-test**, this means the two groups being compared should not be related or influence each other.
    - For a **paired t-test**, the paired observations (e.g., pre-test and post-test) should be related but independent of other pairs.

4. **Normality**

- The data for each group (or the differences in a paired t-test) should be **approximately normally distributed**. This assumption is more critical for small sample sizes (n < 30), but the t-test is fairly robust to violations of normality with larger sample sizes due to the **Central Limit Theorem**.

5. **Homogeneity of Variance (for Independent T-Test)**

- The variances of the two groups should be **equal** (also called homoscedasticity). This assumption is tested by **Levene’s Test**.
- If the assumption of equal variances is violated, a modified version of the t-test, known as **Welch’s t-test**, can be used.

6. **Sample Size**

- The t-test is most appropriate when the sample size is moderate (n > 30) for normal data. For smaller sample sizes, the assumption of normality becomes more important.

7. **No Significant Outliers**

- Outliers can heavily influence the results of a t-test, so it is important to check for and address significant outliers before performing the test. Outliers may affect the normality assumption and distort the mean.

Code Used

T-test function

To conduct a one-sample t-test in R, we use the syntax `t.test(v, mu = 0)` where `v` is the name of our vector of interest and `mu` is set equal to the mean specified by the null hypothesis.

The t.test function is part of base R. No package needs to be loaded.

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

`x`	a (non-empty) numeric vector of data values.
`y`	an optional (non-empty) numeric vector of data values.
`alternative`	a character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"`. You can specify just the initial letter.
`mu`	a number indicating the true value of the mean (or difference in means if you are performing a two sample test).
`paired`	a logical indicating whether you want a paired t-test.
`var.equal`	a logical variable indicating whether to treat the two variances as being equal. If `TRUE` then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
`conf.level`	confidence level of the interval.
`formula`	a formula of the form `lhs ~ rhs` where `lhs` is a numeric variable giving the data values and `rhs` either `1` for a one-sample or paired test or a factor with two levels giving the corresponding groups. If `lhs` is of class `"Pair"` and `rhs` is `1`, a paired test is done.
`data`	an optional matrix or data frame (or similar: see `model.frame` (http://127.0.0.1:8315/help/library/stats/help/model.frame)) containing the variables in the formula `formula`. By default the variables are taken from `environment(formula)`.
`subset`	an optional vector specifying a subset of observations to be used.
`na.action`	a function which indicates what should happen when the data contain `NA` s. Defaults to `getOption("na.action")`.

`...`	further arguments to be passed to or from methods

Sample

A sample of a vector can be obtained by using `sample(x, size, replace = FALSE, prob = NULL)`. The code shows the defaults for `replace` and `prob` and details on each argument can be found in the following table.

`Sample` Arguments

`x`	either a vector of one or more elements from which to choose, or a positive integer. See ‘Details.’
`n`	a positive number, the number of items to choose from. See ‘Details.’
`size`	a non-negative integer giving the number of items to choose.
`replace`	should sampling be with replacement?
`prob`	a vector of probability weights for obtaining the elements of the vector being sampled.

`v = sample(vector, n, replace = FALSE, prob = NULL)` will produce a vector of the desired sample length for our test if desired.

Levene’s Test

From the `car` package we have `leveneTest` which allows us to enter a formula for a linear model in the basic form of `y ~ x`. Where:

- `y` is the data of interest
- `x` is the grouping variable

`car::leveneTest(y ~ x, data = df, center = mean)` (The vignette explains that using `mean = median` provides a more robust test.)

Use `?car::LeveneTest` to find out more.

Normality Test (Shapiro-Wilk)

The normality test in R uses the base function `shapiro.test(x)` which accepts a vector x and outputs a test result that allows us to determine if the data in the vector supplied is sufficiently normally distributed. However, one would have to place each vector of interest into the function and separately compute the result.

The null hypothesis is that the sample of data comes from a normal distribution.

However, there are pipe-friendly ways to do this. First, with the visual methods.

Visual methods

Density plot and **Q-Q plot** can be used to check normality visually.

1. **Density plot**: the density plot provides a visual judgment about whether the distribution is bell shaped.
2. **QQ plot**: QQ plot (or quantile-quantile plot) draws the correlation between a given sample and the normal distribution. A 45-degree reference line is also plotted. In a QQ plot, each observation is plotted as a single dot. If the data are normal, the dots should form a straight line.

Density Plots can be obtained using the `ggpubr` package: `ggpubr::ggdensity(vector, fill = "black")`

Q-Q plots can be obtained by `ggqqplot(vector)` from the same package.

Now, to perform the shapiro test on multiple groups in a dataframe we used the `shapiro_test` function from the `rstatix` package. The the same understanding as before that `y` contains the numerical data of interest and `x` is the grouping variable.

```
df |>
  group_by(x) |>
  rstatix::shapiro_test(y)
```

Effect Size

```
effsize::cohen.d(
                df$var ~ df$factor,
                paired = FALSE,
                )

cohen.d(d,f,pooled=TRUE,paired=FALSE,
                   na.rm=FALSE, mu=0, hedges.correction=FALSE,
                   conf.level=0.95,noncentral=FALSE,
                   within=TRUE, subject=NA, ...)
```

Arguments

`d`	a numeric vector giving either the data values (if `f` is a factor) or the treatment group values (if `f` is a numeric vector)
`f`	either a factor with two levels or a numeric vector of values, if `NA` a single sample effect size is computed
`formula`	a formula of the form `y ~ f` , where `y` is a numeric variable giving the values and `f` a factor with two levels giving the corresponding groups. If using a paired computation (`paired=TRUE`) it is possible to specify the ids of the subjects using the form `y ~ f | Subject(id)` which allow the correct pairing of the pre and post values. A single sample effect size can be specified with the form `y ~ .` .
`data`	an optional matrix or data frame containing the variables in the formula `formula` . By default the variables are taken from `environment(formula)` .
`pooled`	a logical indicating whether compute pooled standard deviation or the whole sample standard deviation. If `pooled=TRUE` (default) pooled sd is used, if `pooled=FALSE` the standard deviation of the the control group (the second argument or the one corresponding the the second level of the factor) is used instead.
`hedges.correction`	logical indicating whether apply the Hedges correction
`conf.level`	confidence level of the confidence interval
`noncentral`	logical indicating whether to use non-central t distributions for computing the confidence interval.

