

Lada Adamic
SI 544
Sept 14, 2006

1 Discrete distributions with R

I was just reading some R documentation (R for Windows FAQ <http://cran.r-project.org/bin/windows/base/rw-FAQ.html>) and found that that you can change the shortcut to specify the working directory (if you would like to have it be different than the default). "You will have a shortcut to R-2.3.1/bin/Rgui.exe on your desktop and/or somewhere on the Start menu file tree. Right-click each shortcut, select Properties... and change the 'Start in' field to your working directory."

I also found out that on the Mac you can click on an icon above the console and have your entire command history displayed in a separate window for easy reference. A tip from Prof. McQuaid is that you can include comments in your command history like so

```
> # now I'm going to add 2 plus 2  
> 2+2
```

```
[1] 4
```

Now I have a narrative of what I was doing in my history file. By the way, you can save your command history by selecting File>Save History. Then you can just load it again later.

2 Coin flips

Coin flipping can be a bit hard on the fingers, so we'll have R do the work for us.

```
> sample(c("H","T"), 10, replace = T)
```

```
[1] "T" "T" "H" "T" "H" "H" "T" "T" "H" "H"
```

So this is what we had R do. We gave it an array consisting of two elements "H" (heads) and "T" (tails). We told it to sample with replacement 10 times. With replacement means that it picks one element at random, puts it back, and draws again.

We can also sample without replacement:

```
> sample(c("blue","yellow","red","green","purple"), 5, replace=F)
```

```
[1] "purple" "yellow" "green" "blue" "red"
```

```
> sample(c("blue","yellow","red","green","purple"), 5, replace=F)
```

```
[1] "green" "blue" "red" "yellow" "purple"
```

```
> sample(c("blue","yellow","red","green","purple"), 5, replace=F)
```

```
[1] "blue" "purple" "yellow" "red" "green"
```

What would happen if we tried to draw more elements without replacement than there are?

```
> sample(c("blue","yellow","red","green","purple"), 8, replace=F)
```

```
Error in sample(length(x), size, replace, prob) :
  cannot take a sample larger than the population
when 'replace = FALSE'
```

Ah, for once a comprehensible error message. We could of course draw 8 elements with replacement.

```
> sample(c("blue", "yellow", "red", "green", "purple"), 8, replace=T)
```

```
[1] "blue" "red" "purple" "yellow" "blue" "yellow" "blue"
"blue"
```

Not all the outcomes need to be equally likely, for example, we can simulate 100,000 births (I know, a lot easier than going through with it for real), and use our probabilities of boy (51.3) and girl (48.7) births. Let's practice with 10 births first.

```
> x = sample(c("boy", "girl"), 10, replace=T, prob=c(0.513, 0.487))
> x
[1] "girl" "boy" "boy" "boy" "girl" "girl" "boy" "girl" "boy" "boy"
```

How many girls were there?

```
> (x=="girl")
[1] TRUE FALSE FALSE FALSE TRUE TRUE FALSE TRUE FALSE FALSE
> sum(x=="girl")
[1] 4
> sum(x=="boy")
[1] 6
```

First we looked at a vector of true/false values that evaluated whether each element of an array was equal to the string "girl". We then summed these vectors (true counts as 1, false as 0).

Now we are ready for 100,000 kids (twice):

```
> x1 = sample(c("boy", "girl"), 1e5, replace=T, prob=c(0.513, 0.487))
> sum(x1=="boy")
[1] 51399
> sum(x1=="girl")
[1] 48601
> sum(x1=="boy")/(sum(x1=="boy")+sum(x1=="girl"))
[1] 0.51399
> x2 = sample(c("boy", "girl"), 1e5, replace=T, prob=c(0.513, 0.487))
> sum(x2=="boy")
[1] 51337
> sum(x2=="girl")
[1] 48663
> sum(x2=="boy")/(sum(x2=="boy")+sum(x2=="girl"))
[1] 0.51337
```

I've used the scientific notation to tell R to take 100,000 samples (1e5). And I've asked it for two samples. In both cases the proportion of boys is pretty close to 51.3%, but not exactly, there is a little bit of variability. How much variability should I expect? Well, that is given by the binomial distribution...

