Bootstrapping and Randomisation tests

MI Stefan

Semester 2, 2018/19

Introduction

Learning Objectives

Rattus Binomialis

Problem

A rat is performing a "2-alternative forced choice" (2AFC) task in which it must identify an odor presented at a central port. If it detects odor 'A' it should choose the right-hand port for a reward; if it detects odor 'B' is should choose the other port.

Early in the rat's training, you want ot know whether the rat has learned the task yet. So you decide to do a test and keep track of his correct rate for a block of 50 trials. After 50 trials, we see that the rat has gotten 31 trials correct (19 trials wrong) for an average of 62 percent correct. You want to know if the rat has learned the task or if he is still guessing.

All hypothesis tests are the same . . .

Since we are performing a hypothesis test, we need to ask ourselves what the world would look like if the Null Hypothesis was correct. What is the Null Hypothesis?

Simulate one experiment under Ho

Assume the Null Hpothesis is true, then the rat is just guessing, with a 50-50 chance of getting the correct answer in each trial. If we code a correct guess as "1" and a wrong guess as "0", how would you simulate a block of 50 trials? [1]

[1] Recall what you learned in lecture 20 about drawing random numbers in R.

```
block <- sample(c(0, 1), 50, replace = TRUE)
block
```

```
##  [1] 0 0 0 0 0 0 0 0 1 1 0 0 1 0 1 1 1 1 0 1
## [21] 0 0 1 0 0 0 1 1 0 1 0 1 0 0 0 1 0 0 0 0
## [41] 0 0 0 0 0 1 0 0 0 1
```

Out of the 50 trials, how many did the rat guess correctly?

```
correct <- sum(block)
```

Simulate 10000 experiments under Ho

Now is where the simulation-based magic happens. Let's say we have not just 1 rat that behaves as if Ho is true, but ten thousand rats. How would you simulate that?[2]

[2] Recall what you learned about loops in R.

First, we define a vector that will hold the number of correct answers per experiment.

```
number_correct <- vector(mode = "numeric")
```

Next, we loop over a counter that goes from 1 to 10000. In each loop, we do a trial just like the one we did before and store the result in `number_correct`

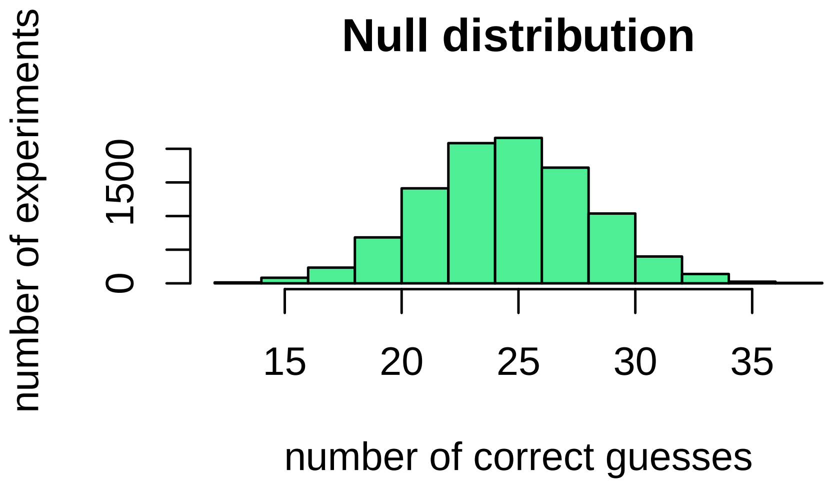
```
nexp <- 10000
for (i in 1:nexp) {
    block <- sample(c(0, 1), 50, replace = TRUE)
    correct <- sum(block)
    number_correct = c(number_correct, correct)
}
```

Visualise simulation results

We have now run the experiment 10000 times assuming H0 is true, and kept track of the number of correct guesses for each experiment. How can we best visualise our results? For instance, by drawing a histogram. Before we do that, answer the following questions for yourself:

- What would you expect the histogram to look like?
- Where would you expect the peak to be?
- Do you remember the name of what this histogram shows you?

```r
hist(number_correct, col = "seagreen2", main = "Null distribution",
    xlab = "number of correct guesses", ylab = "number of experiments")
```



The histogram shows the "Null distribution", i.e. the distribution of outcomes, assuming the Null Hypothesis is correct.

