Binomial Probability Distribution

Eric Scott

2020-01-30

Bolker text pg: 120-122

Probabilty distributions

So far, we have been dealing with Bernoulli random variables: the probability of getting a single event in a single trial. It can also be used to calculate the probability of getting any particular sequence of results in some number of trials.

A probability **distribution** is the probability assigned to each possible value of the random variable (outcome of an experiment or observation)

For a Bernouli random variable, there are only two possible mutually exclusive outcomes, so distribution of probabilities is only defined by P("success").

We write the distribution as:

$$X \sim Bernouli(p)$$

Where X is the random variable (outcome) and p is probability of success and "~" is read as "distributed as".

Binomial distribution

What if we have more than one trial?

The Binomial distribution is related to Bernoulli. It shows the probability of getting k events out of N unordered trials, if each trial has probability p of an event. by convention

- N = # trials
- k = # events (AKA "successes")
- p = probability an event occurs

Example: All possible outcomes from 2 trials (with independent events), and probability p: e.g., say you do **two** walks and see one butterfly [N = 2 trials (a walk), k = 1 event (a butterfly)].

trial 1	trial 2
1	1
1	0
0	1
0	0

So there's only **ONE** possibility where you see two butterflies, only **ONE** possibility that you see no butterflies, but **TWO** that get one butterflies, one not.

-

Thus:

- $Pr\{k = 2|N = 2, p\} = p^2$ (walk 1 AND walk 2)
- $Pr\{k = 0|N = 2, p\} = (1 - p)^2$ (not walk 1 AND not walk 2)
- $Pr\{k = 1|N = 2, p\} = p(1 - p) + (1 - p)p = 2p(1 - p)$ (walk 1 AND not walk 2 OR walk 2 AND not walk 1)

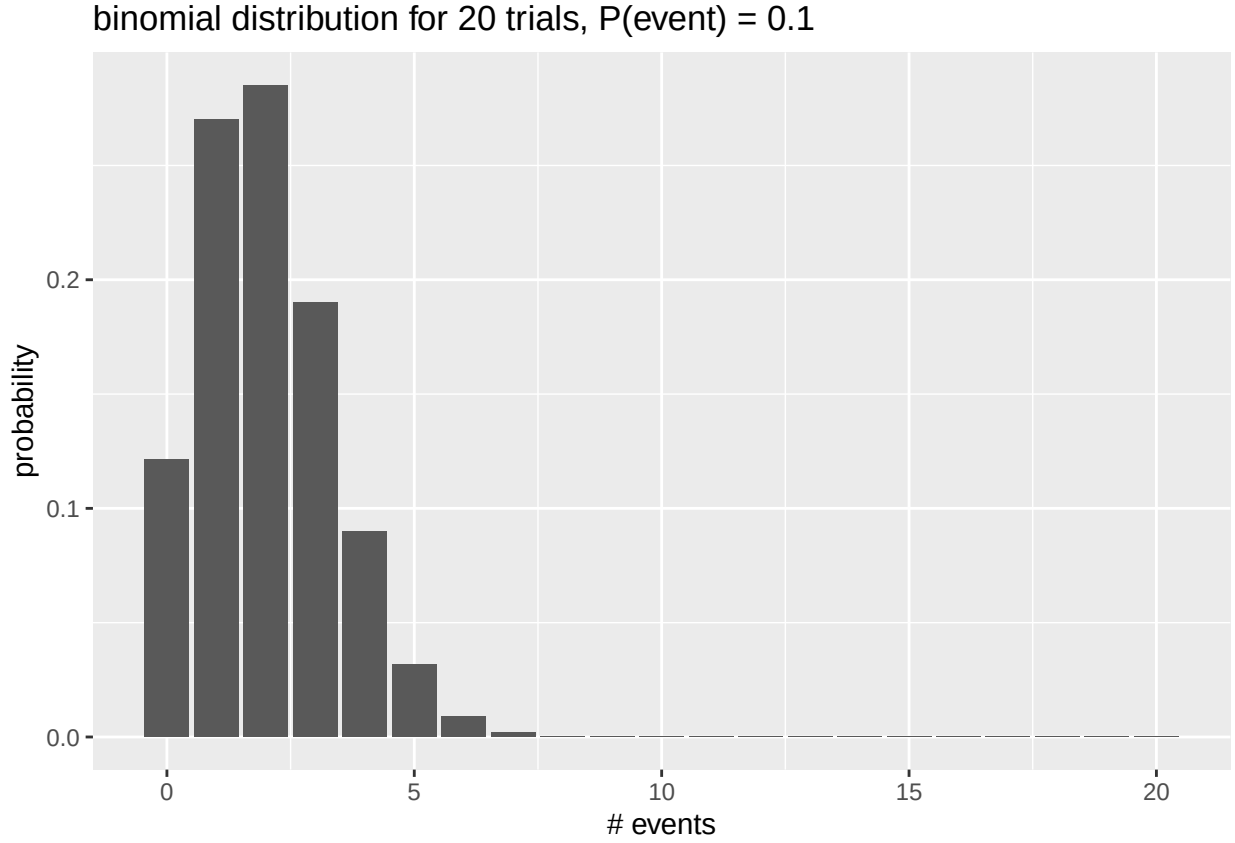
(Axiom #1 – mutually exclusive events)

