Stats with Sparrows - 2

Julia Schroeder

September 2016

Handout 2

Describing distributions

Let's get down to the meat: statistics! We will use the sparrow data as example. But before we begin, we clear our workspace. Never forget!

```
rm(list=ls())

setwd("~/Box Sync/Teaching/IntroStats")

d<-read.table("SparrowSize.txt", header=TRUE)
str(d)

## 'data.frame':    1770 obs. of  10 variables:
##  $ BirdID     : int  4401 4401 4405 4405 4405 4409 4409 4409 4409 4409 ...
##  $ Cohort     : int  1991 1991 1994 1994 1994 1994 1994 1994 1994 1994 ...
##  $ CaptureDate: Factor w/ 414 levels "01-Aug-06","01-Dec-07",..: 272 18
254 41 88 303 174 18 159 164 ...
##  $ CaptureTime: Factor w/ 293 levels "04:00","04:30",..: NA NA NA NA NA NA
NA NA NA NA ...
##  $ Tarsus     : num  18.9 18.8 19.1 19 19.1 ...
##  $ Bill       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Wing       : num  82 79 77 78 77 76 76 73 79 77 ...
##  $ Mass       : num  29.4 31.6 29.9 31.6 31 ...
##  $ Sex        : int  1 1 0 0 0 1 1 1 1 1 ...
##  $ Sex.1      : Factor w/ 2 levels "female","male": 2 2 1 1 1 2 2 2 2 2
...

names(d)

##  [1] "BirdID"      "Cohort"      "CaptureDate" "CaptureTime" "Tarsus"
##  [6] "Bill"        "Wing"        "Mass"        "Sex"         "Sex.1"

head(d)

##   BirdID Cohort CaptureDate CaptureTime Tarsus Bill Wing Mass Sex  Sex.1
## 1   4401   1991   21-Jun-00        <NA>   18.9   NA   82 29.4   1   male
## 2   4401   1991   02-Oct-00        <NA>   18.8   NA   79 31.6   1   male
## 3   4405   1994   20-Jun-00        <NA>   19.1   NA   77 29.9   0 female
## 4   4405   1994   04-Oct-00        <NA>   19.0   NA   78 31.6   0 female
```

```
## 5   4405   1994   07-Oct-00        <NA>   19.1   NA   77 31.0   0 female
## 6   4409   1994   23-Mar-00        <NA>   18.0   NA   76 28.1   1   male
```

What is NA?

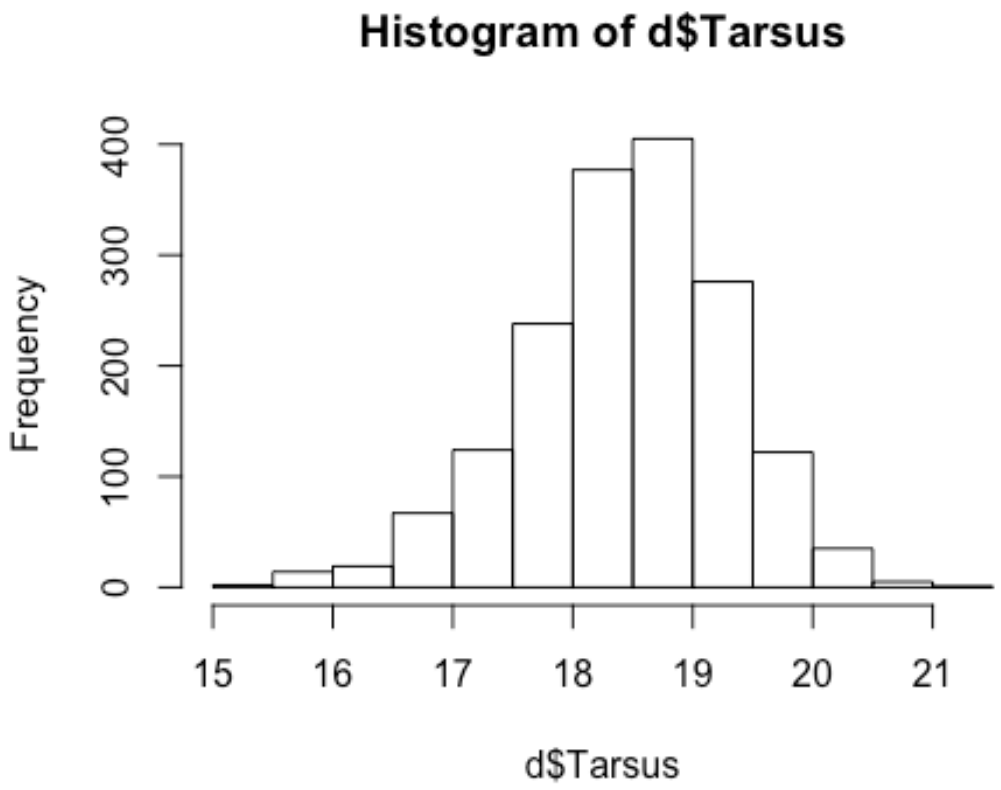
Let's check the distribution of the data! We usually do this with a histogram. For now, we will work with data on the length of a bird's tarsus, that's the leg of a bird. We have a lot of data to deal with here!

```
length(d$Tarsus)

## [1] 1770
```

