

Week 1

Daijiang Li

Contents

0.1	What is R?	1
0.2	Why do we use R as ecologists?	1
0.3	Installing R	1
0.4	How will R be used in this course?	1
0.5	A quick introduction to R	2

0.1 What is R?

Computational programming language that is quite useful to analyse, simulate, understand, and visualize data.

0.2 Why do we use R as ecologists?

Ecology is not just fieldwork. Ecologists attempt to understand species interactions, but in order to generalize across different places, we need theoretical development, which often requires computational programming languages like R. Further, in order to analyze field data to gain insight, computing is necessary. I once stated during a panel discussion that I thought every ecologist should know how to program in at least one language, and I certainly stand by it.

Many people become interested in ecology because they love nature. This is cool. What I would like to communicate though is that you can also love computers or mathematics and get into ecology.

0.3 Installing R

If you want to, here are the resources: <https://www.r-project.org>

0.4 How will R be used in this course?

You will not need to learn R for this course. I will use code that I have written during the second lecture every week to provide demonstrations of the material we learned in the previous lecture. The goal is to visually show how ecological processes can occur, and how the outcome can be influenced by small changes to the system. If you do not like this approach, we can limit the use of code in this course, but I figured it would be a nice interactive way to examine ecological phenomena.

0.5 A quick introduction to R

```
a <- 3
b <- 2

a + b
```

```
## [1] 5
```

```
a - b
```

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

```
a * b
```

```
## [1] 6
```

```
a / b
```

```
## [1] 1.5
```

0.5.1 Looping

```
vec <- c(0)
for(i in 1:100){
  vec[i+1] <- vec[i]+i
}
```

```
#there are many different ways to do the same thing
cumsum(1:100)
```

```
## [1] 1 3 6 10 15 21 28 36 45 55 66 78 91 105 120
## [16] 136 153 171 190 210 231 253 276 300 325 351 378 406 435 465
## [31] 496 528 561 595 630 666 703 741 780 820 861 903 946 990 1035
## [46] 1081 1128 1176 1225 1275 1326 1378 1431 1485 1540 1596 1653 1711 1770 1830
## [61] 1891 1953 2016 2080 2145 2211 2278 2346 2415 2485 2556 2628 2701 2775 2850
## [76] 2926 3003 3081 3160 3240 3321 3403 3486 3570 3655 3741 3828 3916 4005 4095
## [91] 4186 4278 4371 4465 4560 4656 4753 4851 4950 5050
```

0.5.2 Conditional statements

```
name <- 'tad'

if(name == 'tad'){
  print('Hello, tad')
}else{
  print('Who are you?')
}
```

```
## [1] "Hello, tad"
```