`paired`	a logical indicating whether to consider the values as paired, a warning is issued if `paired==TRUE` with the formula interface and not `| Subject(id)` or with data and factor and no `subject` is provided
`within`	indicates whether to compute the effect size using the within subject variation, taking into consideration the correlation between pre and post samples.
`subject`	an array indicating the id of the subject for a paired computation, when the formula interface is used it can be indicated in the formula by adding `| Subject(id)` , where `id` is the column in the data that contains and id of the subjects to be paired.
`mu`	numeric indicating the reference mean for single sample effect size.
`na.rm`	logical indicating whether `NA` s should be removed before computation; if `paired==TRUE` then all incomplete pairs are removed.
`...`	further arguments to be passed to or from methods.

Statistical Power

Compute power of tests or determine parameters to obtain target power (similar to power.t.test)

```
pwr.t.test(n = NULL, d = NULL, sig.level = 0.05, power = NULL,
    type = c("two.sample", "one.sample", "paired"),
    alternative = c("two.sided", "less", "greater"))
```

Arguments

`n`	Number of observations (per sample)
`d`	Effect size (Cohen’s d) - difference between the means divided by the pooled standard deviation
`sig.level`	Significance level (Type I error probability)
`power`	Power of test (1 minus Type II error probability)
`type`	Type of t test : one- two- or paired-samples
`alternative`	a character string specifying the alternative hypothesis, must be one of “two.sided” (default), “greater” or “less”

Details

Exactly one of the parameters ‘d’,‘n’,‘power’ and ‘sig.level’ must be passed as NULL, and that parameter is determined from the others. Notice that the last one has non-NULL default so NULL must be explicitly passed if you want to compute it.

- **Effect Size (d)**: The difference between the means in terms of standard deviation units. If not known, it can be estimated using the formula: \[d = \frac{\text{mean difference}}{\text{standard deviation of differences}}\]
- **Significance Level (α)**: Typically set to 0.05.
- **Sample Size (n)**: The number of paired observations.
- **Power (1 - β)**: The probability of correctly rejecting the null hypothesis when it is false (e.g., 0.8 or 80%).

Outliers

Boxplots can be used to visually detect outliers. Base R and `ggplot2` provide means of doing this.

Using the pre-loaded `graphics` package

```
boxplot(
  y ~ x,
  data = df,
  xlab = "X-Varname",
  ylab = "Y-Varname"
)
```

Using `ggplot2` :

```
df |>
  ggplot(
    mapping = aes(x, y, fill = x)
  )+
  geom_boxplot()
```

Data and Testing Requirements

We can use the Iris dataset, accessed through the command `data("iris")` , for which the description is:

> This famous (Fisher’s or Anderson’s) iris data set gives the measurements in centimeters of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are Iris setosa, versicolor, and virginica.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
data("iris")
```

```
iris |> sample_n(20) |>
  knitr::kable(caption = "Sample of the Iris Dataset") |>
  kableExtra::kable_paper("hover", full_width = TRUE)
```

Sample of the Iris Dataset

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.9	3.0	5.1	1.8	virginica
6.5	2.8	4.6	1.5	versicolor
7.2	3.2	6.0	1.8	virginica
6.1	2.8	4.0	1.3	versicolor
7.7	2.6	6.9	2.3	virginica
6.1	3.0	4.6	1.4	versicolor
5.9	3.2	4.8	1.8	versicolor
4.9	3.1	1.5	0.2	setosa
4.9	3.1	1.5	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.5	3.5	1.3	0.2	setosa
6.5	3.2	5.1	2.0	virginica
5.1	2.5	3.0	1.1	versicolor
6.3	3.3	6.0	2.5	virginica
4.4	2.9	1.4	0.2	setosa
5.6	2.8	4.9	2.0	virginica
5.4	3.4	1.7	0.2	setosa
5.0	3.4	1.6	0.4	setosa
4.9	2.4	3.3	1.0	versicolor
5.1	3.5	1.4	0.2	setosa

Let’s test for normality.