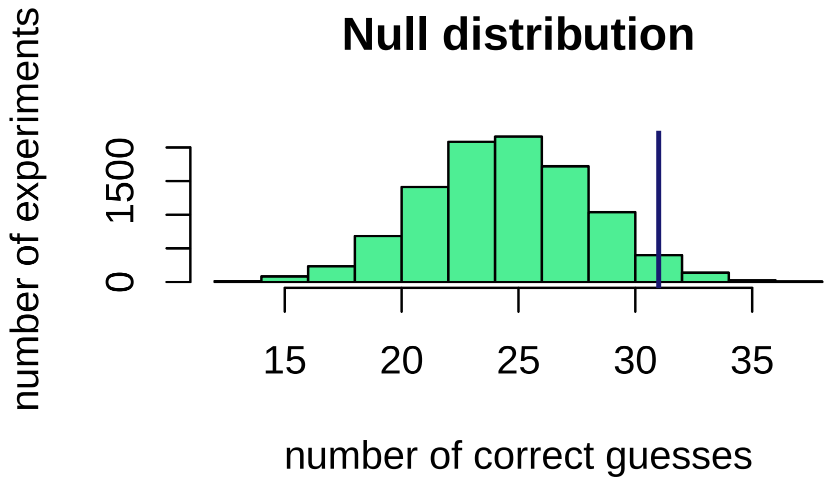
How likely is it to see my experimental results if Ho is true?

As in any hypothesis test, we ask ourselves: "How likely are we to see results as extreme (or even more extreme) as in our experiments if Ho is true?"

First, let's visualise our experimental result (31 out of 50 correct) by adding it as a line to the histogram. This can be done using the abline() function[3], which draws a straight line at a given set of coordinates. What do you see?

[3] Haven't encountered this function before? Remember, you can get more information about any function in R by using help().

```r
hist(number_correct, col = "seagreen2", main = "Null distribution",
    xlab = "number of correct guesses", ylab = "number of experiments")
abline(v = 31, col = "midnightblue", lwd = 2)
```



Most of the experiments in the Null distribution are to the left of the blue line (i.e. less extreme than the results we found). But sometimes, even a rat that is guessing gets 31 or more trials correct. But how often exactly? Well, we can just count!

```
sum(number_correct >= 31)
```

```
## [1] 570
```

But really, what we want is not the absolute number, but the proportion of trials under H0 that gave 31 or more correct. Why? Because *that is our p-value*.

```
p = sum(number_correct >= 31)/nexp
```

Interpreting the p-value and coming to a conclusion

- From this p-value, what do you conclude?
- If two of you both run the simulation, do you get the same p value?[4]
- Is ten thousand trials enough? Should there be more? What would you need to change in the code?

[4] Think back to what you learned in lecture 20.1/2 about the exactness of randomisation-based solution.

Bonus: No-loops solution!

Reading the last section, you may have tried to set `nexp` up to one hundred thousand or even a million. You will have found that the resulting p value is more precise, but also that simulations take longer to run. This is because loops are quite time-consuming. Is there a way of achieving what we did without loops?

Yes, there is! If you think about it, we pick between 0 and 1 a total of 50 times per experiment, and a total of nexp experiment. So, let's just toss that coin $50 \times nexp$ times to begin with. All we have to do then is be clever about how to sum the number of correct guesses per trial.

```
all_experiments <- sample(c(0, 1), 50 * nexp,
    replace = TRUE)
all_experiments <- matrix(all_experiments, ncol = 50)
number_correct <- rowSums(all_experiments)
p = sum(number_correct >= 31)/nexp
p
```

```
## [1] 0.0601
```

Do men and women have different forearm lengths?

