Statistics with Sparrows - 04

Julia Schroeder

2022

HO 04

Precision and standard error

Learning aims

- To understand the link between the standard error of the mean, and the sample size
- To understand that standard errors belong to a statistic (typically a mean)
- To understand that standard errors are a measure of precision

Standard errors

Standard errors (se) are a good way to display uncertainty. You can calculate them using this equation:

$$se = \sqrt{\frac{s^2}{n}}$$

Let's do this for Tarsus in our dataset. Some basic housekeeping to start with:

```
rm(list=ls())

d<-read.table("SparrowSize.txt", header=TRUE)
d1<-subset(d, d$Tarsus!="NA")
seTarsus<-sqrt(var(d1$Tarsus)/length(d1$Tarsus))
seTarsus

## [1] 0.02096211
```

and now only for 2001:

```
d12001<-subset(d1, d1$Year==2001)
seTarsus2001<-sqrt(var(d12001$Tarsus)/length(d12001$Tarsus))
seTarsus2001

## [1] 0.1030623
```

The SE of 2001 is rouhgly five times the one for the total population. That's shocking! So, what determines how larger or small our measurement of precision, the SE is? Let's have a closer look at it:

$$se = \sqrt{\frac{s^2}{n}}$$

with a bit of algebra we find out this is the same as

$$se = \frac{s}{\sqrt{n}}$$

The important bit in this part is to have a good look at the denominator. N represents sample size. s^2 is, as we know, the variance. Thus, if se is an indicator of the uncertainty, then what do we have to do to make this number the smallest possible? We have to increase the sample size! We can do some basic math to find out how much larger our sample size needs to be to half our standard error:

$$\frac{se}{2} = \frac{s}{2\sqrt{n}} = \frac{s}{\sqrt{4n}}$$

This is the square root law of sample size: to improve your precisision by doubling it, you need to increase your sample size by it's squared term: 4!

Another reason why I showed you these equations for the standart error is that it is not only related to variance, but also to our 95% confidence interval. Remember that? Here is the equation to calculate the 95% confidence interval. It is also a measure of precision. Can you spot the standard error?

$$CI_{95\%} = \pm 1.96 \frac{s}{\sqrt{n}}$$

or

$$CI_{95\%} = \pm 1.96 se$$

Note that this works best for large sample sizes (approx. N>50).

Visualizing a change in precision

Let's see if we can visualise this in an appealing way, as I've shown briefly in the lecture. The grand idea now is to produce a plot that shows how the mean varies with increasing sample size, and how the accompanying standard errors shrink with sample size. Many times in statistics, it is useful to first generate a dataset with known quantities, and then run the analyses on it as a test of how it would turn out. To do that, we have to make up some data:

I will use a dataset that I create myself - that way I know what the "true" mean is. I will simulate a dataset of golden glamour dragons' tail lengths. But first I will clear the workspace:

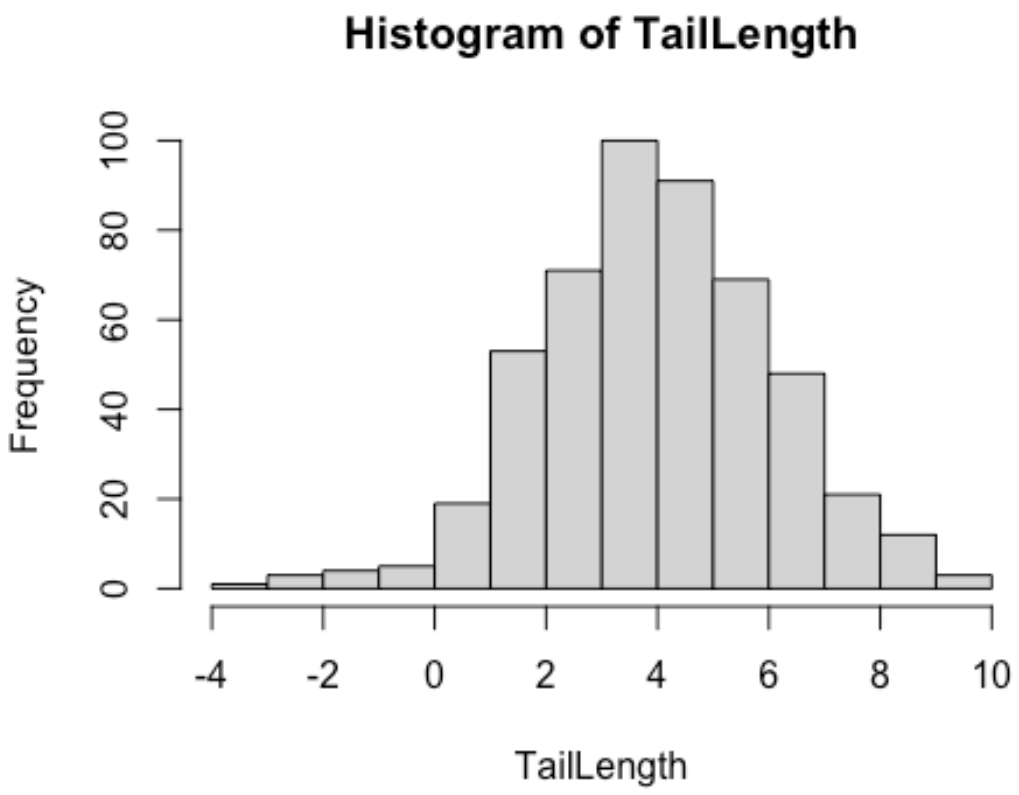
```
rm(list=ls())
```

Then I will create a vector with 500 entries of measurements of dragon tail lengths. I want a mean of 3.8 meter, and a standard deviation of 0.75:

```
TailLength<-rnorm(500,mean=3.8, sd=2)
```

Check out the function `rnorm()` if you haven't done that yet. It's very useful in many situations. This function creates randomly drawn values from a normal distribution, and you can say what mean and sd you want. Next, I'll check out the data, see if it fits my expectations, and plot it:

```
summary(TailLength)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -3.403   2.606   3.939   3.948   5.350   9.964

length(TailLength)

## [1] 500

var(TailLength)

## [1] 4.284017

sd(TailLength)

## [1] 2.069787

hist(TailLength)
```

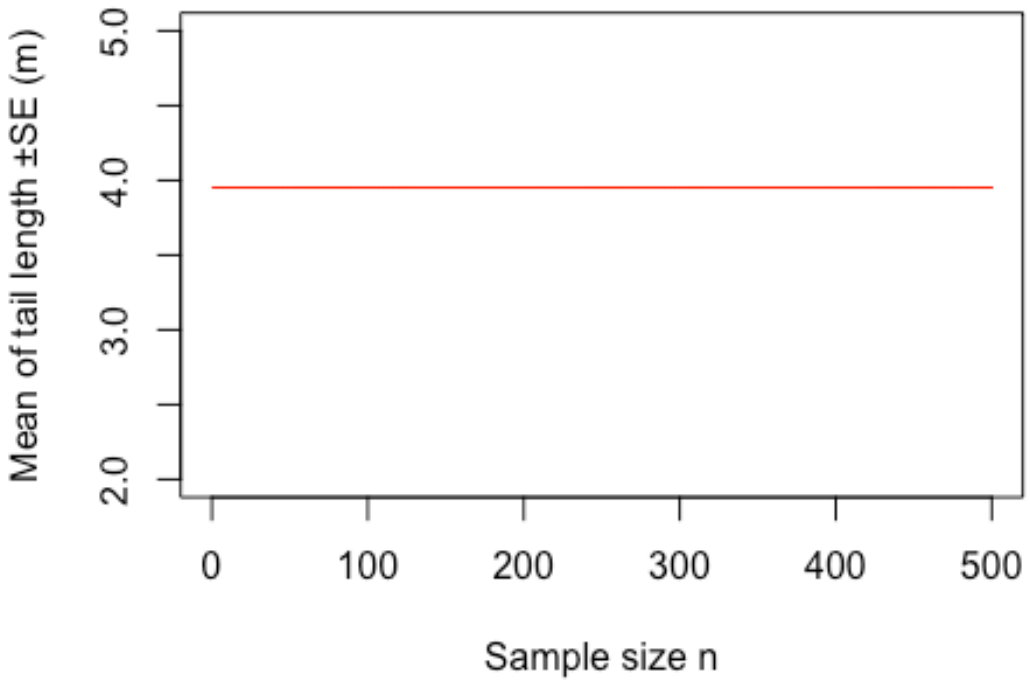


That's reasonable - the mean is 3.79, and the standard deviation is 0.71. *Note that these numbers may look different in your console - that is because `rnorm()` draws random numbers, so of course it will not be exactly the same - that's the whole point of it.*

Now to the real exercise. I want to randomly draw a specified number of observations from this dataset, and calculate the mean, and standard error, and plot it. I actually want to do that for sample sizes from one all the way up to 500. To do that I have multiple options. I chose here a classical, traditional, `for` loop, and plot the points while I'm looping. I first prepare the canvas of the plot. To do that, I need to know the maximum and minimum values that need to be displayed. Obviously, the x-axis will run from up to 500. The y-axis will run from the minimum to the maximum mean, and I will give it some space for the standard error bars. I want to plot the grand total mean - as a black line so I can compare the other means to it. To do that I need to create a dataset with a vector for x that runs from 1 to 500, and a y vector that holds 500 times the grand total mean. I can create y in many ways, but by multiplying it with x I make sure they are both of the same length.

```
x<-1:length(TailLength)
y<-mean(TailLength)+0*x
min(TailLength)

## [1] -3.403074

max(TailLength)
```

```
## [1] 9.964022

plot(x,y, cex=0.03, ylim=c(2,5),xlim=c(0,500), xlab="Sample size n", ylab="Me
an of tail length ±SE (m)", col="red")
```



