04-Distribution and Confidence Intervals of Maple Leaf Samples

2025-11-04

1 Introduction

The example aims to demonstrate estimation and interpretation of confidence intervals. At the end, the two samples are compared with respect to variance and mean values.

The experimental hypotheses is, that the sampling strategy has an influence on the parameters of the distribution, i.e. that a sampling bias may occur. Here we leave it open, if the “subjective sampling” strategy prefers bigger or smaller leaves or if it has an influence on variance. The result is to be visualized with bar charts and box plots. We use the **leave width** as an example, an analysis of the other variables is left as an optional exercise.

We can now derive the following statistical hypotheses **about the variance:**

- H_0: The variance of both samples is the same.
- H_a: The samples have different variance.

and about the mean:

- H_0: The mean of both samples is the same.
- H_a: The mean values of the samples are different.

2 Material and Methods

The data set consists of two samples of maple leaves (*Acer platanoides*), sampled in front of the institute building (Fig. 1).

Figure 1: Fig 1.: Maple leaves in front of the institute

The two samples where collected with different sampling strategies:

- HYB: hydrobiology group, got random sample from the supervisor
- HSE: hydroscience group, had the freedom to collect their leaves themselves

Then length, width and stalk length were measured in millimeters with a ruler (Fig. 2) and the data collected in a spreadsheet table and a csv-file.

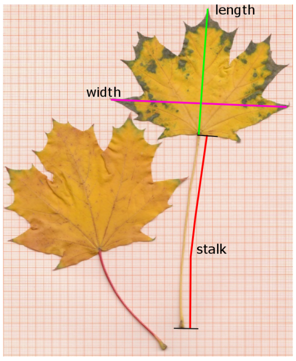

Figure 2: Fig 2.: Sample measures of maple leaves. Note that the stalk length does not include length of the leaf blade.

The statistical analysis is performed with the **R** software for statistical computing and graphics (R Core Team, 2024).

3 Data Analysis

3.1 Prepare and inspect data

The data set is available from your local learning management system (LMS, e.g. OPAL at TU Dresden) or publicly from https://tpetzoldt.github.io/datasets/data/leaves.csv.

- Download the data set `leaves.csv` and use one of RStudio's "Import Dataset" wizards.
- Alternative: use `read.csv()`.

```
#  ... do it
```

- plot everything, just for testing:

```
plot(leaves)
```

First, let's apply a traditional approach and split `leaves` in two separate tables for the samples HSE and MHYB:

```
hyb <- subset(leaves, group == "HYB")
hse <- subset(leaves, group == "HSE")
```

- Then, compare leaf **width** of both groups graphically:

```
boxplot(hse$width, hyb$width, names=c("HSE", "HYB"))
```

3.2 Check distribution

```
# use `hist`, `qqnorm`, `qqline`
# ...
```

3.3 Sample statistics and prediction interval

In a first analysis, we want to estimate the interval that covers 95% of leaves, defined by their width. As a first method, we take the **empirical quantiles** directly from the data. The method is also called "**nonparametric**," because we don't calculate mean and standard deviation and do not assume a normal or any other distribution.

```
quantile(hse$width, p = c(0.025, 0.975))
```

Now, we compare this empirical result with a method that relies on a specific distribution. If our initial graphical visualization (e.g., the histogram) suggests the data is reasonably symmetric and bell-shaped, we can proceed with a parametric assumption.

We first calculate mean, sd, N and se for "hse" data set:

```
hse.mean <- mean(hse$width)
hse.sd   <- sd(hse$width)
hse.N    <- length(hse$width)
hse.se   <- hse.sd/sqrt(hse.N)
```

Then we estimate an approximate two-sided 95% **prediction interval** (PI) for the sample using a simplified approach based on the quantiles of the theoretical normal distribution ($z_{\alpha/2} \approx 1.96$) and the sample parameters mean $\bar{x}$ and standard deviation (s):

$$PI = \bar{x} \pm z \cdot s$$

This is the interval where we would predict a new single observation to fall with 95% confidence.

```
hse.95 <- hse.mean + c(-1.96, 1.96) * hse.sd
hse.95
```

Instead of using 1.96, we could also use the quantile function `qnorm(0.975)` for the upper interval or `qnorm(c(0.025, 0.975))` for the lower and upper in parallel:

```
hse.95 <- hse.mean + qnorm(c(0.025, 0.975)) * hse.sd
hse.95
```

Now we plot the data and indicate the 95% interval:

```
plot(hse$width)
abline(h = hse.95, col="red")
```

… and the same as histogram:

```r
hist(hse$width)
abline(v = hse.95, col="red")
rug(hse$width, col="blue")
```

3.4 Confidence interval of the mean

The **confidence interval** (CI) of the mean tells us how precise a mean value was estimated from data. If the sample size is "large enough", the distribution of the raw data does not necessarily need to be normal, because then mean values tend to approximate a normal distribution due to the **central limit theorem**.