2.1 Binomial Distribution

The binomial distribution tells us the total number of outcomes of a particular kind (boy birth, coin landing heads, other binary outcomes...) given a number of trials and the probability of "success". We can plot the density function (the probability of obtaining any given number of successes) using the `dbinom()` function.

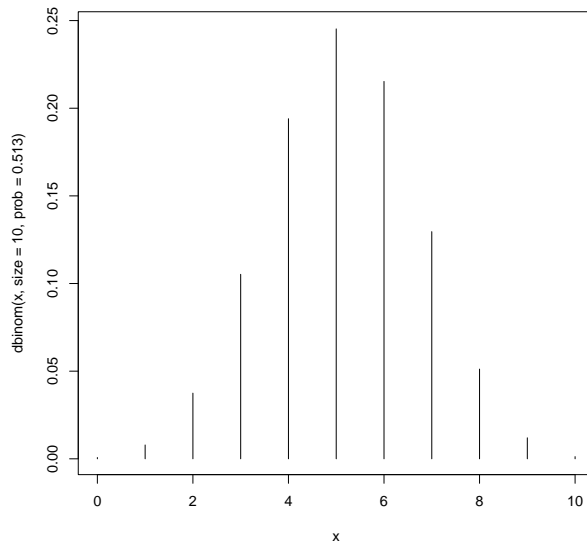


Figure 1: The probability distribution of the number of boy births out of 10.

```
> x = 0:10
> plot(x,dbinom(x,size=10,prob=0.513),type="h")
```

We’ve created a dummy `x` vector that just enumerates all the possibilities (0 .. 10), then we invoked the binomial discrete distribution function with $n = 10$ and $p = 0.513$, and plotted it as a histogram (`type="h"`).

The binomial distribution is given by:

$$P(X = x) = \binom{n}{x} p^x (1 - p)^{(n-x)} \quad (1)$$

2.2 Cumulative distribution functions

Often times we want to know not whether there will be exactly 6 boy births ($P(X = x)$), but whether there will be 3 or fewer boy births ($P(X \leq x)$). By convention the cumulative distribution functions begin with a “p” in R, as in `pbinom()`. Let’s try it out:

```
> pbinom(3,size=10,prob=0.513)
[1] 0.1513779
```

We can compare this with the probability of having exactly 3 boy births

```
> dbinom(3,size=10,prob=0.513)
[1] 0.1052534
```

Sometimes we want to know what the probability is of having more than x events of a certain kind, in which case we need to take $1 - P(X \leq x - 1)$. The probability that there are 8 or more boy births is given by $(1 - \text{Pr}(\text{seven or fewer}))$.

```
> 1-pbinom(7,size=10,prob=0.513)
[1] 0.06443873
```

2.3 Plotting a cumulative distribution

If we generate 100 sequences of 100 coin tosses, we can plot the cumulative distribution of the number of heads that came up.

```
> for (i in 1:100) {  
+ x = sample(c("H","T"), 100, replace = T)  
+ y[i] = sum(x=="H")  
+ }  
> n = length(y)  
> plot(sort(y), (1:n)/n, type="s", ylim=c(0,1))
```

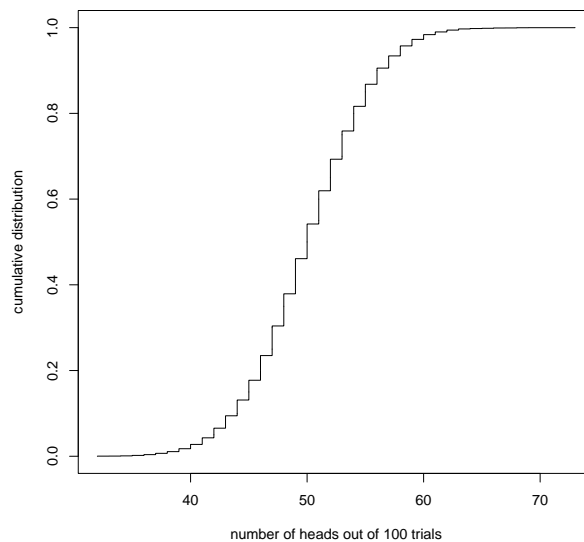


Figure 2: Cumulative distribution of the number of heads in repeated experiments of 100 coin tosses.

