Processing and Analysis of Biological Data

Preface and summary statistics

Øystein H. Opedal

3 Nov 2023

Preface

These lectures notes are written for the course BIOS14 at Lund University. They will cover what I consider to be necessary (or at least useful) knowledge of data analysis and quantitative methods in ecology, evolutionary biology, and related fields. This document is work-in-progress and is not comprehensive. I therefore strongly encourage the reader to consult the many excellent text books available on 'biostatistics', R, quantitative biology etc.

Beyond these lecture notes, the main course literature is the book

Gerry P Quinn and Michael J Keough. 2002. Experimental Design and Data Analysis for Biologists, Cambridge University Press, ISBN: 9780521009768.

Like most biostatistics texts this is a reference in which to look up specific topics, rather than a book to be read chapter by chapter. Personally I have also found Michael Crawleys´ brick 'The R Book' useful in the past, it's as much a statistics text as an R text.

What separates this document (and this course) from 'typical' texts in biostatistics is the limited focus on statistical hypothesis testing. Though we will discuss key aspects of hypothesis testing in the traditional sense, focus will be throughout on quantification, parameter estimation, and biological interpretation of results. This choice is made in an attempt to mitigate an issue that occurs frequently in scientific writing, namely that the presentation of results in biological publications has increasingly tended towards a focus on statistical hypothesis testing ('significance testing'), at the cost of focus on the biological significance of any result obtained. An example is sentences on the form 'the treatment significantly ($P < 0.05$) affected the growth of plants' in place of an appropriate sentence such as 'plants in the high-fertilizer treatment grew 40% larger than those in the low-fertilizer treatment'. There is now a growing focus on turning this trend (Wasserstein & Lazar 2016), and this must start with the way statistics are taught in biostatistical courses such as this one.

Wasserstein, R. L., and N. A. Lazar. 2016. The ASA Statement on p-Values: Context, Process, and Purpose. The American Statistician 70: 129-133.

Beyond quantification and data analysis, the following chapters will also introduce some basic data handling, scientific programming, and graphics. These skills are by now essential for most practicing researchers, and I like to think the skills will be useful also for those that put their Biology education to use outside of academia.

Measurement and meaning in biology

Measurement theory is concerned with the relationship between measurements and the theoretical context in which they will function. In other words, what are the numbers we record meant to represent? The title of this section is shared with an important paper published in 2010 by evolutionary biologist David Houle

and colleagues, in which they pointed out the common misconnect in biology between the measurements taken in the field or laboratory, and the biological properties they are meant to represent. This exemplifies a more general problem across biology, where the interpretation of measurements and analyses fails to focus on the biological questions that motivated the study in the first place. Before reading on, I strongly suggest to read at least the first pages of Houle et al. 2010, to get a grasp of the problem at hand.

Scale types and transformations

The concept of scale types has rarely been taught in biology, yet any quantitative measurement is placed on a specific scale type, with some associated properties such as permissible transformations. The perhaps most familiar scale type is the *ratio scale*, represented for example by any linear size measurement (with units such as *mm* or *m*). Importantly, the ratio scale has a natural zero, and negative values do not occur.

In contrast, consider the measurement of days since January 1st (often used to record for example breeding dates of birds). Dates are on an *interval scale*, and lacks a natural zero (i.e., the zero is chosen arbitrarily). This has consequences, for example, for how we compare raw and proportional variances among groups (see below). On this topic, it is common to see the number of days since January 1st labelled as a 'Julian date'. I write this on the 20th of October 2022, and the Julian date is 22293 (the number of days since the beginning of the Julian period).

A third common scale type in biology is the *ordinal scale*, where measurements are in order (e.g. 2 is larger than 1), but the intervals between values are not fixed (i.e. the difference between 1 and 2 may differ from the difference between 2 and 3). Examples include where species are scored within a plot as "dominant", "common", "rare", and "very rare". Most will agree that "common" > "rare", but it does not make sense to translate these categories into numerical values (e.g. 4 for "dominant", 3 for "common" etc.) and then compute the mean.

When measurements are categories (e.g. red vs. blue vs. green), they are on a *nominal scale*.

Summary statistics

Before departing on any data analysis, it is advisable to explore the data at hand both graphically, and by obtaining relevant summary statistics. Summary statistics are those that aim to describe the properties of the data. For example, the central tendency of a dataset is generally measured by the arithmetic mean (typically denoted μ or $\bar{x}$), median, or, less frequently, the mode. Which of these to choose depends in part on the distribution of the data. If the data are skewed (tendency towards large or small values), the mean can give a misleading picture of the central tendency. The same issue arises if there are extreme values (outliers) that will tend to affect the mean much more than they affect the median. In some cases, a good option can be to report both the mean and the median, which together will give a more complete impression of the data distribution. For ordinal or nominal variables, the mode may be a suitable option.