```
iris |>
  group_by(Species) |>
  rstatix::shapiro_test(Sepal.Length) |>
  knitr::kable(
    caption = "Normality Test Results for Different Groups of Sepal Lengths"
  ) |>
  kableExtra::kable_paper("hover", full_width = TRUE)
```

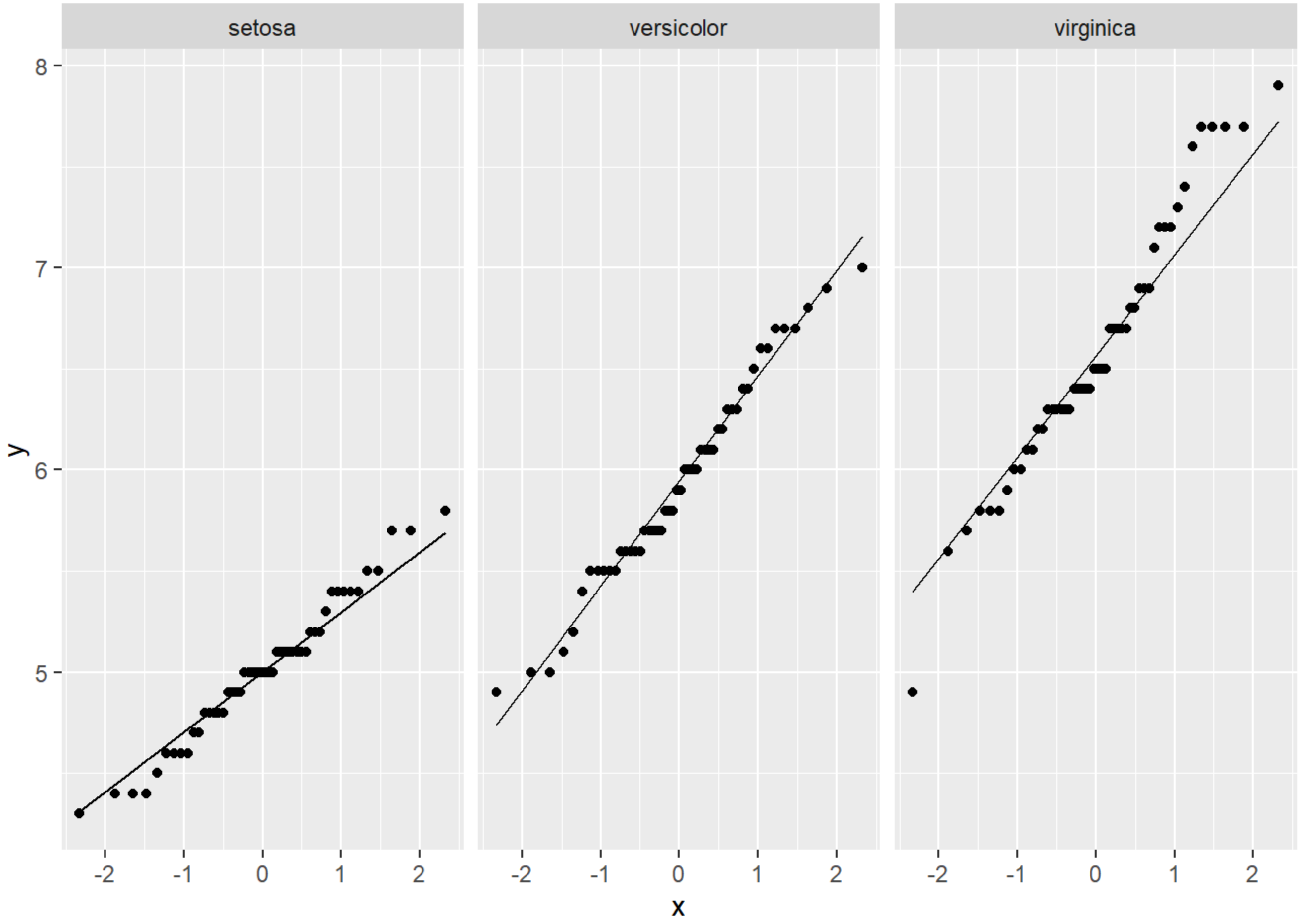
Normality Test Results for Different Groups of Sepal Lengths

Species	variable	statistic	p
setosa	Sepal.Length	0.9776985	0.4595132
versicolor	Sepal.Length	0.9778357	0.4647370
virginica	Sepal.Length	0.9711794	0.2583147

None of the results are statistically significant, so we assume normality within each group and carry on.

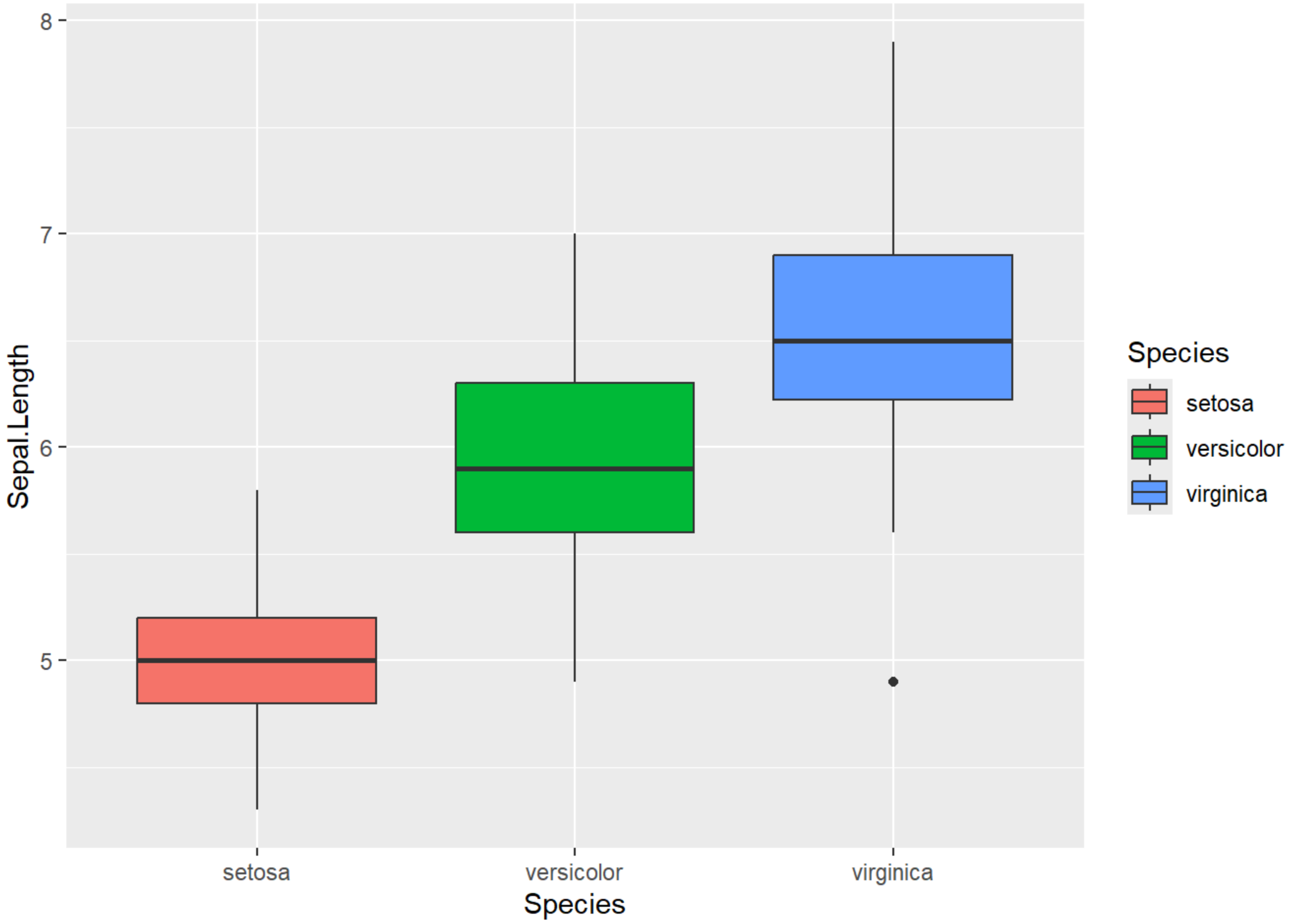
We can also do Q-Q Plots:

```
iris |>
  ggplot(mapping = aes(sample = Sepal.Length)) +
  geom_qq()+
  stat_qq_line()+
  facet_grid(. ~ Species)
```



and boxplots

```
iris |>
  ggplot(mapping = aes(x = Species, y = Sepal.Length, fill = Species))+
  geom_boxplot()
```



From both the Q-Q plot and the boxplots we can see that there is one outlier in the Virginica set.

Now, to finally see if variances are equal.

```
car::leveneTest(Sepal.Length ~ Species, data = iris, center = mean)
```

From the boxplots we already know that Setosa has lower Sepal Lengths, so we exclude it. The result means that all the variances are not equal among the species.

```
car::leveneTest(Sepal.Length ~ Species,
                data = iris |> filter(!Species == "setosa"),
                center = mean)
```

Now, it returns a p-value of 0.2363937 which means that we can assume that the variances are equal and continue.

Carrying out different T-Tests

One Sample T-Test

To carry out a t-test we must first isolate a vector we want and run it through the function.

The first example takes all the Sepal.Length data and tests whether the true mean is equal to 6.

```
t.test(Sepal.Length ~ 1, data = iris, mu = 6)
```

```
##
##  One Sample t-test
##
## data:  Sepal.Length
## t = -2.3172, df = 149, p-value = 0.02186
## alternative hypothesis: true mean is not equal to 6
## 95 percent confidence interval:
##  5.709732 5.976934
## sample estimates:
## mean of x
##  5.843333
```

Results suggest that the mean of the Sepal-Lengths does not differ significantly from 6 cm.

```
t.test(Sepal.Length ~ 1, data = iris, alternative = "less", mu = 6)
```

```
##
##  One Sample t-test
##
## data:  Sepal.Length
## t = -2.3172, df = 149, p-value = 0.01093
## alternative hypothesis: true mean is less than 6
## 95 percent confidence interval:
##     -Inf 5.95524
## sample estimates:
## mean of x
##  5.843333
```

There is no evidence at the 5% significance level that the mean is less than 6 cm

Let’s test one of them, Virginica

```
t.test(iris[iris$Species == "virginica",]$Sepal.Length, mu = 6, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  iris[iris$Species == "virginica", ]$Sepal.Length
## t = 6.5386, df = 49, p-value = 1.72e-08
## alternative hypothesis: true mean is greater than 6
## 95 percent confidence interval:
##  6.437233      Inf
## sample estimates:
## mean of x
##     6.588
```

We can see that the p-value for this test is 0.00000001720484 < 0.05. The result is significant and we can say that there is sufficient evidence at the 5% level to suggest that the mean of the Sepal Lengths of the Virginica species is greater than 6 cm.

The `rstatix` package makes it easy to carry out tests in a `dplyr`-friendly way.

```
rstatix::t_test(
  data,
  formula,
  comparisons = NULL,
  ref.group = NULL,
  p.adjust.method = "holm",
  paired = FALSE,
  var.equal = FALSE,
  alternative = "two.sided",
  mu = 0,
  conf.level = 0.95,
  detailed = FALSE
  )
```

Arguments

`data`	a data.frame containing the variables in the formula.
`formula`	a formula of the form `x ~ group` where `x` is a numeric variable giving the data values and `group` is a factor with one or multiple levels giving the corresponding groups. For example, `formula = TP53 ~ cancer_group` .
`comparisons`	A list of length-2 vectors specifying the groups of interest to be compared. For example to compare groups “A” vs “B” and “B” vs “C”, the argument is as follow: `comparisons = list(c("A", "B"), c("B", "C"))`
`ref.group`	a character string specifying the reference group. If specified, for a given grouping variable, each of the group levels will be compared to the reference group (i.e. control group). If `ref.group = "all"` , pairwise two sample tests are performed for comparing each grouping variable levels against all (i.e. basemean).
`p.adjust.method`	method to adjust p values for multiple comparisons. Used when pairwise comparisons are performed. Allowed values include “holm”, “hochberg”, “hommel”, “bonferroni”, “BH”, “BY”, “fdr”, “none”. If you don’t want to adjust the p value (not recommended), use p.adjust.method = “none”.
`paired`	a logical indicating whether you want a paired test.
`var.equal`	a logical variable indicating whether to treat the two variances as being equal. If `TRUE` then the pooled variance is used to estimate the variance otherwise the Welch (or Satterthwaite) approximation to the degrees of freedom is used.
`alternative`	a character string specifying the alternative hypothesis, must be one of `"two.sided"` (default), `"greater"` or `"less"` . You can specify just the initial letter.
`mu`	a number specifying an optional parameter used to form the null hypothesis.
`conf.level`	confidence level of the interval.
`detailed`	logical value. Default is FALSE. If TRUE, a detailed result is shown.

`pool.sd` logical value used in the function `pairwise_t_test()` . Switch to allow/disallow the use of a pooled SD.

The `pool.sd = TRUE` (default) calculates a common SD for all groups and uses that for all comparisons (this can be useful if some groups are small). This method does not actually call t.test, so extra arguments are ignored. Pooling does not generalize to paired tests so pool.sd and paired cannot both be TRUE.

If `pool.sd = FALSE` the standard two sample t-test is applied to all possible pairs of groups. This method calls the `t.test()` , so extra arguments, such as `var.equal` are accepted.

`...` other arguments to be passed to the function `t.test` (http://127.0.0.1:8315/help/library/stats/help/t.test).