So let's get cracking!

Histograms, mean, median and mode

```
hist(d$Tarsus)
```



This looks like a normal distribution, doesn't it? It might lean a bit to the right, though. What is a normal distribution? Many data that we collect will be expected to approximately follow a normal distribution. Therefore, many statistics, - more explicitely, parametric statistics, rely on the data being normally distributed. Now, that means, we should have a

way to find out IF our data is actually normally distributed... Luckily for us, there are a few tests we can use. We will get back to this later.

Now, how can we describe this distribution of values best? Usually, we use a description of centrality, and one of the variability. What is centrality?

Centrality, mean, median and mode in normally distributed data

```
mean(d$Tarsus)

## [1] NA
```

Uuups. We have missing values in our dataset (NAs). This is reflected by the error message. How to deal with this? A quick look up in help reveals that we can use an argument na.rm that is by default set to "FALSE". When set to TRUE, it means that NAs are stripped before computation. That's what we want. Let's give it a try:

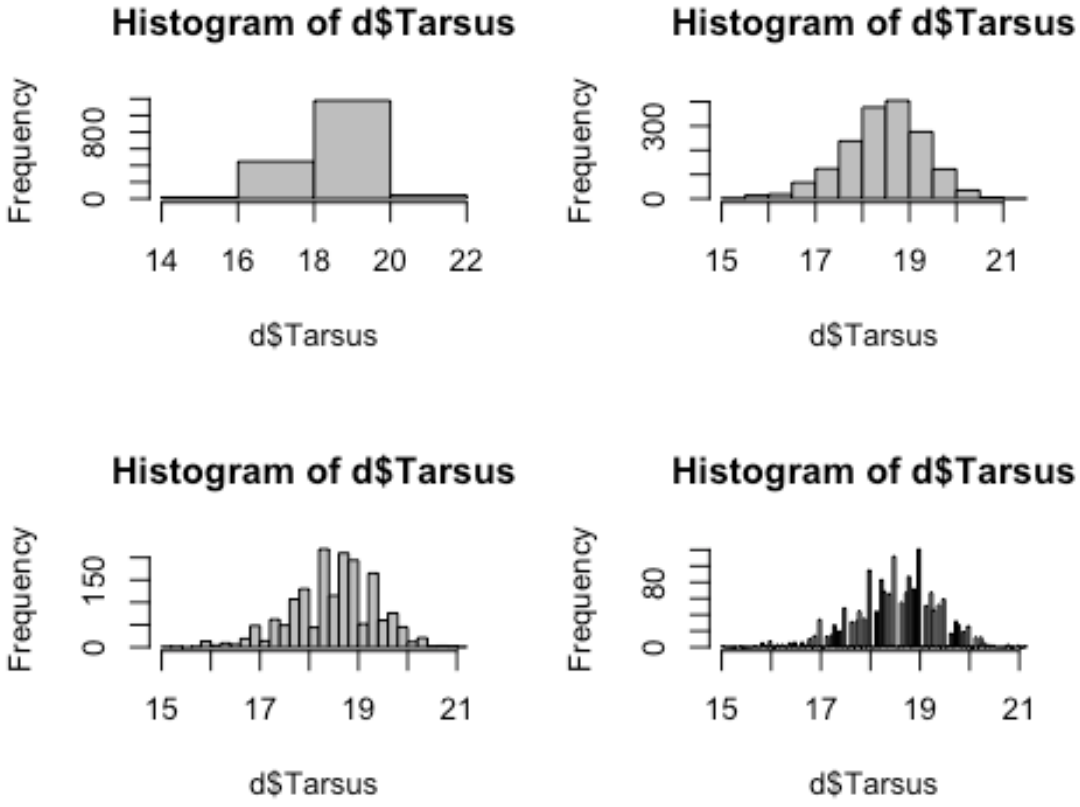
```
help(mean)
mean(d$Tarsus, na.rm = TRUE)

## [1] 18.52335

median(d$Tarsus, na.rm = TRUE)

## [1] 18.6

mode(d$Tarsus)

## [1] "numeric"
```