Another measure of central tendency that buffers the influence of large values is the geometric mean $(\Pi_{i=1}^{n} x_i)^{\frac{1}{n}}$. The geometric mean is equal to the exponent of the mean of the natural logs of the data. The geometric mean increases towards the arithmetic mean (here 20, solid line) with decreasing variance in the data. The slight scatter in the plot below arises from the use of a randomly generated dataset to generate each datapoint.

One application of the geometric mean in biology is the evaluation of the long-term success of a given (evolutionary) strategy, such as the fitness consequences of variance-reducing strategies such as bet-hedging. A bet-hedging strategy is sometimes defined as one that maximizes the geometric rather than arithmetic mean fitness (by reducing e.g. the among-year variance in reproductive success).

Variation in the data is typically described by the variance (typically denoted σ^2) or its square root, the standard deviation ($\sqrt{\sigma^2} = \sigma$). As pointed out by R. A. Fisher, the advantage of the variance (mean squared deviation from the mean) is that variance components are additive, allowing us to partition the total variance

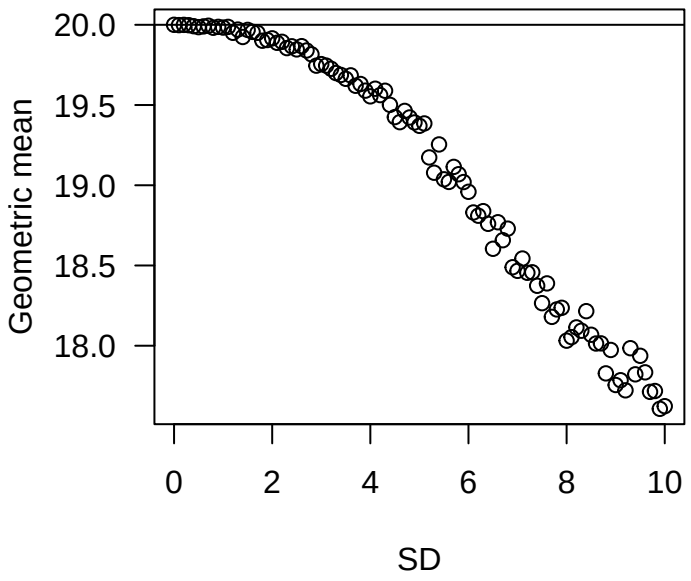


Figure 1: Relationship between arithmetic (solid line) and geometric mean with increasing variance in the data

in the data into different components (see more on Variance partitioning below). To understand why square values are additive, recall Pythagoras theorem

$c^2 = a^2 + b^2$, but $c \neq a + b$.

The standard deviation measures the mean deviation of each data point from the mean, and thus has the advantage of having the same units as the original measurements. If we have measured a trait in *mm*, the standard deviation gives the average deviation in *mm* from the mean, and is thus easy to interpret.

Note that the variance and standard deviation are measures of dispersion in the data, *not* the certainty of an estimate. Therefore, unlike *standard errors* ($SE = SD/\sqrt{n}$), the standard deviation should not be given with the $\pm$ sign. This mistake is very frequent in papers in biology.

Very often we are interested in measuring the proportional variation in a variable. The idea is that, because larger things are more variable than smaller things, we may want to compare how variable different entities are, *proportional to their mean*. One common measure is the coefficient of variation (CV), defined as

$CV = \frac{\sigma}{\mu}$

The CV is often miscalculated in published studies, so keep you eyes open. Because the standard deviation σ is the square root of the variance σ^2, the CV is often written as

$CV = \frac{\sqrt{\sigma^2}}{\mu}$

which is sometimes misinterpreted or even mistyped by the journal typesetter as

$CV = \sqrt{\frac{\sigma^2}{\mu}}$.

Another method for placing data on a proportional scale is taking natural logarithms. The standard deviation of a log-transformed variable is very similar to the standard deviation on arithmetic scale divided by the trait mean (the CV), as long as the variance is much smaller than the mean.

Another way to see the nice proportional properties of the log-scale is to recall that $log(\frac{a}{b}) = -log(\frac{b}{a})$

```
log(1.1/1)
```

```
## [1] 0.09531018
```

```
-log(1/1.1)
```

```
## [1] 0.09531018
```

Note also that when a (1.1) is 10% larger than b (1.0), the *log ratio* is ~0.1. Recall that $log(\frac{a}{b}) = log(a) - log(b)$, and thus differences on log scale multiplied by 100 are roughly interpretable as difference in percent (when a and b are not very different). Log ratios are therefore often good measures of effect size.