0.5.3 The importance of the boolean as output (TRUE/FALSE)

```
name=='tad'
```

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

```
1 == 2
```

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

```
1 != 2
```

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

```
1 < 2
```

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

```
1 > 2
```

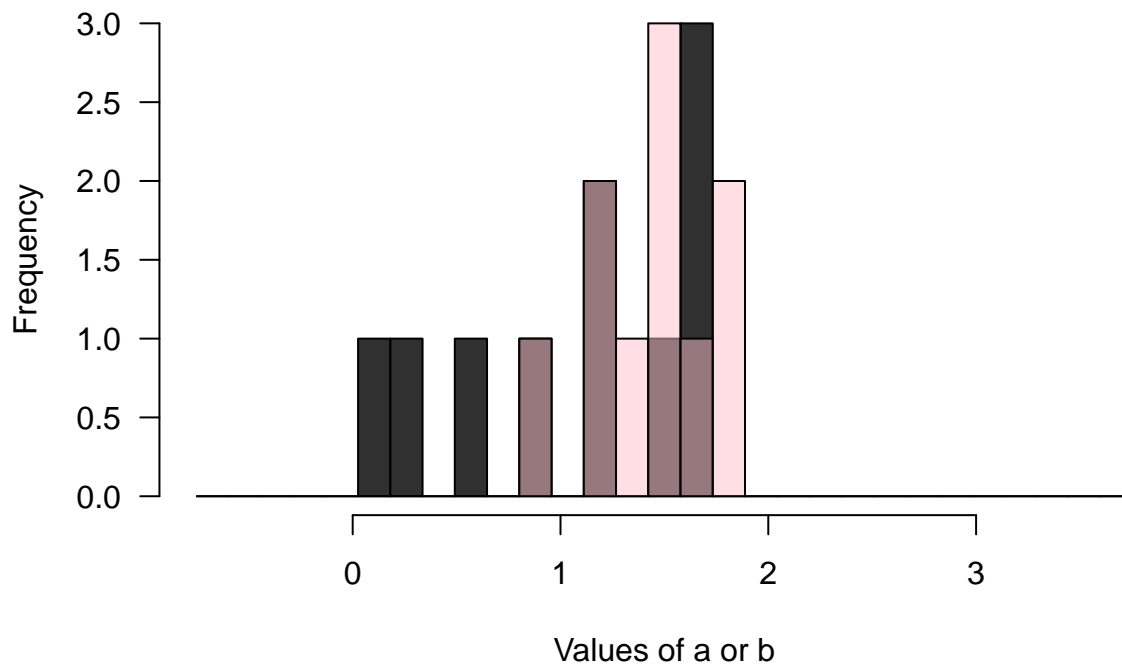
```
## [1] FALSE
```

We can also think about how coding allows to get around some of the limitations of sample size in experimentation

Sample size of $n = 10$

```
a <- rnorm(10, 1, 0.5)
b <- rnorm(10, 1.5, 0.5)

h <- hist(a, las=1, col=grey(0.1,0.9),
  breaks=seq(-0.75, 3.75, length.out=30),
  main='', xlab='Values of a or b')
hist(b, add=TRUE, breaks=h$breaks,
  col=adjustcolor('pink', 0.5))
```



```
t.test(a,b)
```

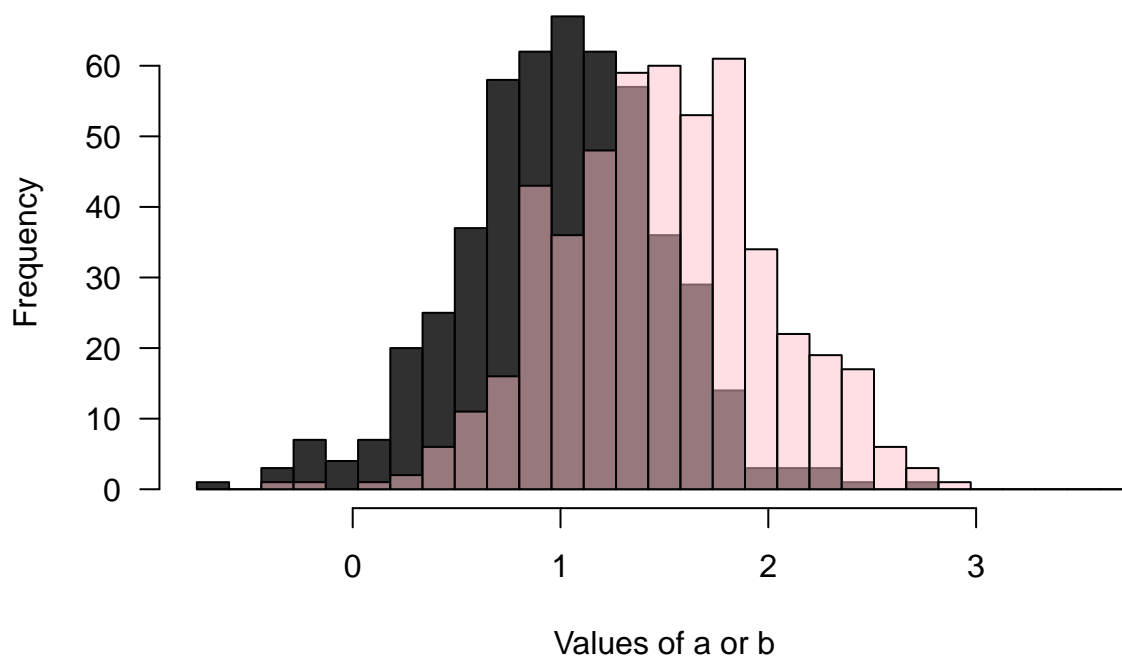
```
##
##  Welch Two Sample t-test
##
## data:  a and b
## t = -1.8191, df = 13.888, p-value = 0.09052
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## -0.79839808  0.06590476
## sample estimates:
## mean of x mean of y
##  1.068576  1.434823
```

Sample size of n = 500

```
a2 <- rnorm(500, 1, 0.5)
b2 <- rnorm(500, 1.5, 0.5)

h <- hist(a2, las=1, col=grey(0.1,0.9),
  breaks=seq(-0.75, 3.75, length.out=30),
  main='', xlab='Values of a or b')

hist(b2, add=TRUE, breaks=h$breaks,
  col=adjustcolor('pink', 0.5))
```



```
t.test(a2,b2)
```

```
##
##  Welch Two Sample t-test
##
## data:  a2 and b2
## t = -15.57, df = 994.56, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.5508261 -0.4275186
## sample estimates:
## mean of x mean of y
##  1.001454  1.490627
```