Now, this worked at least two of three times. What's with the odd result for mode? The mode function returns a description of the type of object. It tells us that d$Tarsus is a numerical vector. What does this mean? If we have continous data, we have a hard time estimating the mode. That is because the mode is the mmost frequently occuring value. Why do you think it is hard to estimate it in a continous dataset?

Because most values occur only once.

Let's play with the data a bit more and look how the distribution changes, and the (supposed) mode. Also, can you remember what par(mfrow=c(2,2)) does?

```
par(mfrow = c(2, 2))
hist(d$Tarsus, breaks = 3, col="grey")
hist(d$Tarsus, breaks = 10, col="grey")
hist(d$Tarsus, breaks = 30, col="grey")
hist(d$Tarsus, breaks = 100, col="grey")
```



Think about this. Why do you think there are these odd gaps? The number of breaks determines the number of bins that is used to draw the histogram. And at some point, the resolution is larger than the resolution of the measuring device! Clearly the mode is somewhere between 18 and 19. It would be nice to get this more precisely.

```
install.packages("modeest")

require(modeest)

## Loading required package: modeest

##
## This is package 'modeest' written by P. PONCET.
## For a complete list of functions, use 'library(help = "modeest")' or
'help.start()'.

?modeest
```

We quickly check out the help section, scroll down to the end, discover mlv. We can give this a try:

```
mlv(d$Tarsus)
```

Ups. We - again - ran into problems with the missing values. Sometimes it is just easier to recode the dataset we work with into one that doesn't contain NAs for tarsus.

```
d2<-subset(d, d$Tarsus!="NA")
length(d$Tarsus)

## [1] 1770

length(d2$Tarsus)

## [1] 1685
```

Ok. Now we know that there were quite some lines with missing values for tarsus. d2 is the dataset that does not contain those any longer.

```
mlv(d2$Tarsus)

## Warning in mlv.default(d2$Tarsus): argument 'method' is missing. Data are
## supposed to be continuous. Default method 'shorth' is used

## Warning in .deal.ties(ny, i, tie.action, tie.limit): encountered a tie,
and the difference between minimal and maximal value is > length('x') *
'tie.limit'
## the distribution could be multimodal

## Mode (most likely value): 18.57361
## Bickel's modal skewness: 0.0005934718
## Call: mlv.default(x = d2$Tarsus)
```