```
iris |>
  group_by(Species) |>
  rstatix::t_test(Sepal.Length ~ 1,
                  mu = 6,
                  alternative = "greater") |>
  knitr::kable() |>
  kableExtra::kable_paper(full_width = TRUE)
```

Species	.y.	group1	group2	n	statistic	df	p
setosa	Sepal.Length	1	null model	50	-19.9399916	49	1.000
versicolor	Sepal.Length	1	null model	50	-0.8767409	49	0.808
virginica	Sepal.Length	1	null model	50	6.5386402	49	0.000

We can quickly see which group yields a significant result, i.e., Virginica.

Independent T-Test

As stated before, this version of the t-test compares the means of two independent groups.

Having already done the work to evaluate the prerequisites for the test:

- normality check
- equal variances test
- outliers check

We can use the base function, `t.test()` to test the means between two groups. We can choose Virginica and Versicolor. First, we need to check if the variances are equal, so we can use this result in the t-test. We know that he data is nearly normally distributed for each group.

```
t.test(x, y = NULL,
       alternative = c("two.sided", "less", "greater"),
       mu = 0, paired = FALSE, var.equal = FALSE,
       conf.level = 0.95, ...)
```

```
car::leveneTest(Sepal.Length ~ Species, data = iris[iris$Species != "setosa",], center = mean)
```

```
t.test(iris[iris$Species == "versicolor",]$Sepal.Length,
       iris[iris$Species == "virginica",]$Sepal.Length,
       paired = FALSE,
       var.equal = TRUE
       )
```

```
##
##  Two Sample t-test
##
## data:  iris[iris$Species == "versicolor", ]$Sepal.Length and iris[iris$Species == "virginica", ]$Sepal.L
ength
## t = -5.6292, df = 98, p-value = 1.725e-07
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.8818516 -0.4221484
## sample estimates:
## mean of x mean of y
##     5.936     6.588
```

The evidence suggests that there is a significant difference in the means given that the p-value for the t-test is 0.0000001724856 < 5.

Using the `rstatix` package we can perform multiple independent sample t-tests. However, since we know from before that the variance of the Setosa set is not equal to that of Virginica and Vertsicolor, we will set `var.equal = FALSE`.

```
iris |>
  rstatix::t_test(Sepal.Length ~ Species) |>
  knitr::kable() |>
  kableExtra::kable_paper("hover", full_width = TRUE)
```

.y.	group1	group2	n1	n2	statistic	df	p	p.adj	p.adj.signif
Sepal.Length	setosa	versicolor	50	50	-10.520986	86.53800	0e+00	0e+00	****
Sepal.Length	setosa	virginica	50	50	-15.386196	76.51587	0e+00	0e+00	****
Sepal.Length	versicolor	virginica	50	50	-5.629165	94.02549	2e-07	2e-07	****

The table easily shows us significant differences in the means between each pair of species. This is easier, else we'd have to do 3 separate t-tests.

Paired Sample T-Test

The data in the iris dataset is not suitable to perform a paired sample t-test. We need data for which there are two sets of measurements taken from the SAME SUBJECT. E.g., a before and after scenario.

To do this, we use the Neonatal Hypothermia Dataset (https://www.kaggle.com/datasets/zahrazolghadr/neonatal-hypothermia) from Kaggle. The following quote is from Zahra Zolghadr, a Kaggle user:

> The paired samples t-test is a powerful statistical method designed to assess whether there is a significant difference between the means of two related groups. It is particularly useful when dealing with paired observations, such as before-and-after measurements or matched pairs in experimental designs. The test focuses on the differences within each pair and evaluates whether the average of these differences is significantly different from zero. The paired samples t-test takes into account the inherent correlation between the paired observations, making it a valuable tool for scenarios where individual variability might obscure the overall treatment effect. By comparing the mean of the differences to the expected mean under the null hypothesis, the test enables researchers to draw conclusions about the presence or absence of a systematic effect. This method is widely employed in various fields, including medicine, psychology, and social sciences, to investigate the impact of interventions, treatments, or changes over time in a paired manner.

Further notes from the same user:

> When applying a two-paired t-test, it is essential to first separate the two sets of observations, typically labeled as “before” and “after”. This test is commonly employed in research settings where the goal is to evaluate whether there is a statistically significant difference in the means of paired measurements, such as pre- and post-treatment data. To conduct the analysis, one must formulate the null and alternative hypotheses. The null hypothesis posits that there is no significant difference between the means of the two paired groups, suggesting that any observed variations are due to random chance. The alternative hypothesis, on the other hand, asserts that there is a significant difference between the means, indicating a systematic effect. Depending on the research question, the alternative hypothesis can take different forms, such as “greater than”, “less than”, or “not equal to”.

> For example, in the context of a clinical study evaluating the effectiveness of a new drug, the null hypothesis might state that there is no significant difference in the mean health scores before and after treatment. The alternative hypothesis could be formulated as “the mean health scores after treatment are significantly greater than the mean health scores before treatment”. The two-paired t-test is then applied to the data, providing a statistical assessment of whether the observed differences are likely to be due to a real effect rather than random variability.