We can read off the probability of getting a result less than x directly from the cumulative distribution. This would be harder to do from a regular histogram (Figure 3):

2.4 Simulating switches and runs in coin tossing experiments

One may be interested in finding the longest sequence of either heads or tails, if we toss the coin 100 times. Or rather the distribution of the longest sequence if we run the experiment over and over again. We may also be curious how many times the outcome will switch between heads and tails.

```
experiments = 1000  
tosses = 100  
for (i in 1:experiments) {  
  x = sample(c("H","T"), tosses, replace = T)  
  y[i] = sum(x=="H")  
  longestrun[i] = 1  
  switches[i] = 0  
  currentrun = 1  
  for (j in 2:tosses) {  
    if (x[j] != x[j-1]) {
```

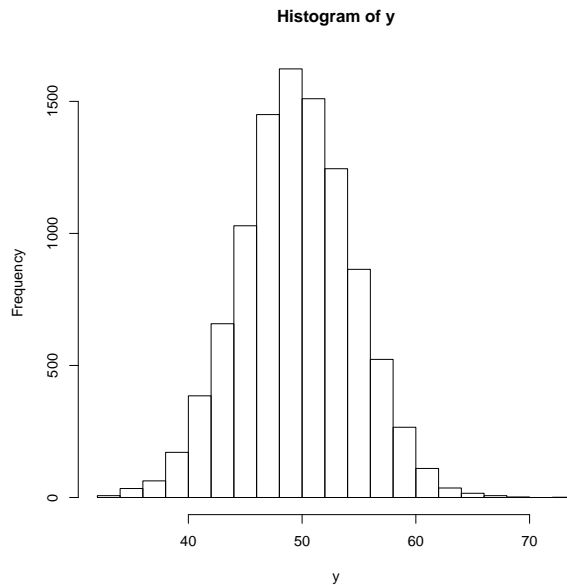


Figure 3: Histogram of the number of heads in repeated experiments of 100 coin tosses.

```

    if (currentrun > longestrun[i]) {
      longestrun[i] = currentrun
    }
    currentrun = 1
    switches[i] = switches[i] + 1
  } else {
    currentrun = currentrun + 1
  }
}
}
plot(jitter(switches),jitter(longestrun),cex=0.5)

```

So we ran the experiment 1000 times and show a jitter plot (otherwise the outcomes would overlap exactly, and we wouldn't get a good sense). From the plot we can deduce the distribution in the number of switches and also the length of the longest run.

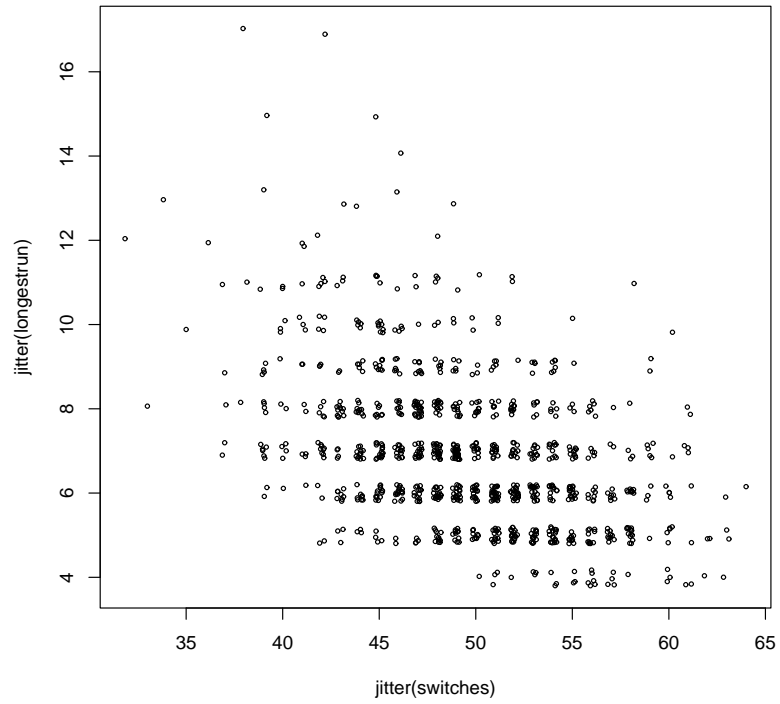


Figure 4: Jittered plot of the number of switches between heads and tails in a coin toss experiment and the length of the longest run of either