Ok. We get some warning messages, but we can live with that because we understand them! So, let's try this again:

```
mean(d$Tarsus, na.rm = TRUE)

## [1] 18.52335

median(d$Tarsus, na.rm = TRUE)

## [1] 18.6

mlv(d2$Tarsus)

## Warning in mlv.default(d2$Tarsus): argument 'method' is missing. Data are
## supposed to be continuous. Default method 'shorth' is used

## Warning in .deal.ties(ny, i, tie.action, tie.limit): encountered a tie,
and the difference between minimal and maximal value is > length('x') *
'tie.limit'
## the distribution could be multimodal

## Mode (most likely value): 18.57361
## Bickel's modal skewness: 0.0005934718
## Call: mlv.default(x = d2$Tarsus)
```

In normally distributed data, mean, median and mode should be fairly similar. If the distribution is perfectly normal, they should even be identical. As the skew of the distribution increases, these three measures diverge.

Range, variance and standard deviation

R is really good because you can guess the name of certain functions. We can just guess that "range" might give us the range of values in a vector.

```
range(d$Tarsus, na.rm = TRUE)

## [1] 15.0 21.1

range(d2$Tarsus, na.rm = TRUE)

## [1] 15.0 21.1

var(d$Tarsus, na.rm = TRUE)

## [1] 0.7404059

var(d2$Tarsus, na.rm = TRUE)

## [1] 0.7404059
```

Now, range is easy - that's the minimum and maximum values. But what exactly is the var, the variance?

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n-1}$$

So, the variance is described as σ^2. It should be apparent, from the lecture, why it's a square. Let's have a closer look at the formula. It includes the mean, but also other stuff. The numerator is called the sum of squares, and is sometimes denoted by SS. This is super important. Why is it a sum of squares? Let's a have a look at how to get this. It's the sum of the squares of the *deviations* from the mean.

Let's write out how to calculate this in R:

```
sum((d2$Tarsus - mean(d2$Tarsus))^2)/(length(d2$Tarsus) - 1)

## [1] 0.7404059
```

We use the tarsus bit without NAs as then the length is correct. Now, why is it denoted as σ^2? If we square-root the variance, we get σ, and that is the standard deviation:

```
sqrt(var(d2$Tarsus))

## [1] 0.8604684

sqrt(0.74)
```

```
## [1] 0.8602325
```

```
sd(d2$Tarsus)
```

```
## [1] 0.8604684
```

Cool. Now we understand how to describe data that is about normally distributed - with measures of centrality (mean, median, mode) and measures that describe the spread (range, variance, standard deviation).

Z-scores and quantiles

Z-values, and z transformation (z-score, or standardized scores) is a super important topic that you will encounter very often, and use very often. Z-scores come from a standardized normal distribution, with a mean of 0 and a standard deviation of 1. What variance does this distribution have?

When we do stats it is often useful to transform our data so it follows exactly these rule - z-transforming data. You can do that by deviding the deviation from the mean by the standard deviation:

$$z = \frac{y\text{-}\overline{y}}{s_y}$$

Here, s_y means standard deviation of y. Ok, let's do this, and check is all went according to plan:

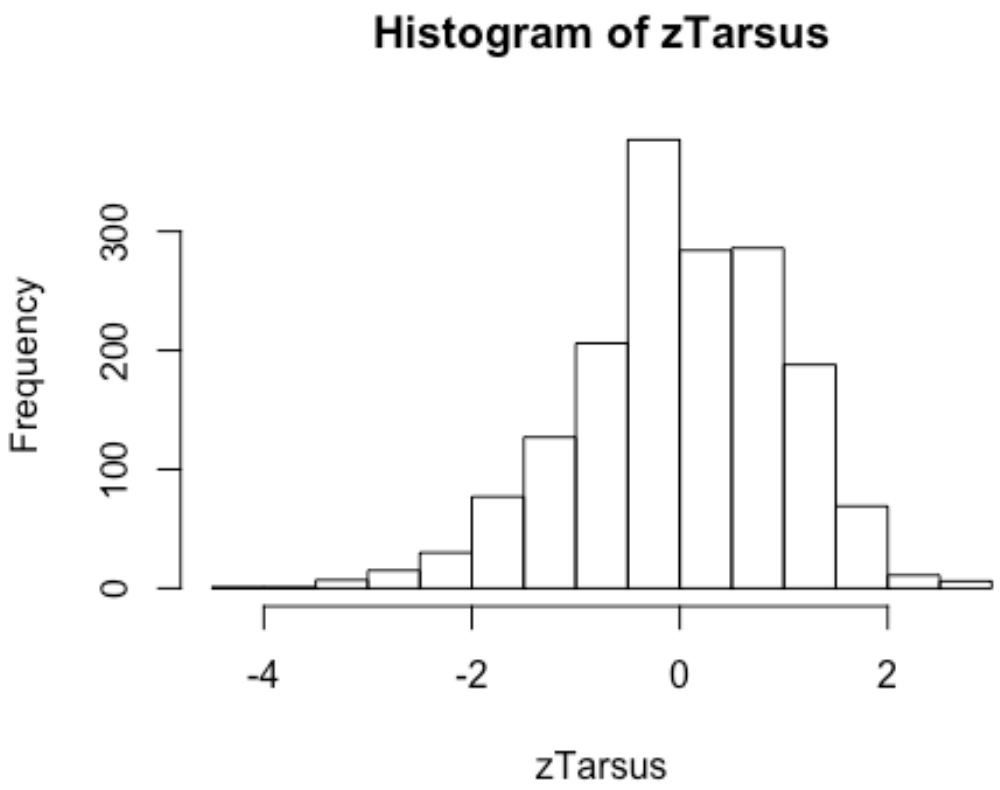
```
zTarsus <- (d2$Tarsus - mean(d2$Tarsus))/sd(d2$Tarsus)
var(zTarsus)
```

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

```
sd(zTarsus)
```

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

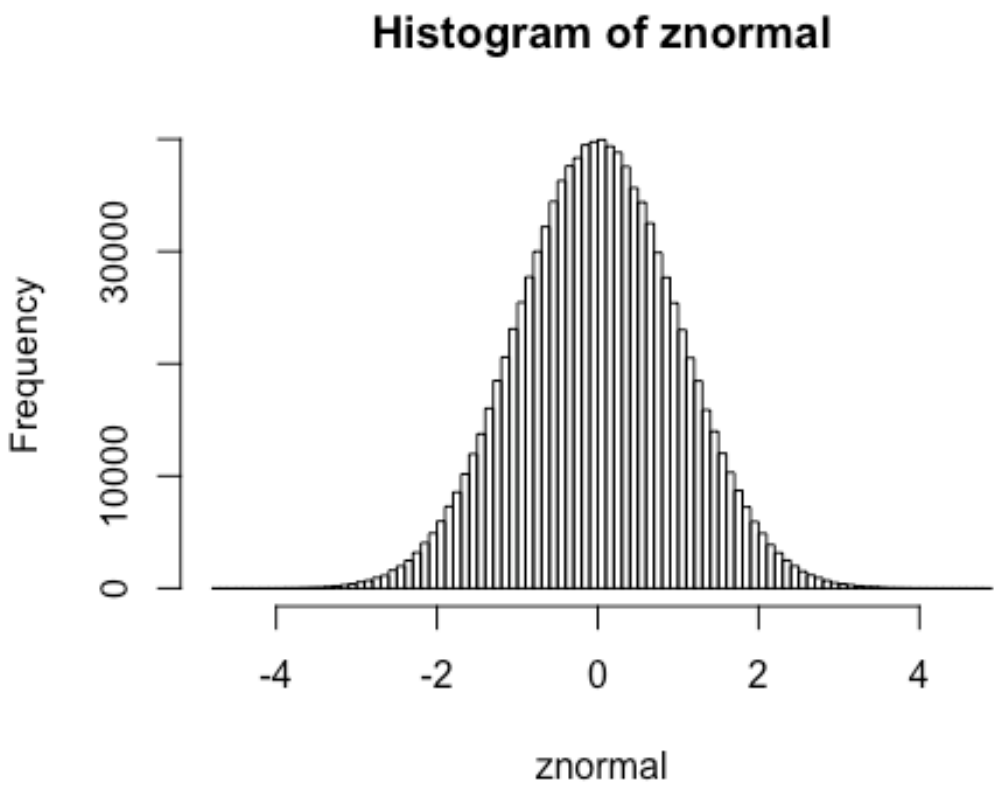
```
hist(zTarsus)
```



But, of course, there is a function for this in R. Use Google to find it, remember it, and use it often!