```
neonatal_hypothermia <- readr::read_csv("data/Hypothermia.csv")
```

```
## Rows: 200 Columns: 13
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (13): case, code, date, time, weight, t.nur, t.or, t.1, t.2, t.3, t.4, t...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
neonatal_hypothermia <- neonatal_hypothermia |>
  mutate_at(c(1:3,4:13), as.numeric)

neonatal_hypothermia |> sample_n(20) |> knitr::kable(caption = "Sample of 20 Rows from Neonatal Hypothermia
Data") |> kableExtra::kable_paper("hover", full_width = TRUE)
```

Sample of 20 Rows from Neonatal Hypothermia Data

case	code	date	time	weight	t.nur	t.or	t.1	t.2	t.3	t.4	t.5	t.6
141	96808	6.00	11.45	3120	30	30	35.5	37.4	NA	NA	NA	NA
135	95623	2.00	6.40	2780	30	30	35.5	36.5	37.0	NA	NA	NA
175	97967	9.00	11.00	1750	30	28	35.8	36.8	NA	NA	NA	NA
87	95942	3.00	16.35	2820	32	30	35.4	36.6	NA	NA	NA	NA
187	99233	81.01	9.50	3000	29	24	35.8	36.2	36.8	NA	NA	NA
159	99283	81.01	11.15	2450	27	24	35.6	36.8	NA	NA	NA	NA
114	95987	4.00	5.05	1450	30	32	35.5	36.5	37.0	NA	NA	NA
109	96589	5.00	11.30	1400	31	31	35.5	36.9	NA	NA	NA	NA
92	99324	81.01	9.50	3260	27	28	35.4	36.0	36.8	NA	NA	NA
198	98106	9.00	18.30	1720	29	24	35.9	36.8	NA	NA	NA	NA
151	95967	4.00	15.40	1600	30	31	35.6	36.5	36.8	NA	NA	NA
77	97252	7.00	19.45	1220	29	29	35.4	36.4	37.0	NA	NA	NA
59	97601	8.00	9.18	2050	28	23	35.2	35.8	36.0	36.8	NA	NA
169	98880	12.00	23.45	3840	29	28	35.6	36.0	36.6	36.8	NA	NA
48	97918	98.00	5.20	2500	29	24	35.0	36.8	NA	NA	NA	NA
178	99279	81.01	4.40	2140	29	25	35.8	36.2	37.0	NA	NA	NA
177	99097	81.01	12.20	2000	28	24	35.8	36.3	37.1	NA	NA	NA
68	97348	7.00	10.00	1220	28	26	35.3	36.0	NA	NA	NA	NA
33	97601	8.00	9.05	1650	28	23	35.0	35.6	35.8	NA	NA	NA
126	96285	4.00	3.55	2400	30	30	35.5	36.6	36.9	NA	NA	NA

The aim now is to see whether a significant difference exists between the first and second temperature readings.

Using base R, we'll obtain the first and second temperatures by:

- `neonatal_hypothermia$t.1`
- `neonatal_hypothermia$t.2`

```
car::leveneTest(neonatal_hypothermia$t.1, neonatal_hypothermia$t.2, center = mean)
```

```
## Warning in leveneTest.default(neonatal_hypothermia$t.1,
## neonatal_hypothermia$t.2, : neonatal_hypothermia$t.2 coerced to factor.
```

```
t.test(neonatal_hypothermia$t.2, neonatal_hypothermia$t.1, alternative = "greater", paired = TRUE, var.equa
l = TRUE)
```

```
##
##  Paired t-test
##
## data:  neonatal_hypothermia$t.2 and neonatal_hypothermia$t.1
## t = 22.284, df = 199, p-value < 2.2e-16
## alternative hypothesis: true mean difference is greater than 0
## 95 percent confidence interval:
##  0.9925032       Inf
## sample estimates:
## mean difference
##           1.072
```

Evidence suggests that a significant temperature increase has taken place. We need to modify how the data is arranged to use `rstatix`. Using `tidyr`'s `pivot_longer` will help us achieve creating two different temperature categories for analysis.

```
temperatures <- neonatal_hypothermia |>
  select(t.1, t.2) |>
  tidyr::pivot_longer(
    cols = everything() ,
    names_to = "temperature_reading",
    values_to = "temp"
    )

