03-Distribution and confidence intervals of maple leaf samples

Thomas Petzoldt

2024-10-29

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 was, 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 or box plots. We use only the leave width as an example, analysis of the other statistical parameters 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 strategy:

- 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 millimeter with a ruler (Fig. 2) and the data collected in a spreadsheet table and a csv-file.

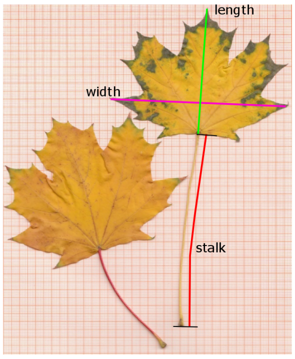

Figure 2: Fig 2.: Sample measures of maple leaves. Note that the stalk length does not include length of the leaf blade.

The statistical analysis is performed with the **R** software for statistical computing and graphics (R Core Team, 2024), version 4.4.1

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://raw.githubusercontent.com/tpetzoldt/datasets/main/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)
```

- split table for HSE and MHYB:

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

- compare leaf **width** of both groups:

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

3.2 Check distribution

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

3.3 Sample statistics

If we assume normal distribution of the data, we can estimate an approximate prediction interval from the sample parameters, i.e. in which size range are 95% of the leaves **SAMPLE** of one group. 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 the two-sided 95% prediction interval for the sample, assuming normal distribution:

```
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 of the normal distribution instead, e.g. `qnorm(0.975)`for the upper interval or `qnorm(c(0.025, 0.975))` for the lower and upper.

If the data set is large enough, we can compare the prediction interval from above with the **empirical quantiles**, i.e. take it directly from the data. Here we do not assume a normal or any other distribution.

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

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

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

… and the same as histogram:

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

3.4 Confidence interval of the mean

The **confidence interval** 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.

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

- Calculate the confidence interval of the mean value of the "hse" data set,
- use +/- 1.96 or (better) the quantile of the t-distribution:

```
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.

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

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 interval. This can be done with the add-on package **gplots** (not to be confused with **ggplot**) or with some creativity

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")
)
```

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.

```r
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.

```r
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: Summary statistics and CI with tidyverse

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)
```

References

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