Simulating data from statistical distributions

Throughout this course we will use simulated data to illustrate concepts and ideas, and to understand how statistical models work. `R` has built-in functions for simulating data from many statistical distributions, including the normal distribution. The function `rnorm()` takes three main arguments, the number of samples `n`, the mean, and the standard deviation `sd`.

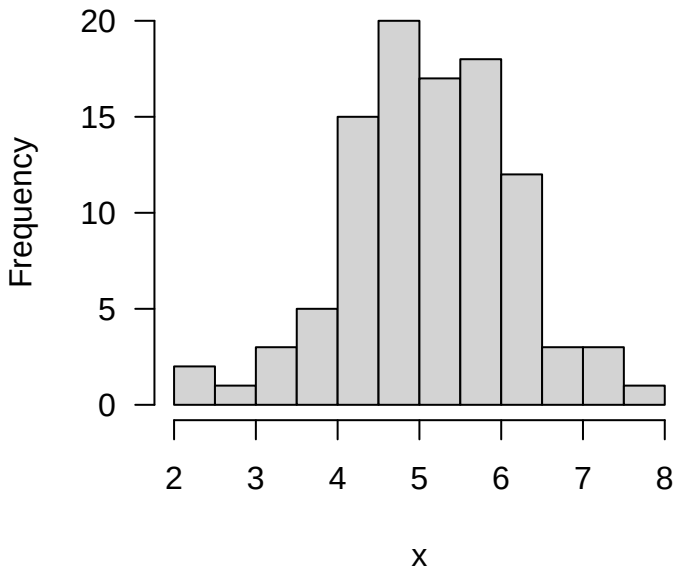
```
x = rnorm(n=100, mean=5, sd=1)
mean(x)
```

```
## [1] 5.102443
```

```
sd(x)
```

```
## [1] 1.05942
```

```
hist(x, las=1, main="")
```



R functions know the position of arguments, so that `rnorm(100, 5, 1)` will give the same result as above.

Bootstrapping

Bootstrapping is a common resampling technique used to assess uncertainty in variables. This can be especially relevant when the variable of interest is already a summary of a statistical population, such as a coefficient of variation (CV). As a first example, we will use bootstrapping to obtain the standard error of the mean, which we have already seen is given theoretically by $SE = \sqrt{Var/n}$.

To ensure reproducibility of the results (i.e. that we will get the same result every time, even when we work on different computers), we set the "seed" of the random number generator using the `set.seed()` function.

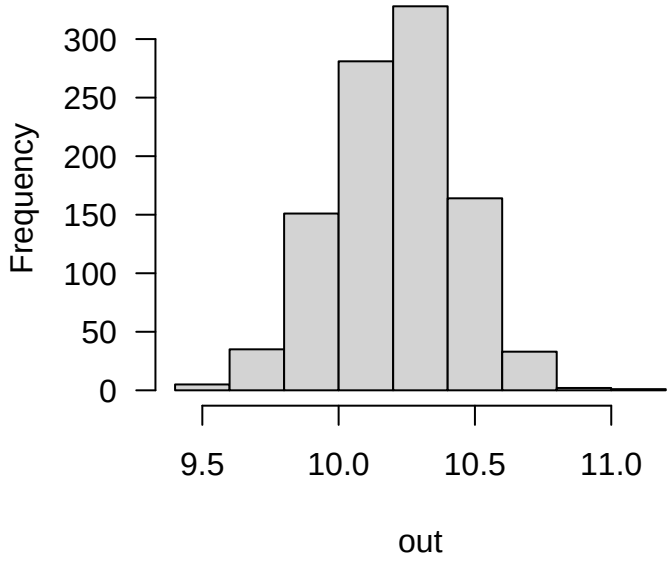
```
set.seed(1)
x = rnorm(50, 10, 2)
se_x = sqrt(var(x)/length(x))
```

We will do a non-parametric bootstrap, where we resample the data many times. For each resampling, we maintain the original sample size (here 50), but we draw from the data with replacement, so that by chance some values will be sampled several times and others not at all. Before running the *for*-loop, we have to define the variable `out` that we will use to store the results for each iteration of our loop.

```
out = NULL
for(i in 1:1000){
  sample = sample(x, replace=TRUE)
  out[i] = mean(sample)
}
```

The variable `out` now contains what we can call the *sampling distribution* of the mean of x. The standard deviation of the sampling distribution gives an approximation of the standard error.

```
hist(out, las=1, main="")
```




```
sd(out)
```

```
## [1] 0.2307834
```