Dataset

In the past, you have learned about t-tests using the question of whether men and women have different forearm lengths as an example. In the 1017/18 academic year, a group of ZJE first-year students did the experiment as follows: Each student measured their forearm length (in mm) and entered their (self-identified) gender and their forearm length into a table. The results are in file `forearm_length.csv`

Let's start by importing the dataset into R.

```
forearm_data <- read.csv("forearm_length.csv",
    header = TRUE)
head(forearm_data)
```

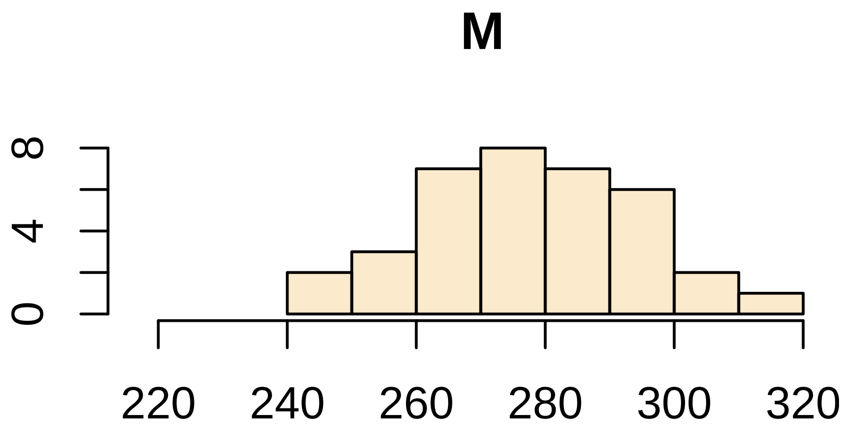
```
##   Gender ForearmLength
## 1      M         279.0
## 2      M         282.0
## 3      F         269.0
## 4      M         276.5
## 5      M         278.5
## 6      M         249.0
```

Let's also take a quick look at the dataset.[5]

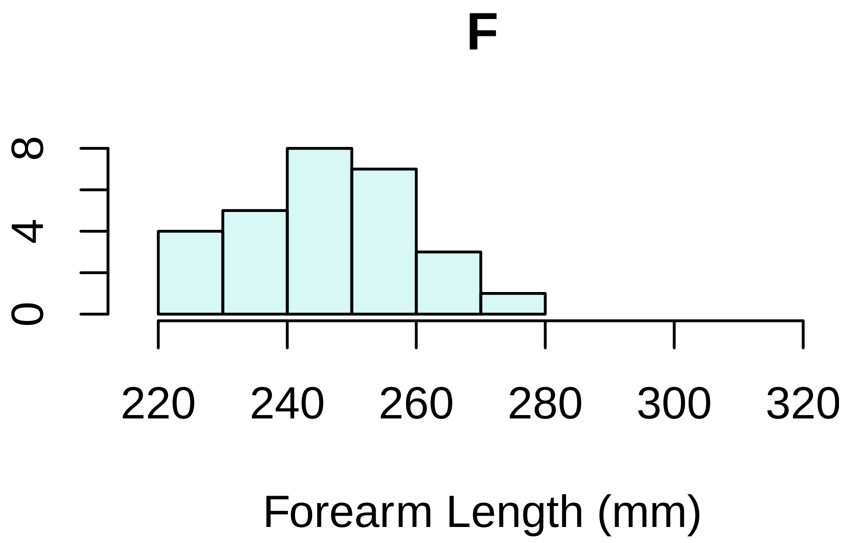
[5] In the following code, why do we define lowx and highx?

```
lowx <- min(forearm_data$ForearmLength) - 10
highx <- max(forearm_data$ForearmLength) + 10

hist(forearm_data[forearm_data$Gender == "M",
    "ForearmLength"], col = adjustcolor("orange2",
    alpha.f = 0.2), xlim = c(lowx, highx), xlab = "",
    ylab = "", main = "M")
```



```
hist(forearm_data[forearm_data$Gender == "F",
    "ForearmLength"], col = adjustcolor("turquoise",
    alpha.f = 0.2), xlim = c(lowx, highx), xlab = "Forearm Length (mm)",
    main = "F", ylab = "")
```



What is the difference between mean male and female forearm length in our dataset?^{As you can see, we take the absolute difference. Why?}

```
mean_M <- mean(forearm_data[forearm_data$Gender ==
    "M", "ForearmLength"])
mean_F <- mean(forearm_data[forearm_data$Gender ==
    "F", "ForearmLength"])
diff <- abs(mean_M - mean_F)
diff
```

```
## [1] 31.27183
```

You can see that the difference is 31 mm.

Simulate one experiment under Ho

Let's now pretend that we don't know how to do a t-test (or that we do not feel comfortable performing a t-test)[6] Is there a bootstrap-based alternative that we can use?