Now I need to populate my graph with means of samples of this data. To do this, I run the `for` loop. But before I do that I make vectors for the means (`mu`) and the standard errors (`SE`):

```
SE<-c(1)
SE

## [1] 1

mu<-c(1)
mu

## [1] 1
```

These two vectors will be filled with the data. Having a vector allows me to use the elements of it - if I want to fill the first element with the mean of a sample size of n=1, I can say `SE[1]<-`. Now when I use a placeholder for the sample size, in this case, `n`, I can loop through it and fill up the vectors one by one:

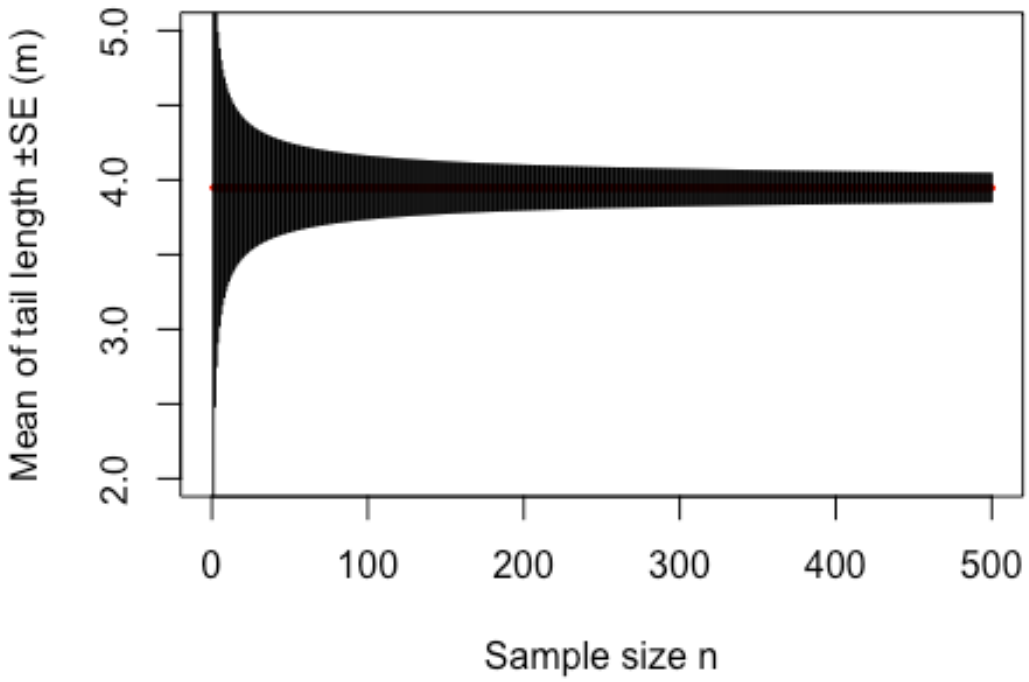
```
for (n in 1:length(TailLength)) {
  d<-sample(TailLength, n, replace=FALSE)
  mu[n]<-mean(TailLength)
  SE[n]<-sd(TailLength)/sqrt(n)
}
```

Now I want to first see if what I've done did what I wanted it to do:

```
head(SE)

## [1] 2.0697867 1.4635602 1.1949919 1.0348933 0.9256367 0.8449869

head(mu)

## [1] 3.948466 3.948466 3.948466 3.948466 3.948466 3.948466

length(SE)

## [1] 500

length(mu)

## [1] 500
```

That looks fabulous. Now I can add to the previous plot. I want to plot the new means on the y-axis, and the sample sizes (n) on the x-axis. Then I want to plot error bars that go from the means to mean±SE.