The formula for the CI of the mean is:

$$CI = \bar{x} \pm t_{\alpha/2,n-1} \cdot \frac{s}{\sqrt{N}}$$

3.4.1 Confidence interval of the mean for the "hse" data

Task: Calculate the confidence interval of the mean value of the "hse" data set using the quantile function (`qt`) of the t-distribution[1]:

```r
hse.ci <- hse.mean + qt(p = c(0.025, 0.975), df = hse.N - 1) * hse.se
```

Now indicate the confidence interval of the mean in the histogram.

```r
abline(v = hse.ci, col="magenta")
```

3.4.2 Confidence interval for the mean of the "hyb" data

```r
#  Do the same for the "hyb" data, calculate mean, sd, N, se and ci.
# ...
```

3.4.3 Visualization

Instead of a boxplot, we can also use a bar chart with confidence intervals.
This can be done with the add-on package **gplots** (not to be confused with **ggplot**):

Solution A) with package gplots

```r
library("gplots")
barplot2(height = c(hyb.mean, hse.mean),
         ci.l   = c(hyb.ci[1], hse.ci[1]),
         ci.u   = c(hyb.ci[2], hse.ci[2]),
         plot.ci = TRUE,
         names.arg=c("Hyb", "HSE")
)
```

[1] as the sample size is not too small, you may also compare this with 1.96 or 2.0

Solution B) without add-on packages (optional)

Here we use a standard bar chart, and line segments for the error bars. One small problem arises, because `barplot` creates an own x-scaling. The good news is, that `barplot` returns its x-scale. We can store it in a variable, e.g. `x` that can then be used in subsequent code.

```
x <- barplot(c(hyb.mean, hse.mean),
  names.arg=c("HYB", "HSE"), ylim=c(0, 150))
segments(x0=x[1], y0=hyb.ci[1], y1=hyb.ci[2], lwd=2)
segments(x0=x[2], y0=hse.ci[1], y1=hse.ci[2], lwd=2)
```

3.5 Compare samples with t- and F-Test

Hypotheses:

Null: Both samples have the same mean width and variance.

Alternative: The mean width (and possibly also the variance) differ because of more subjective sampling of HSE students. They may have prefered bigger or the nice small leaves.

```
t.test(width ~ group, data = leaves)
```

Perform also the classical t-test (`var.equal=TRUE`) and the F-test (`var.test`). Calculate absolute and relative effect size (mean differences) and interpret the results of all 3 tests.

```
# var.test(...)
# t.test(...)
# ...
```

4 Appendix

The following is purely optional for all who feel underchallenged or just want to learn more.

4.1 Calculation of summary statistics with dplyr

```
library("dplyr")
leaves <- read.csv("leaves.csv")

stats <-
  leaves |>
    group_by(group) |>
    summarize(mean = mean(width), sd=sd(width), N=length(width), se=sd/sqrt(N),
              lwr = mean + qt(p = 0.025, df = N-1) * se,
              upr = mean + qt(p = 0.975, df = N-1) * se
             )

stats
```

4.2 Barchart and errorbars with ggplot2

```
library("ggplot2")
stats |>
  ggplot(aes(x=group, y=mean, min=lwr, max=upr))  +
    geom_col() + geom_errorbar(width=0.2)
```

4.3 A footnote about prediction intervals

The simplified $\bar{x} \pm z \cdot s$ formula used above is an approximation. A statistically rigorous 95% prediction interval, especially for smaller samples, needs two corrections.

First, we would use the **t-distribution** with the quantile $t_{\alpha/2,n-1}$ (or `qt(alpha/2, n-1)` in R) instead of the normal quantiles (± 1.96). Then we add a term $\sqrt{1 + 1/N}$ that corrects for the sample parameters. The full formula for a single future observation is then:

$$\mathrm{PI} = \bar{x} \pm t_{\alpha/2,n-1} \cdot s \cdot \sqrt{1 + \frac{1}{N}}$$

The prediction interval is related to the so-called “tolerance interval”. Both are the same if the population parameters μ, σ are known or the sample size is very big. However, there are theoretical and practical differences in case of small sample size.

References

R Core Team. (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. https://www.R-project.org/