As expected, this is close to the theoretical standard error

```
se_x
```

```
## [1] 0.2351537
```

Given that we also have the full sampling distribution, we can also choose to derive some quantiles, such as a 95% confidence interval.

```
quantile(out, c(0.025, 0.975))
```

```
##      2.5%     97.5%
##  9.760249 10.624404
```

Recall that we could also have derived the 95% confidence interval analytically as $\pm 1.96SE$. This follows from the properties of the *standard normal distribution* (with a mean of zero and a variance of one), for which the 2.5 and 97.5 percentiles falls ~1.96 standard deviations from the mean. The quantiles of the standard normal distribution are available in `R` through the `qnorm` function. The same is true for other probability distributions, e.g. `qbinom` for the binomial distribution etc.

```r
qnorm(c(0.025, 0.975))
```

```
## [1] -1.959964  1.959964
```

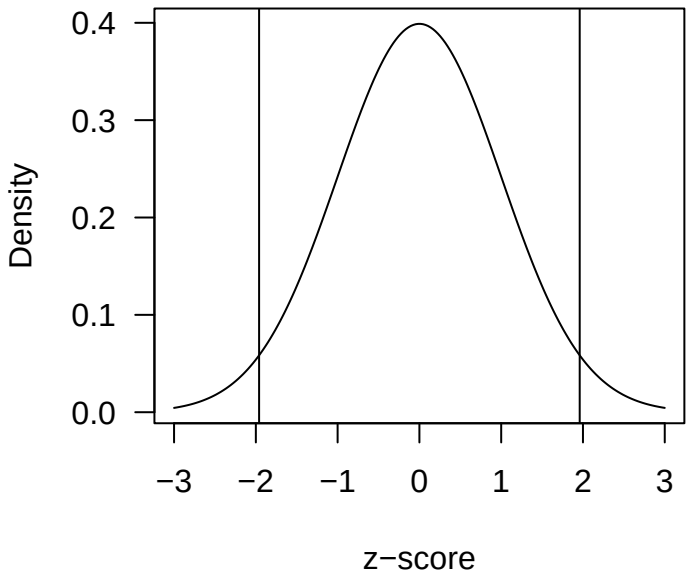


Figure 2: Density of the standard normal distribution, with veritical lines illustrating 95% of the distribution

```r
mean(x) - 1.96*se_x
```

```
## [1] 9.739995
```

```r
mean(x) + 1.96*se_x
```

```
## [1] 10.6618
```

EXERCISE: Use non-parametric bootstrapping to derive a 95% confidence interval for the CV of x.

Optional exercise: The proportional properties of the natural log

Use simulated data to show the close relationship between the SD of log-transformed data and the *CV* on arithmetic scale. You may need e.g. the `rnorm` function and a *for*-loop to achieve this. One strategy would be to start with comparing the two values for a single case, then build a matrix to hold the paired values, and finally use a *for*-loop to populate the matrix. See Appendix 1 for help to get started with programming. The following figure illustrates the kind of pattern we expect.

Before starting to work on exercises in R, read the Appendix on reproducibility, and think about whether you would like to create a GitHub account for all your materials from this course.

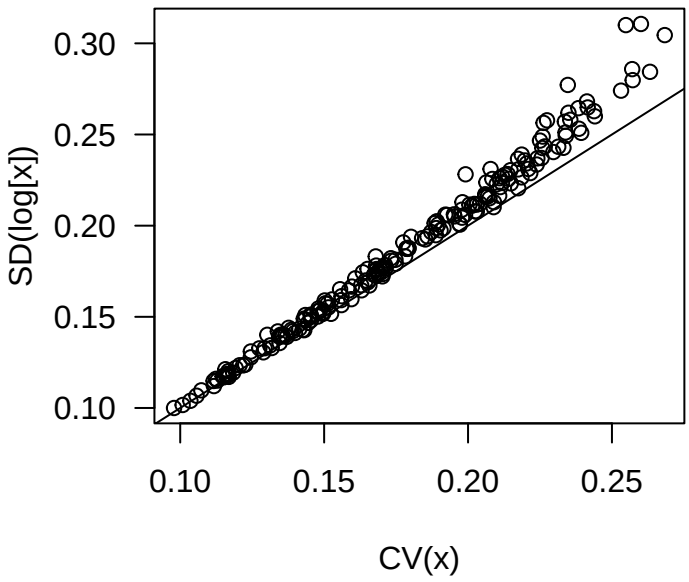


Figure 3: Similarity of the standard deviation of log-transformed data to the coefficient of variation