Exercises for statistical inference and stuff

Timothée Bonnet

December 6, 2018

Contents

1 Statistical inference and random numbers

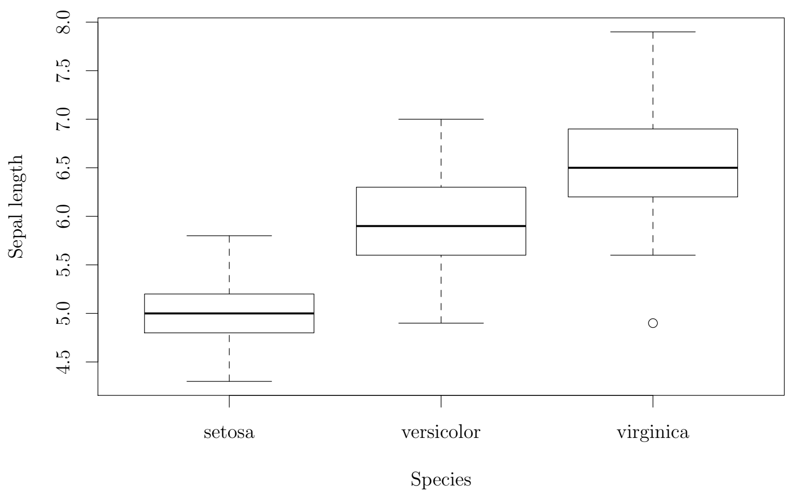
1.1 Iris

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 the species *setosa* and *versicolor* differ in their Sepal length? Use a t-test, an anova, and a linear model to answer. Compare the p-values between the three approaches.

** Exercise 3

Now fit all species (*setosa*, *versicolor* and *virginica*) in a lm and an anova (you cannot fit a t-test with three levels) comparing Sepal length. Compare the model outputs, in particular the p-values. What is different, why?

1.2 P-values and loops

If we draw two sets of random numbers from the same normal distribution, we do not expect them to be associated. In the case below, the p-value for the slope of y on x is 0.802, non-significant.

```
set.seed(1234)
x <- rnorm(100)
y <- rnorm(100)
summary(lm(y~x))

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.88626 -0.61401  0.00236  0.58645  2.98774
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03715    0.10498   0.354    0.724
## x           -0.02608    0.10378  -0.251    0.802
##
## Residual standard error: 1.037 on 98 degrees of freedom
## Multiple R-squared:  0.0006443,Adjusted R-squared:  -0.009553
## F-statistic: 0.06318 on 1 and 98 DF,  p-value: 0.8021
```

** Exercise 4

Are we every going to find a significant p-value with two sets of random numbers? Write a while loop to find out. How many iterations until you find a p-value below 0.05?

*** Exercise 5

How often do you observe a significant test with randomly drawn numbers? Use a for-loop to record the distribution of p-values. Does this distribution depend on the sample size of x and y? What does increasing sample size do to the significant tests?

2 R-studio tricks

2.1 Column selection

* Exercise 6

Use the shortcut `Alt+click` to change the code below so that you plot the five linear models defined at the beginning:

```
lmadd  <- lm(y ~ x1 + x2)
lmnull <- lm(y ~ 1)
lmff   <- lm(y ~ x1*x2)
lmx2   <- lm(y ~ x2)
lmx1   <- lm(y ~ x1)


plot(lm1)
plot(lm2)
...
```

** Exercise 7

What if my code was not well aligned? Use `Ctrl + Alt + clicks` to create multiple cursors, then `Shift + Home` and `Ctrl + C`

```
lmadd <- lm(y ~ x1 + x2)
lmnull <- lm(y ~ 1)
lmff <- lm(y ~ x1*x2)
lmx2<- lm(y ~ x2)
lmx1  <- lm(y ~ x1)


plot(lm1)
plot(lm2)
```

2.2 Short-cuts

R-Studio short-cuts are listed in Tools - Keyboard Shortcuts Help, also accessible using the shortcut `Alt + Shift + K`.

* Exercise 8

Read them, find one that would be helpful to you, and memorize it

3 Linear models

3.1 Diagnostics and assumptions

** Exercise 9

1. Load Cdata.csv
2. fit a linear model of y as a function of x2 and x3. Something is weird, what is going on? How to interpret and what to do?
3. fit a linear model of y as a function of x1 and x2. Something else is weird, what is going on? How to interpret and what to do?

** Exercise 10

Load the dataset Anscombe.csv. It contains four sets of distributions for a x and a y variable. Create a subset of the data for each distribution, and fit a linear regression of y on x for each subset. **Compare the summaries. Use the function plot() to diagnose the models, and to visualize the data. Which models do you trust? For what?**
Try and confirm your confidence in various models by ploting model predictions with confidence intervals and actual observations together.

3.2 Prediction

** Exercise 11

What explains variation in parasitic load? You collected ecto-parasites on some furry large mammals at three locations. Parasites break easily when we collect them and are impossible to count, so we decide to measure parasitic load as their mass. **Why do some mammals have larger parasitic load?**

- Load the `Para.csv` data (don't forget: str(), summary(), plot()...)
- Model `Parasite_Mass` using `lm()`
- Find what variables predict `Parasite_Mass`
- How good are your models? Assumptions? Prediction?
- What biological interpretation can you imagine?

*** Exercise 12

Write your own code to obtain a prediction from a lm (that is, a simpler version of the predict function). Use any dataset to test it.

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?

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

## [1] FALSE
```

Are they ever going to be equal?

** Exercise 13

Write a while loop to find a case with all die equal How many attempts does it take

*** Exercise 14

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

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

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

For instance:

```r
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         NaN         NaN -0.13945976         NaN
##   [6]         NaN -0.28076466         NaN  0.28280278         NaN
##  [11]         NaN         NaN         NaN -0.24153763  0.39083014
##  [16] -0.89205035  0.41686065 -1.47938530  0.17152584 -5.40572299
##  [21] -1.97340644 -1.09956183         NaN -0.27621570  0.56079323
##  [26] -0.75692177 -1.30795634         NaN  0.22497455 -3.15852737
##  [31]         NaN         NaN -1.70380275         NaN  0.53018548
##  [36]  0.69133404         NaN         NaN -1.88363140         NaN
##  [41]         NaN -0.63025491         NaN -0.33254907         NaN
##  [46]         NaN         NaN -1.41561080 -0.16800018         NaN
##  [51]  0.01880018 -0.04872261 -0.19752483  0.48644978 -1.45312758
##  [56] -0.24180106         NaN         NaN         NaN  0.36259497
##  [61] -0.14340530         NaN         NaN -1.14680744 -0.17891184
##  [66]  0.93315813 -0.91500711 -3.33021869 -1.26326119         NaN
##  [71]         NaN  0.17550910  0.31174778         NaN         NaN
##  [76]  0.64699793         NaN -0.04532025  0.84760991  0.09682937
##  [81]  0.08799275 -1.25392199         NaN -0.15577952 -0.84878816
##  [86]         NaN         NaN -0.10479329         NaN         NaN
##  [91]         NaN  0.66450438 -1.22026925  0.15290773 -1.41192914
##  [96] -0.79087742 -0.66279093         NaN         NaN -1.88159017
```

Let's say we want 0 instead of NaN.

** Exercise 15

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

*** Exercise 16

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