temperatures |> sample_n(20) |>  knitr::kable(caption = "Sample of Desired Temperature Data") |> kableExtr
a::kable_paper("hover", full_width = TRUE)
```

Sample of Desired Temperature Data

temperature_reading	temp
t.1	35.5
t.1	35.4
t.1	35.2
t.1	35.9
t.1	35.5

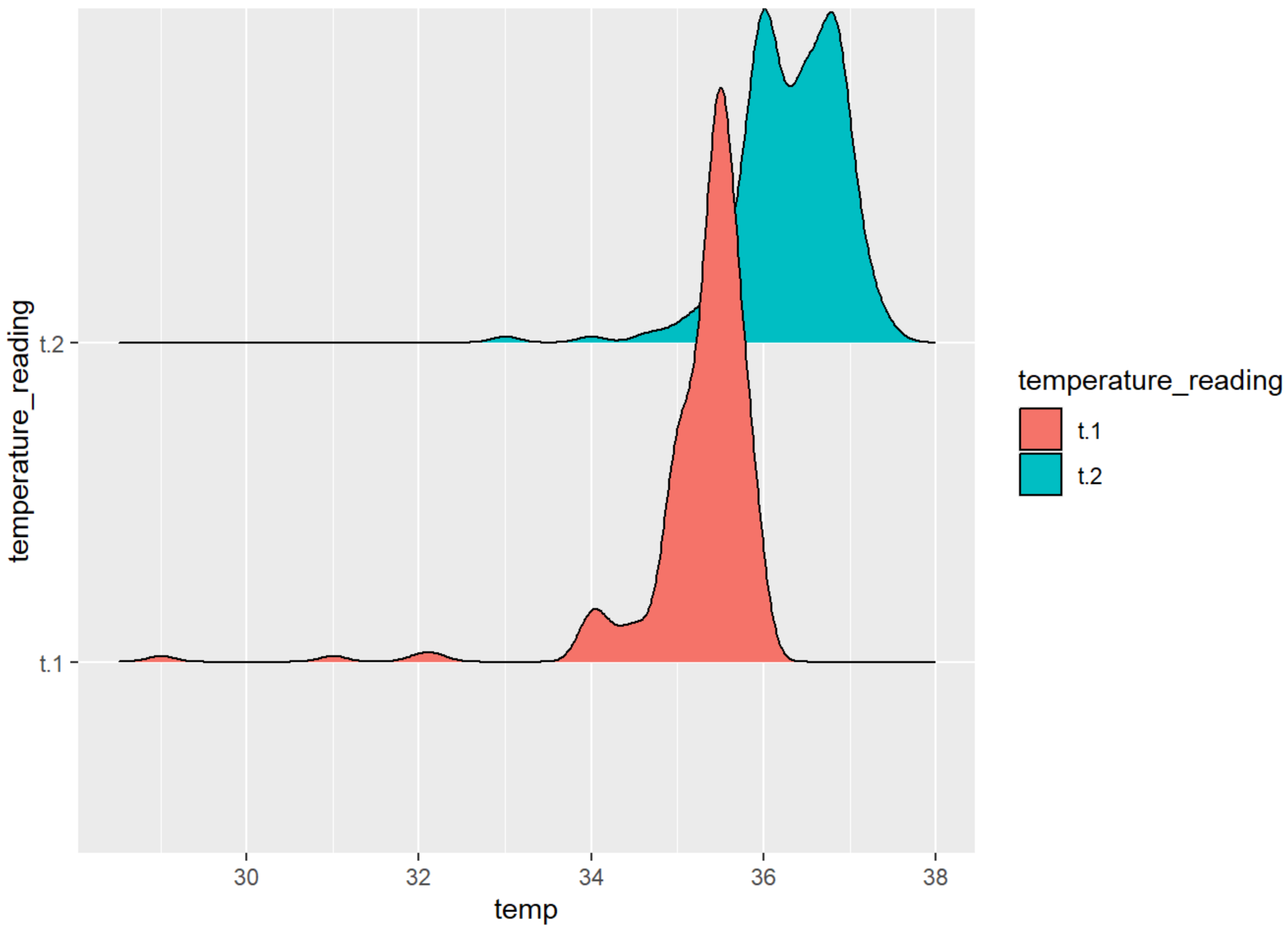
temperature_reading	temp
t.1	35.6
t.2	36.8
t.2	36.9
t.2	35.8
t.1	35.0
t.2	35.2
t.1	35.8
t.2	36.3
t.1	35.5
t.2	35.6
t.2	36.9
t.2	36.5
t.1	34.2
t.1	35.0
t.2	35.2

Now, we conduct the test. We can see that the normality assumption is violated AND we have many outliers for each category. For the sake of demonstration, we proceed with the t-test anyway and will, in a separate section, make adjustments to see what can be done with the data to properly test it.

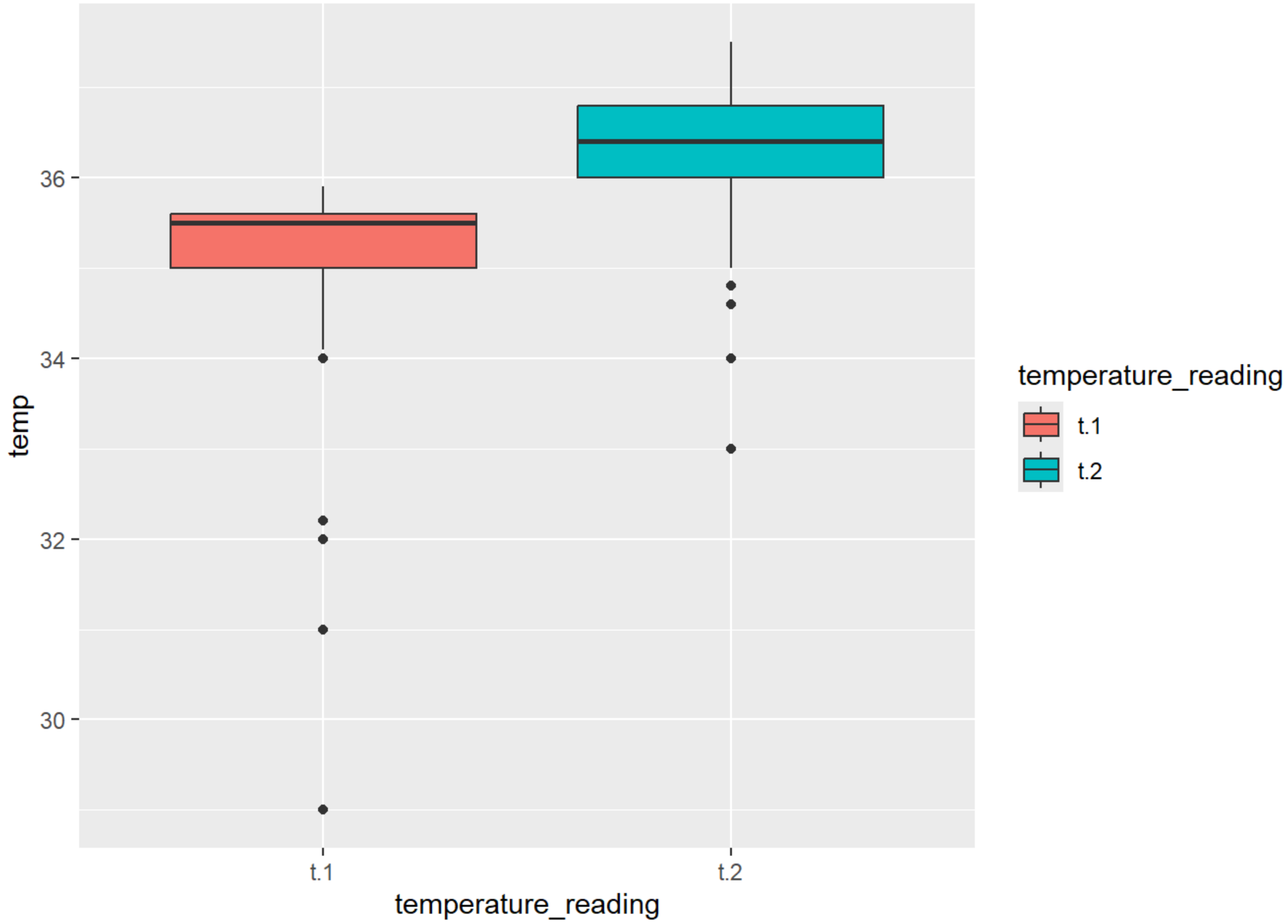
```
#Normality
temperatures |> group_by(temperature_reading)|> rstatix::shapiro_test(temp)
```