BTW, the reason we call it z-scores is because the normal distribution is also called z-distribution. The neat thing about R is that it allows you to make us datasets from scratch that follow this distribution:

```
set.seed(123)
znormal <- rnorm(1e+06)
hist(znormal, breaks = 100)
```



set.seed is used so that the random values are not always the same - we can prevent this by using a different "seed" every time. That's a beautiful normal distribution. Let's examine it a bit closer:

```
summary(znormal)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max. 
## -4.799000 -0.674400 -0.000260 -0.000521  0.673300  4.851000
```

Very helpful. We even get the central tendencies in one go. The quantiles refer to the respective cut-offs of the data distribution, think back to the beginning of this lecture! The median is the second quantile where 50% of data points are included. R can also give us the proper normal distribution (not data randomly sampled from it, which we did above):

```
qnorm(c(0.025, 0.975))
pnorm(.Last.value)
```

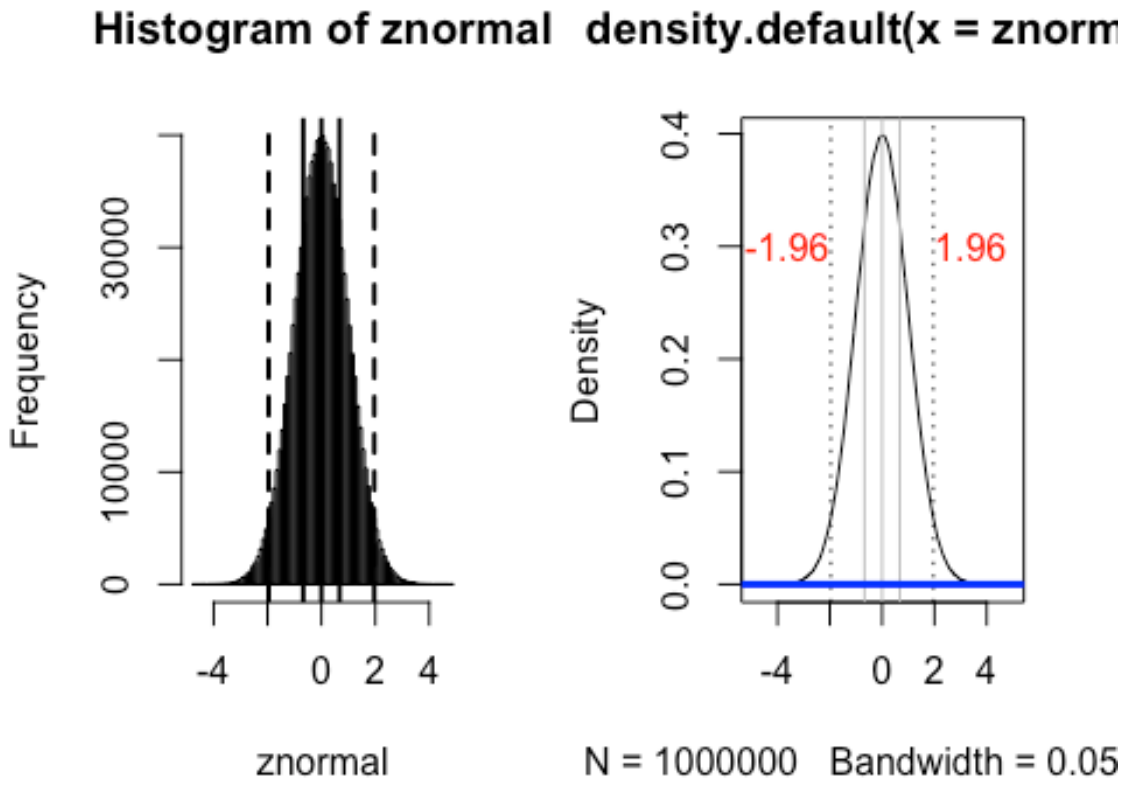
```
## [1] -1.959964  1.959964
```

```
## [1] 0.025 0.975
```