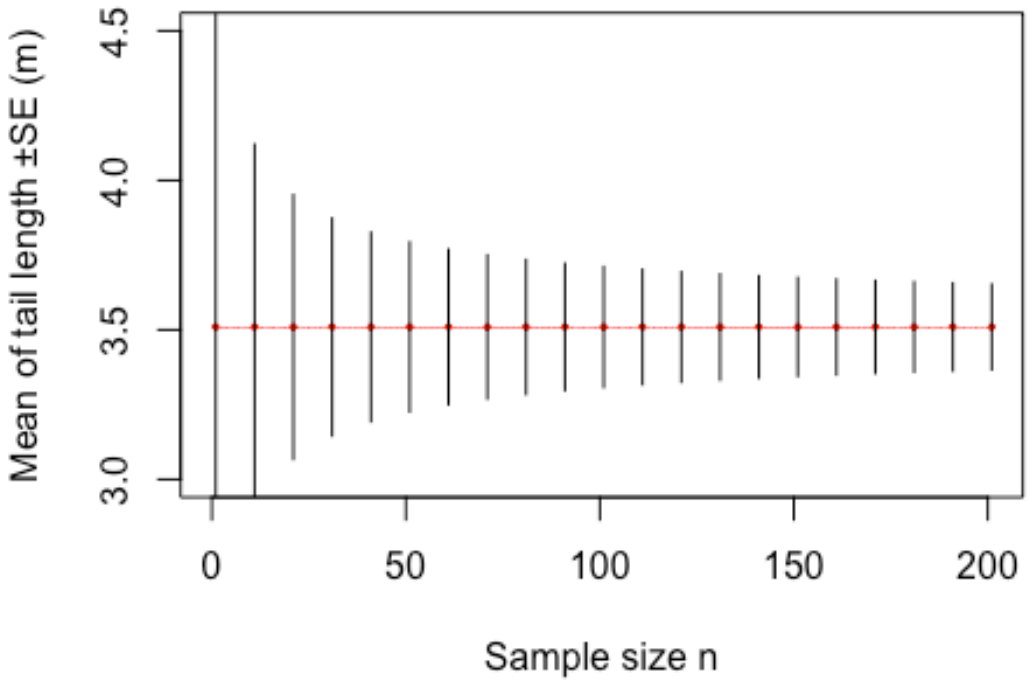
```
up<-mu+SE
down<-mu-SE
x<-1:length(SE)
segments(x, up, x1=x, y1=down, lty=1)
```



While this seems to be doing what it's supposed to do, it's not so nice on the eye. Maybe I need to thin it out - also, 500 seems to be a bit too long - 200 will do. I'll do 201 because then I can sample starting from 1, by 10:

```
rm(list=ls())
TailLength<-rnorm(201,mean=3.8, sd=2)
length(TailLength)

## [1] 201

x<-1:201
y<-mean(TailLength)+0*x
plot(x,y, cex=0.03, ylim=c(3,4.5),xlim=c(0,201), xlab="Sample size n", ylab="
Mean of tail length ±SE (m)", col="red")
n<-seq(from=1, to=201, by=10)
n

##  [1]   1  11  21  31  41  51  61  71  81  91 101 111 121 131 141 151 161 1
71 181
## [20] 191 201

SE<-c(1)
mu<-c(1)
for (i in 1:length(n)) {
```

```
  d<-sample(TailLength, n[i], replace=FALSE)
  mu[i]<-mean(TailLength)
  SE[i]<-sd(TailLength)/sqrt(n[i])
}
up<-mu+SE
down<-mu-SE
length(up)

## [1] 21

length(n)

## [1] 21

plot(x,y, cex=0.03, ylim=c(3,4.5),xlim=c(0,201), xlab="Sample size n", ylab="
Mean of tail length ±SE (m)", col="red")
points(n,mu,cex=0.3, col="red")
segments(n, up, x1=n, y1=down, lty=1)
```



And it becomes clear how the standart error shrinks with increasing sample size.

Exercises:

Easy: Calculate the standard error of Tarsus, Mass, Wing and Bill length of the complete population sample (as opposed to all sparrows in this world) Note N of each. Then, subset the dataset to only 2001 data `d1<-subset(d, d$Year==2001)`, as we did in the lecture. Calculate SE for Tarsus, Mass, Wing and Bill length for the 2001 sample. Calculate the 95% CI for each mean.

More difficult but important: Play around with the last plot we made. Change the values of the simulation. Change the mean and the standard deviation and see how it affects your plot. This should help you to get better at

- Understanding how mean and SE and sample size are linked
- Understanding how to plot things in R
- Understanding how to use for loops in R
- Understanding how to make, access and use variables, vectors and data frames in R

Note: it is ok if you find this difficult. This is something that you need to do, and do again, and redo again, and the more often you practice this the better you will become at understanding it. So use this time to repeat and practice.

Non-compulsory:

Difficult: If you found the above easy and want to test out your coding and stats skills, this is a task for you.

Sample mass of the whole sparrow dataset. Then produce a plot to see at what sample size the standard error becomes so small that you'd be confident to get a reasonably precise mean - precise with respect to the grand total mean of the whole dataset. (The next part is rather philosophical). Then look at sample sizes in different years, and think about what sort of biological questions one could answer with one year's data, and what not.