It's a distribution!

This is a "distribution" because it shows the probabilities of different outcomes, given the data

- x-axis = # of events
- y-axis = probability

Show binomial distributions with N = 4, 20 and p = 0.1, 0.4



Binomial distribution as an equation:

$$Pr\{x = k|N, p\} = \left(p^k(1 - p)^{N-k}\right)\binom{n}{x}$$

Walking through this equation:

- p^k = probability of recapturing k particular events (Axiom # 2 – independent events)
- $(1\text{-}p)^{N\text{-}k}$ = the probability of not recapturing N-k particular particular (Axiom #2)
- $\binom{n}{k}$ = the number of ways you can get k events in N trials

Show how this simplifies in our example with two walks.

N choose k

to understand the meaning of "the number of ways you can get k events in N trials", write out all possibilities.

$\binom{n}{k}$ is mathematical notation for the number of ways you can get k events in N trials:

$$\binom{n}{k} = \frac{N!}{k!(N-k)!} = \frac{N \times (N-1) \times (N-2) \times ... \times [N-(N-1)]}{\{k \times (k-1) \times ... \times [k-(k-1)]\}\{(N-k) \times (N-k-1) \times ... \times [N-k-(N-k-1)]}$$

e.g.,

$$\binom{2}{1} = \frac{2!}{2!(2-1)!} = \frac{2 \times 1}{1 \times 1} = 2$$

and

$$\binom{2}{2} = \frac{2!}{2!(0!)} = \frac{2 \times 1}{2 \times 1 \times 1} = 1$$

(NOTE: By definition, 0! = 1)

Using Binomial distribution to calculate support for a model:

- Recall that L(model|data) is proportional to P(data|model).
- use the Binomial distribution to calculate the likelihood of values of p, given N = 2, k = 1

Possible parameters: Pr{k=1 |N=2, p} L{p | k=1, N=2}:

$$Pr(k=1|N=2,p) = (p^k(1-p)^{N-k})\binom{N}{k}$$

$$= (p^1(1-p)^{2-1})\binom{2}{1} = 2p(1-p)$$

ASIDE: The equation for the Binomial distribution shows its similarity to the Bernoulli distribution. The *probability* of getting k successes differs between ordered (Bernoulli) and unordered (Binomial) data sets, but the *likelihood* is effectively the same because it differs by a constant, specifically N-choose-k. This falls out in the wash with the mysterious "arbitrary constants".

SO: you can compare different models to a data set using Bernoulli probabilities for each event

OR you can compare different models to a data set using pooled binomial probabilities

BUT you can't compare one model fit using Bernoulli probabilities to another fit using binomial probabilities

. . . in this last case the data are different because one data set is ordered (we know which walks included butterflies) and the other data set is unordered (we know we saw a butterfly on, say 2/10 walks, but we don't know which ones they were. . .

Binomial distribution in R

In R, you can get the probability of data given a binomial model using the function `dbinom()`

`dbinom(x, size, prob, log)`

where

- `x` = # events, k
- `size` = # trails, N
- `prob` = probability of an event in a single trial, p
- `log = TRUE` or `FALSE` . . . do you want the log of the probability, or the probability?

