Probability Basics

Intro to Stats, Spring 2017

Prof. Gaston Sanchez

Learning Objectives

- How to use R to simulate chance processes
- Getting to know the function `sample()`
- Simulate a coin
- Simulate a die

Introduction

Drawing samples with `sample()`

R provides the function `sample()` which lets you draw samples, with or without replacement, from an input vector.

For example, say you have a "box" with tickets 1, 2, 3. One way to create such a box in R is with a vector:

```
# box with tickets
tickets = c(1, 2, 3)
```

To draw one ticket, use `sample()` like this:

```
# draw one ticket
sample(tickets, size = 1)
```

```
## [1] 3
```

To draw two tickets WITHOUT replacement, use `sample()` like this:

```
# draw 2 tickets without replacement
sample(tickets, size = 2)
```

```
## [1] 3 1
```

To draw two tickets WITH replacement, use `sample()` and specify its argument `replace = TRUE`, like this:

```
# draw 2 tickets with replacement
sample(tickets, size = 2, replace = TRUE)
```

```
## [1] 3 2
```

The way `sample()` works is by taking a random sample from the input vector. This means that every time you invoke `sample()` you will likely get a different output.

In order to make the examples replicable (so you can get the same output as me), we need to specify what is called a **random seed**. This is done with the function `set.seed()`. By setting a seed, every time you use one of the random generator functions, like `sample()`, you will get the same values.

```
# set random seed
set.seed(1234)

# draw 4 tickets with replacement
sample(tickets, size = 4, replace = TRUE)
```

```
## [1] 1 2 2 2
```

Try the code above. You should get the exact same sample.

Simulating a coin

Now that we've talked about `sample()`, let's use R to implement code that simulates tossing a fair coin one or more times.

To toss a coin using R, we first need an object that plays the role of a coin. So let's start by creating a `coin` object using a character vector with two elements: `"heads"` and `"tails"`:

```
# coin object
coin <- c("heads", "tails")
```

Tossing a coin is a random experiment: you either get heads or tails. To get a random output in R we can use the function `sample()`. Here's how to simulate a coin toss using `sample()` to take a random sample of size 1 from `coin`:

```
# one toss
sample(coin, size = 1)
```

```
## [1] "tails"
```

We can also use `sample()` to take samples of sizes different than one, and also to sample with replacement. To simulate multiple tosses, we can change the value of the `size` argument, and set `replace = TRUE`:

```r
# 3 tosses
sample(coin, size = 3, replace = TRUE)
```

```
## [1] "tails" "heads" "heads"
```

```r
# 6 tosses
sample(coin, size = 6, replace = TRUE)
```

```
## [1] "tails" "tails" "tails" "tails" "heads" "tails"
```

To make our code reusable, it's better to create a **function** that lets us toss a coin multiple times. Here's an example of a `toss()` function based on the use of `sample()`:

```r
toss <- function(coin, times = 1) {
  sample(coin, size = times, replace = TRUE)
}
```

To define your own functions you use the homonym function: `function()`. You can define one or more argument inside `function()`.

Once you've defined a function, you can use it as any other R function:

```r
toss(coin, times = 1)
```

```
## [1] "heads"
```

```r
toss(coin, times = 4)
```

```
## [1] "tails" "heads" "heads" "heads"
```

Frequencies

Typical probability problems that have to do with coin tossing, require to compute the total proportion of `"heads"` and `"tails"`:

```r
# five tosses
five <- toss(coin, times = 5)

# proportion of heads and tails
sum(five == "heads") / 5
```

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

```r
sum(five == "tails") / 5
```

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

It is also customary to compute the relative frequencies of `"heads"` and `"tails"` in a series of tosses:

```r
# relative frequencies of heads
cumsum(five == "heads") / 1:length(five)
```

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

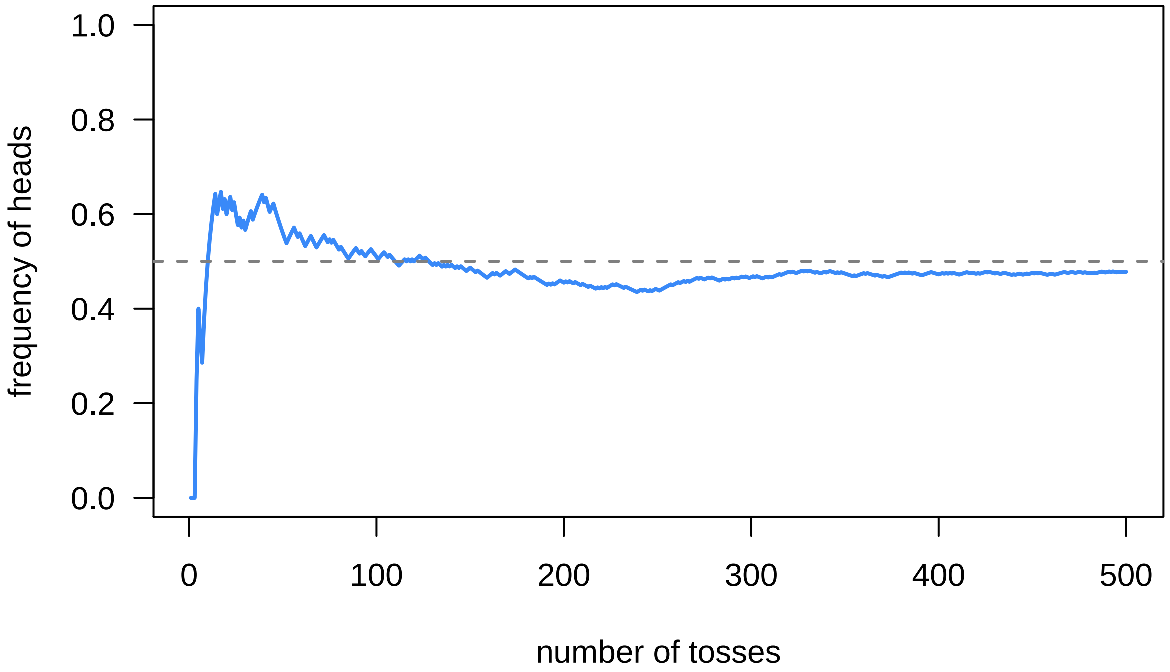
```r
# relative frequencies of tails
cumsum(five == "tails") / 1:length(five)
```

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

Likewise, it is common to look at how the relative frequencies of heads or tails change over a series of tosses:

```r
set.seed(5938)
hundreds <- toss(coin, times = 500)
head_freqs = cumsum(hundreds == "heads") / 1:500

plot(1:500, head_freqs, type = "l", ylim = c(0, 1), las = 1,
     col = "#3989f8", lwd = 2,
     xlab = 'number of tosses',
     ylab = 'frequency of heads')
# reference line at 0.5
abline(h = 0.5, col = 'gray50', lwd = 1.5, lty = 2)
```



So far we have written code in R that simulates tossing a coin one or more times. We have included commands to compute proportion of heads and tails, as well the relative frequencies of heads (or tails) in a series of tosses. In addition, we have produced a plot of the relative frequencies and see how, as the number of tosses increases, the frequency of heads (and tails) approach 0.5.