```
temperatures |> ggplot(
  mapping = aes(x = temp, y = temperature_reading, fill = temperature_reading)
)+
  ggridges::geom_density_ridges()
```

```
## Picking joint bandwidth of 0.162
```



```
temperatures |> ggplot(
  mapping = aes(x = temperature_reading, y = temp, fill = temperature_reading)
)+
  geom_boxplot()
```



```
#LeveneTest
temperatures |> rstatix::levene_test(temp ~ temperature_reading, center = mean)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
temperatures |> rstatix::t_test(temp ~ temperature_reading, paired = TRUE, var.equal = TRUE, alternative =
"greater") |> knitr::kable() |> kableExtra::kable_paper(full_width = TRUE)
```

.y.	group1	group2	n1	n2	statistic	df	p
temp	t.1	t.2	200	200	-22.2843	199	1

```
temperatures |> rstatix::cohens_d(temp ~ temperature_reading, paired = TRUE) |> knitr::kable() |> kableExtr
a::kable_paper(full_width = TRUE)
```

.y.	group1	group2	effsize	n1	n2	magnitude
temp	t.1	t.2	-1.575738	200	200	large

All the evidence, graphs included, point to a significant increase in temperature from t.1 to t.2.

Estimates of Effect Size and Statistical Power

The effect size for a paired-samples t-test can be calculated by dividing the mean difference by the standard deviation of the difference.

> T-test conventional effect sizes, proposed by Cohen, are: 0.2 (small efect), 0.5 (moderate effect) and 0.8 (large effect) (Cohen 1998, Navarro (2015)). This means that if two groups' means don't differ by 0.2 standard deviations or more, the difference is trivial, even if it is statistically significant.

\[Cohen's\ d\ formula: d = \frac{mean_d}{st.dev_D}\]

A **Power Analysis** to assess the likelihood that your test correctly rejects the null hypothesis when it is false (i.e., avoiding a Type II error).

```
test <- t.test(temp ~ temperature_reading, data = temperatures, paired = TRUE, var.equal = TRUE)

test
```

```
## 
##  Paired t-test
## 
## data:  temp by temperature_reading
## t = -22.284, df = 199, p-value < 2.2e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  -1.1668622 -0.9771378
## sample estimates:
## mean difference 
##          -1.072
```

```
effect <- effsize::cohen.d(temperatures$temp ~ temperatures$temperature_reading, paired = TRUE)
```

```
## Warning in cohen.d.formula(temperatures$temp ~
## temperatures$temperature_reading, : Trying to compute paired samples Cohen's d
## using formula input. Results may be incorrect if cases do not appear in the
## same order for both levels of the grouping factor. Use the format 'value ~
## treatment | Subject(id)' to specify a subject id variable.
```

```
effect
```

```
## 
## Cohen's d
## 
## d estimate: -1.551556 (large)
## 95 percent confidence interval:
##     lower     upper 
## -1.754751 -1.348362
```

```
pwr::pwr.t.test(n = nrow(neonatal_hypothermia),
                d = effect$estimate,
                sig.level = 0.05,
                type = "paired",
                alternative = "less"
                )
```

```
##
##      Paired t test power calculation
##
##               n = 200
##               d = -1.551556
##       sig.level = 0.05
##           power = 1
##     alternative = less
##
## NOTE: n is number of *pairs*
```

So, we have enough power to tell if one is significantly less than the other but, and we do have enough power to tell that there is a significant difference. It is understood that t.1 is being used as the reference.

```
pwr::pwr.t.test(n = nrow(neonatal_hypothermia),
                d = effect$estimate,
                sig.level = 0.05,
                type = "paired",
                alternative = "two.sided"
                )
```

```
##
##      Paired t test power calculation
##
##               n = 200
##               d = 1.551556
##       sig.level = 0.05
##           power = 1
##     alternative = two.sided
##
## NOTE: n is number of *pairs*
```