Worked example: Anne's calypso orchids. . .

Load a few convenience packages

```
library(here) #reproducible file paths
```

```
## here() starts at /Users/scottericr/Documents/Tufts/ecological-stats
```

```
library(readr) #'safer' version of read.csv()
```

Read in data

```
orchids <- read_csv(here("data", "orchid.seedlings.csv"))
```

```
## Parsed with column specification:
## cols(
##   site = col_double(),
##   year = col_double(),
##   seedlings.0 = col_double(),
##   seeds.m1 = col_double(),
##   seeds.m2 = col_double(),
##   seeds.m3 = col_double(),
##   seeds.m4 = col_double()
## )
```

```
head(orchids)
```

```
## # A tibble: 6 x 7
##    site  year seedlings.0 seeds.m1 seeds.m2 seeds.m3 seeds.m4
##   <dbl> <dbl>       <dbl>    <dbl>    <dbl>    <dbl>    <dbl>
## 1     1  2005          14      500      550      750      640
## 2     1  2006          41      590      500      550      750
## 3     1  2007          16      660      590      500      550
## 4     1  2008          25      600      660      590      500
## 5     1  2009          70      670      600      660      590
## 6     1  2010          25      560      670      600      660
```

We want to know the number of seedlings in year 0 (`seedlings.0`) as a function of seeds in year -1, -2, -3, or -4.

Seedlings in year 0 is our `x` (k), or number of events. Our size, N, is how many seeds we think we started out with, depending on the lag (of 1, 2, 3, or 4 years). We can use the binomial distribution here to calculate the probability of the data, given an arbitrary model. Here our "model" has two parts:

1. our hypothesis about the lag, i.e., how many years
2. our binomial probability of seeds per seedling

Take the case where the lag is 4, and prob = 0.05

```
dbinom(x, size, prob, log=F)
```

```
lnprob_m1 = dbinom(x = orchids$seedlings.0, size = orchids$seeds.m4, prob = 0.05, log = TRUE)
lnprob_m1 # show the list of probabilities for each observation, given the model
```

```
##  [1]  -8.923719  -2.916742  -5.255764  -2.506072 -24.468373  -3.622435
##  [7]  -2.587884  -3.565091 -16.165716  -3.402405  -2.369908 -23.133150
## [13]  -6.249062  -4.051376  -9.303817 -11.797458 -11.913662  -7.290864
```

```
sum(lnprob_m1) # log-likelihood of the whole data set
```

```
## [1] -149.5235
```

Compare to a lag of 4 and seeds per seedling of 0.04

```
lnprob_m2 = dbinom(x = orchids$seedlings.0, size = orchids$seeds.m4, prob = 0.04, log = TRUE)
lnprob_m2 # show the list of probabilities of each observation, given the model
```

```
##  [1]  -5.492695  -4.641476  -3.234387  -3.110793 -34.643346  -2.551748
##  [7]  -2.773753  -2.502362 -24.415435  -2.492767  -2.829497 -32.816704
## [13]  -4.622032  -2.891575  -7.129033  -9.389059  -8.925363  -5.269170
```

```
sum(lnprob_m2) # log-likelihood of the whole data set
```

```
## [1] -159.7312
```

Compare to a lag of 3 and seeds per seedling of 0.05

```
lnprob_m3 = dbinom(x = orchids$seedlings.0, size = orchids$seeds.m3, prob = 0.05, log = TRUE)
lnprob_m3 # show the list of probabilities of each observation, given the model
```

```
##  [1] -12.324381  -5.789786  -4.238321  -2.890223 -19.741308  -2.974263
##  [7]  -2.744871  -4.681920 -18.470007  -2.875618  -4.257611 -36.226396
## [13]  -6.249062  -5.027170  -9.303817 -16.926787  -8.800772  -1.620357
```

```r
sum(lnprob_m3) # log-likelihood of the whole data set
```

```
## [1] -165.1427
```

We could do this for the entire parameter space of binomial probabilities and lags, but this is inefficient. Next lab you will learn to write for loops that can do this kind of thing with less code.