Yes, there is. Again, the key is to ask ourselves: Would we be likely to see a difference as or more extreme than the difference we saw if the Null Hypothesis was true?

In our case, the Null Hypothesis is that there is no difference in forearm length between women and men, or in other words, that all forearm lengths in the sample are drawn from the same underlying population. What do we know about that population? Not very much, except that all our data has been drawn from it. So, the best available representation of our population is our dataset.

A bootstrap sample, then, is a random sample from that combined population (with replacement). One experiment consists of taking one sample we call "Male" and one sample we call "Female". How big should both samples be?

Since we are basically simulating our real-life experiment, it makes sense to have samples the same size as in our original experiment.

[6] Recall the conditions that need to be met for a t-test!

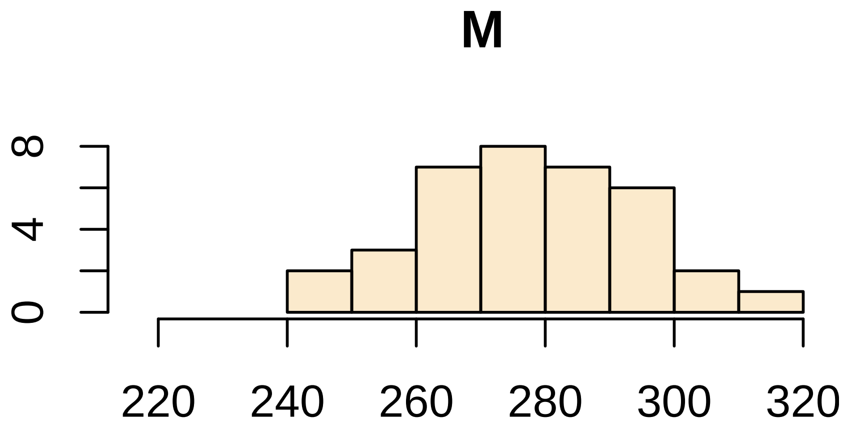
```
n_male <- sum(forearm_data$Gender == "M")
n_female <- sum(forearm_data$Gender == "F")
```

This gives us the size for both bootstrap samples.^{How would you convince yourself that this works and we have counted correctly?}. Now, let's take the actual sample

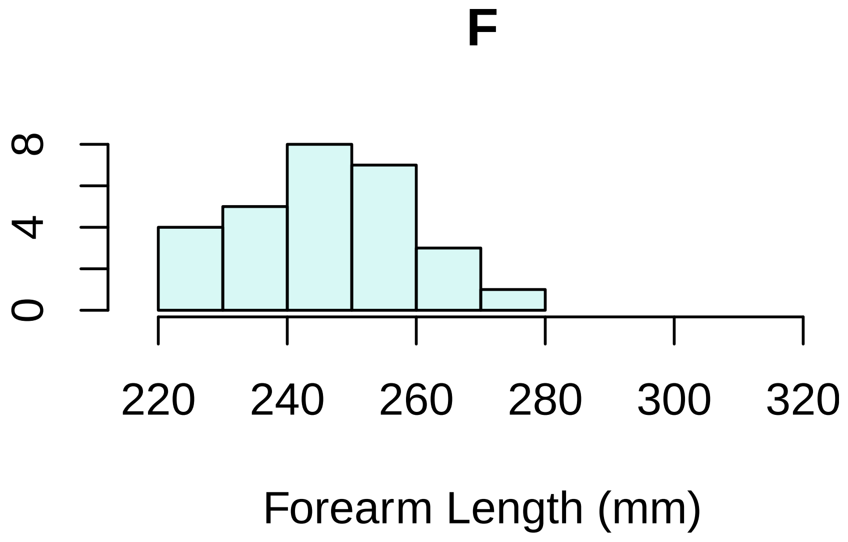
```
male <- sample(forearm_data$ForearmLength, n_male,
    replace = TRUE)
female <- sample(forearm_data$ForearmLength, n_female,
    replace = TRUE)

lowx <- min(forearm_data$ForearmLength) - 10
highx <- max(forearm_data$ForearmLength) + 10

hist(forearm_data[forearm_data$Gender == "M",
    "ForearmLength"], col = adjustcolor("orange2",
    alpha.f = 0.2), xlim = c(lowx, highx), xlab = "",
    ylab = "", main = "M")
```



```
hist(forearm_data[forearm_data$Gender == "F",
    "ForearmLength"], col = adjustcolor("turquoise",
    alpha.f = 0.2), xlim = c(lowx, highx), xlab = "Forearm Length (mm)",
    main = "F", ylab = "")
```



```
sample_mean_M <- mean(male)
sample_mean_F <- mean(female)
sample_diff <- abs(sample_mean_M - sample_mean_F)
```

