05-Distribution and Confidence Intervals of Clementines

2025-12-16

1 Introduction

A given data set consists of two samples of clementine fruits. The experimental hypotheses is, that weight and size of two samples of clementines differ. The result is to be visualized with bar charts or box plots. We use the weight as an example, analysis of the other statistical parameters is left as an optional exercise.

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

We can set up 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 Clementines, either from different shops or different brands. Weight, width and height of the fruits were measured with a scale and a caliper.

The statistical analysis is performed with the **R** software for statistical computing and graphics (R Core Team, 2024) and the packages **dplyr** (Wickham et al., 2023) and **ggplot2** (Wickham, 2016):

```
library("dplyr")   # for pipelines, group_by and summarize
library("ggplot2") # modern plotting package "grammar of graphics"
```

3 Data Analysis

3.1 Prepare and inspect data

- Download the data set `fruits-2022.csv` and use one of RStudio's "Import Dataset" wizards.
- A better alternative is to use `read.csv()`.
- The data set is available in OPAL[1] or from: https://tpetzoldt.github.io/datasets/data/fruits-2022.csv
- The data set contains several subsets.

```
#  ... do it
# fruits <- ...
# fruits <- filter(brand %in% c("box1", "box2"))
```

- plot everything, just for testing:

```
plot(fruits)
```

We first split the overall data frame for box1 and box2 in two separate data frames, so we can analyse one after the other:

```
sample1 <- subset(fruits, brand == "box1")
sample2 <- subset(fruits, brand == "box2")
```

First, let's compare the weight of both groups:

```
boxplot(sample1$weight, sample2$weight, names=c("sample1", "sample2"))
```

Here it is easier to use `boxplot` with the model formula syntax. This is the preferred way, because it does not require to split the data set beforehand:

```
boxplot(weight ~ brand, data = fruits)
```

A third option would be to use **ggplot**:

```
fruits |>
  ggplot(aes(brand, weight)) +
  geom_boxplot()
```

3.2 Check distribution

We can check the shape of distribution graphically. We do it for now only for the "sample1" subset, because it contains more data.

Histogram

```
hist(... ... ...)

qqnorm(... ... ...)
qqline(... ... ...)
```

[1] OPAL is the learning management system used at TU Dresden.

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 we find 95% of the weights within one group.

We first calculate mean, sd, N and se for "sample1" data set:

```
sample1.mean <- mean(sample1$weight)
sample1.sd   <- sd(sample1$weight)
sample1.N    <- length(sample1$weight)
sample1.se   <- sample1.sd/sqrt(sample1.N)
```

Then let's 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.

```
sample1.pi <- sample1.mean + c(-1.96, 1.96) * sample1.sd
sample1.pi
```

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:

```
sample1.pi <- sample1.mean + qnorm(c(0.025, 0.975)) * sample1.sd
sample1.pi
```

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

```
plot(sample1$weight)
abline(h = sample1.pi, col="forestgreen")
```

... and the same as histogram:

```
hist(sample1$weight)
abline(v = sample1.pi, col="forestgreen")
rug(sample1$weight, col="blue")
```

Task: Count the number of points in the scatterplot and the histogram that are outside of the lines. Which percentage would you expect?

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 "sample1" data

Task: Calculate the confidence interval of the mean value of the "sample1" data set using the quantile function (`qt`) of the t-distribution[2]:

```
sample1.ci <- sample1.mean + qt(p = c(0.025, 0.975), df = sample1.N - 1) * sample1.se
```

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

```
abline(v = sample1.ci, col="magenta")
```

3.4.2 Add density plots

```
hist(sample1$weight, probability = TRUE, ylim = c(0, 0.25))
xnew <- seq(50, 100, length = 500)
lines(xnew, dnorm(xnew, sample1.mean, sample1.sd), col = "forestgreen")
lines(xnew, dnorm(xnew, sample1.mean, sample1.se), col = "magenta")
abline(v = sample1.pi, col="forestgreen")
abline(v = sample1.ci, col="magenta")
```

4 Compare samples with the t-Test

Null Hypothesis: Both samples have the same mean weight.

Alternative: The mean weight differs. The fruits bought in different shops or at different days are expected to differ in quality and size.

To supply the data to the `t.test`-function, we have again two options. One can either supply the two different groups directly:

```
t.test(sample1$weight, sample2$weight)
```

or use a model formula (`weight ~ groups`). Here we have to make sure, that the data set contains only the two grous of interest:

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

```r
fruits2 <- filter(fruits, brand %in% c("sample1", "sample2"))
```

```r
t.test(weight ~ brand, data = fruits2)
```

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.

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

5 Summary statistics and CI with tidyverse

The following shows how to calculate summary statistics in a more modern and efficient way.

The approach uses the **dplyr** and **ggplot2** packages from the so-called **tidyverse** family of packages Wickham et al. (2023). Furthermore, we use the pipeline operator `|>`, that transfers the output of one data manipulation step directly to the next.

Some slides about the use of pipelines can be found under https://tpetzoldt.github.io/elements/

5.1 Calculation of summary statistics with dplyr

Summarizing can be done with two functions, `group_by` that adds grouping information to a data frame and `summarize` to calculate summary statistics. In the following, we use the full data set with all groups.

```r
stats <-
  fruits |>
    group_by(brand) |>
    summarize(mean = mean(weight), sd = sd(weight),
              N = length(weight),  se = sd/sqrt(N),
              pi.lo = mean + qt(p = 0.025, df = N-1) * sd,
              pi.up = mean + qt(p = 0.975, df = N-1) * sd,
              ci.lo = mean + qt(p = 0.025, df = N-1) * se,
              ci.up = mean + qt(p = 0.975, df = N-1) * se
             )

stats
```

5.2 Barchart and errorbars with ggplot2

We can then use the table of summary statistics directly for a bar chart.

```r
library("ggplot2")
stats |>
  ggplot(aes(x = brand, y = mean, min = ci.lo, max = ci.up))  +
  geom_col() + geom_errorbar()
```

5.3 Additional tasks

Repeat the analysis with other properties of the fruits, e.g. width and height. Create box plots, analyse distribution, create bar charts.

5.4 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:

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

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/

Wickham, H. (2016). *ggplot2: Elegant graphics for data analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org

Wickham, H., Averick, M., Bryan, J., Chang, W., McGowan, L. D., François, R., Grolemund, G., Hayes, A., Henry, L., Hester, J., Kuhn, M., Pedersen, T. L., Miller, E., Bache, S. M., Müller, K., Ooms, J., Robinson, D., Seidel, D. P., Spinu, V., … Yutani, H. (2019). Welcome to the tidyverse. *Journal of Open Source Software*, *4*(43), 1686. https://doi.org/10.21105/joss.01686

Wickham, H., François, R., Henry, L., Müller, K., & Vaughan, D. (2023). *Dplyr: A grammar of data manipulation*. https://CRAN.R-project.org/package=dplyr