

Exercises for statistical inference and stuff

Timothée Bonnet

December 6, 2018

Contents

1	Statistical inference and random numbers	1
1.1	Iris	1
	Exercise 1	2
	Exercise 2	2
2	R-studio tricks	2
2.1	Column selection	2
2.2	Short-cuts	2
3	Linear models	2
	Exercise 3	2
4	While-loop	3
4.1	What you need to know	3
4.2	Practice	3
5	If-else statement	4
5.1	What you need to know	4
5.2	Practice	5

1 Statistical inference and random numbers

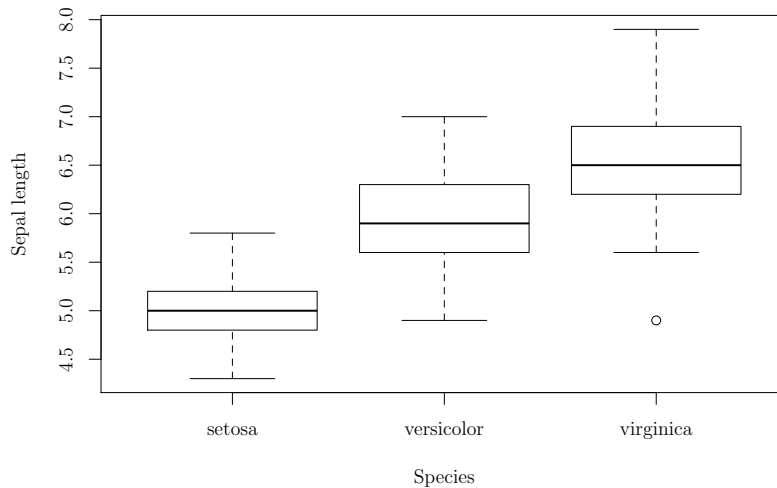
1.1 Iris

Some datasets are shipped with R (in R-base or in packages) and you can load them with the function `data`:

```
data("iris")
```

The dataset looks like that:

```
boxplot(Sepal.Length ~ Species,  
  data = iris,  
  drop = TRUE, ylab="Sepal length", xlab="Species")
```



* Exercise 1

If you like ggplot, redo a boxplot of the iris data using that package.

** Exercise 2

Do species differ in their Sepal length? Use a t-test, an anova, and a linear model to answer.

2 R-studio tricks

2.1 Column selection

2.2 Short-cuts

3 Linear models

** Exercise 3

1. Load Cdata.csv, fit models of y predicted by x1 and x2, or x2 and x3. Something is weird, what is going on? What to do?

2. For model that can be fitted with `t.test`, `aov`, and `lm`, is one of the function faster?
3. Write your own code to obtain a prediction from a `lm` (that is, a simpler version of the `predict` function), with confidence interval. (extra toughness: do it using the matrix formulation of the analytical solution to a linear model)

4 While-loop

4.1 What you need to know

```
while(condition TRUE)
{
  something
}
```

For instance:

```
x <- 0
while(x<10)
{
  x <- x+1
  print(x)
}

## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

4.2 Practice

The function `sample()` takes 5 number between 1 and 6 (like 5 dice!):

```
x <- sample(x = 1:6, size = 5, replace = TRUE)
```

Are all die equal?

```
all(x == x[1])
```

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

Are they ever going to be equal?

Write a while loop to find a case with all die equal

How many attempts does it take

Write a for while loop within a for loop to estimate how long it take on average.

5 If-else statement

5.1 What you need to know

```
if(condition)
{
  do something
}
```

```
if(condition)
{
  do something
}else{
  do something else
}
```

For instance:

```
for (i in 1:10)
{
  if(i < 6)
  {
    print("tofu")
  }else{
    print("bacon")
  }
}
```

```
## [1] "tofu"
## [1] "tofu"
## [1] "tofu"
## [1] "tofu"
## [1] "tofu"
## [1] "bacon"
## [1] "bacon"
## [1] "bacon"
## [1] "bacon"
## [1] "bacon"
```

5.2 Practice

We can draw 100 random number following a random distribution of mean 0 and variance one with:

```
x <- rnorm(n = 100, mean = 0, sd = 1)
```

If we take their logarithm we obtain many “NaN” (Not A Number), because the log of a negative number is undefined:

```
log(x)
## Warning in log(x): NaNs produced
## [1] NaN 0.262355303 NaN NaN 0.480623011
## [6] NaN NaN NaN NaN NaN
## [11] -1.289816855 -1.532779062 NaN NaN NaN
## [16] NaN 0.305145556 NaN 0.039441479 -0.778691386
## [21] NaN NaN -1.939128268 0.655815243 -0.886269071
## [26] 0.179153516 NaN NaN -0.005867485 NaN
## [31] -0.152579789 -2.603725485 -0.384121531 -2.291570403 NaN
## [36] NaN -1.305283273 NaN NaN -1.060265882
## [41] NaN NaN -1.367737843 NaN -4.354962419
## [46] -0.609828572 NaN NaN NaN -0.689051693
## [51] NaN NaN NaN NaN 0.319704580
## [56] -0.727459865 -1.602378549 0.082982817 NaN -1.040568591
## [61] NaN -1.303174012 NaN NaN NaN
## [66] 0.048945943 NaN -0.248965507 NaN NaN
## [71] 0.435781667 -0.239488784 NaN NaN -3.399333253
## [76] 0.187409365 NaN -0.318429368 -0.788822126 0.765125687
## [81] -0.902969119 -0.502387608 NaN -0.915182357 -1.611256568
## [86] NaN -0.890043539 NaN 0.102983162 -2.318588918
## [91] -0.752306641 NaN NaN NaN NaN
## [96] NaN NaN 0.312132730 NaN NaN
```

Let's say we want 0 instead of NaN.

Use a for loop and an if-else statement to do that.

More difficult: Use a for loop and a while loop to re-draw random numbers until they are all positive.