Simulate 10000 experiments under Ho

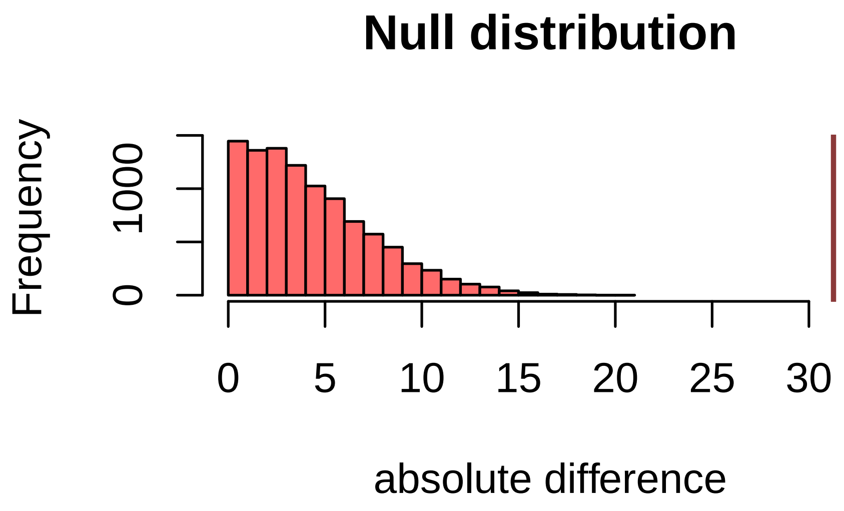
Now again, in order to get a p value, we want to repeat this bootstrap experiment many times, for instance ten thousand. Try and write down the code yourself before turning the page!

```r
experiments <- 10000
diffs <- vector(mode = "numeric")

for (i in 1:experiments) {
    male <- sample(forearm_data$ForearmLength,
        n_male, replace = TRUE)
    female <- sample(forearm_data$ForearmLength,
        n_female, replace = TRUE)
    sample_mean_M <- mean(male)
    sample_mean_F <- mean(female)
    sample_diff <- abs(sample_mean_M - sample_mean_F)
    diffs <- c(diffs, sample_diff)
}
```

Visualising the Null distribution and plotting the p value is very similar to our previous example

```r
xmax <- max(diffs, diff)
hist(diffs, main = "Null distribution", xlab = "absolute difference",
    col = "indianred1", xlim = c(0, xmax + 2))
abline(v = diff, col = "indianred4", lwd = 2)
```



```r
pval <- sum(diffs >= diff)/experiments
pval
```

```
## [1] 0
```

What is your p value? How do you interpret it? Do you see potential problems with it? How would you report it?

Your simulation will tell you that your p value is 0. Zero means an outcome is mathematically impossible. But we know that this is not exactly true. We know that there is at least one possible sample that gives a difference like the one we found in the experiment (namely the sample that we took during the experiment). So, reporting p=0 is problematic. The only thing we can say is that it is less than one in the total number of bootstrap experiments we did, so in this case less than 1 in ten thousand: $p < 0.0001$

Compare to a "traditional" t-test

Last year, you learned how to do a t-test on the same data in R.[7] Let's quickly run the t-test and compare it to the result you had!

[7] Before we do this here, convince yourself that the conditions for doing a t-test are met!

```
t.test(ForearmLength ~ Gender, data = forearm_data)
```

```
##
##  Welch Two Sample t-test
##
## data:  ForearmLength by Gender
## t = -8.4871, df = 61.998, p-value =
## 5.676e-12
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -38.63727 -23.90638
## sample estimates:
## mean in group F mean in group M
##        247.2143        278.4861
```

Acknowledgments