qnorm(c(0.025,0.975)) gives us the 2.5% and 97.5% quantiles from the corresponding probability distribution. Both bracket 95% of all values in the distribution. And pnorm gets us the corresponding probabilities.

These last quantiles are important when we get to hypothesis testing, so remember them!

```
par(mfrow = c(1, 2))
hist(znormal, breaks = 100)
abline(v = qnorm(c(0.25, 0.5, 0.75)), lwd = 2)
abline(v = qnorm(c(0.025, 0.975)), lwd = 2, lty = "dashed")
plot(density(znormal))
abline(v = qnorm(c(0.25, 0.5, 0.75)), col = "gray")
abline(v = qnorm(c(0.025, 0.975)), lty = "dotted", col = "black")
abline(h = 0, lwd = 3, col = "blue")
text(2, 0.3, "1.96", col = "red", adj = 0)
text(-2, 0.3, "-1.96", col = "red", adj = 1)
```

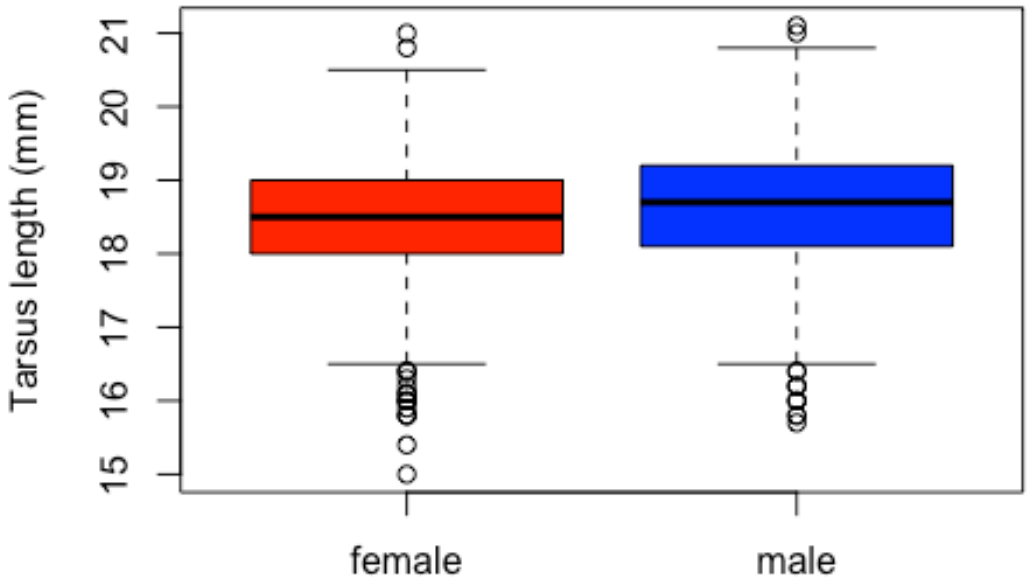



```
## null device
##           1
```

The 95% Confidence Interval (95% CI) is a very important property. It is the range of values the encompass the population true value with 95% probability. That means we will make an error in 5% of the times. Later more.

Now, let's a have a look at a sparrow example. I'll plot the tarsus length between the two sexes. Why did I use d$Sex.1 here and not d$Sex?

```
boxplot(d$Tarsus~d$Sex.1, col = c("red", "blue"), ylab="Tarsus length (mm)")
```



What do you think these boxes and lines represent? You can find that our using

```
?boxplot
```