# R Tutorial-04

T. Scofield

You may click here to access the .qmd file.

In this issue

- working with binomial distributions in R
- working with geometric distributions in R
- simulating Bernoulli trials; `rbinom(), rgeom()`

## Working with binomial distributions in R

A random variable $X \sim \text{Binom}(n, p)$

- represents the number of successes in `n` independent Bernoulli trials where the probability of success on any single trial is `p`
- has sample space $S = \{0, 1, 2, \ldots, n\}$
- has pmf given by $\text{P}(X = x) = \dfrac{n!}{x!(n-x)!} p^x (1-p)^{n-x} = \dbinom{n}{x} p^x (1-p)^{n-x}$

### Specific instance: $X \sim$ Binom$(5, 0.35)$

We can evaluate the pdf in R using several ways. For instance, if $X \sim \text{Binom}(5, 0.35)$, then $\text{P}(X = 3)$ is found by

```
choose(5,3) * (0.35)^3 * (1 - 0.35)^2
```

```
[1] 0.1811469
```

This is such an important calculation, however, that there is the function `dbinom()` to streamline it:

```r
dbinom(3, 5, 0.35)        # computes P(X=3) when X~Binom(5,0.35)
```

```
[1] 0.1811469
```

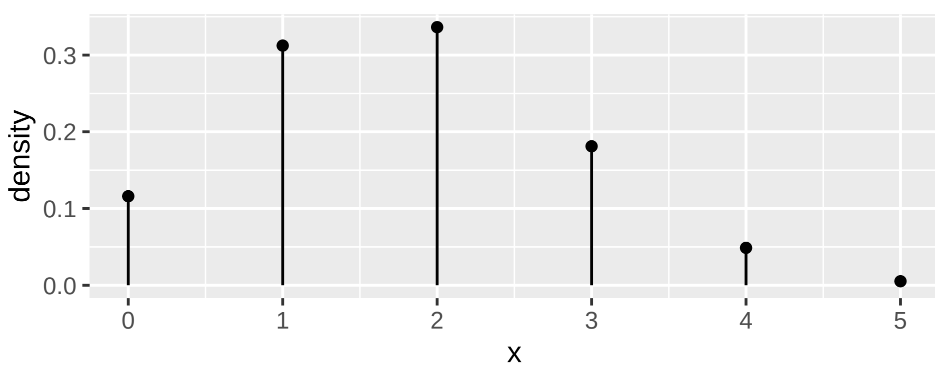
We can get obtain $\text{P}(X = x)$ for the entire sample space $S$ of values $x$:

```r
dbinom(0:5, 5, 0.35)
```

```
[1] 0.116029063 0.312385937 0.336415625 0.181146875 0.048770313 0.005252187
```

The result of the previous command, along with the fact that $\text{P}(X = x) = 0$ whenever $x$ is *not* a value in the sample space, tells us everything there is to know about the pmf. The **ggformula** package (loaded each time you load **mosaic**) gives us the `gf_dist()` command (along with other plotting commands beginning with `gf_`) which we may use to display this pmf:

```r
gf_dist("binom", size=5, prob=0.35)
```




The pmf is **unimodal** (i.e., has one "peak", or highest point) and **right-skewed**.

Since we know

$$
\begin{aligned}
\mathrm{P}(X=0) &= \binom{5}{0}(0.35)^0(0.65)^5 = 0.11603 \\
\mathrm{P}(X=1) &= \binom{5}{1}(0.35)^1(0.65)^4 = 0.31239 \\
\mathrm{P}(X=2) &= \binom{5}{2}(0.35)^2(0.65)^3 = 0.33642 \\
\mathrm{P}(X=3) &= \binom{5}{3}(0.35)^3(0.65)^2 = 0.18115 \\
\mathrm{P}(X=4) &= \binom{5}{4}(0.35)^4(0.65)^1 = 0.04877 \\
\mathrm{P}(X=5) &= \binom{5}{5}(0.35)^5(0.65)^0 = 0.00525,
\end{aligned}
$$

we deduce the following cumulative probabilities (i.e., values of the cdf):

$$
\begin{aligned}
P(X \le 1) &= 0.11603 + 0.31239 = 0.42842 \\
P(X \le 2) &= 0.11603 + 0.31239 + 0.33642 = 0.76483 \\
P(X \le 20) &= 0.11603 + 0.31239 + 0.33642 + 0.18115 + 0.04877 + 0.00525 = 1.0.
\end{aligned}
$$

The calculation of $\mathrm{P}(X \le 2)$ can be implemented in R this way,

```r
sum(dbinom(0:2, 5, 0.35))
```

```
[1] 0.7648306
```

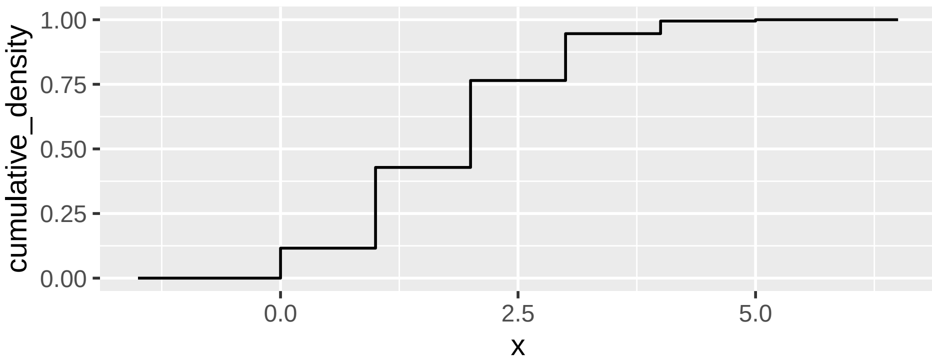
but there is a built-in function, `pbinom()`, to streamline the calculation of binomial cdfs.

```r
pbinom(c(1,2,20), 5, 0.35)
```

```
[1] 0.4284150 0.7648306 1.0000000
```

Study the graph of the cumulative distribution function, to understand its shape why it must be a function with values increasing from 0 to 1:

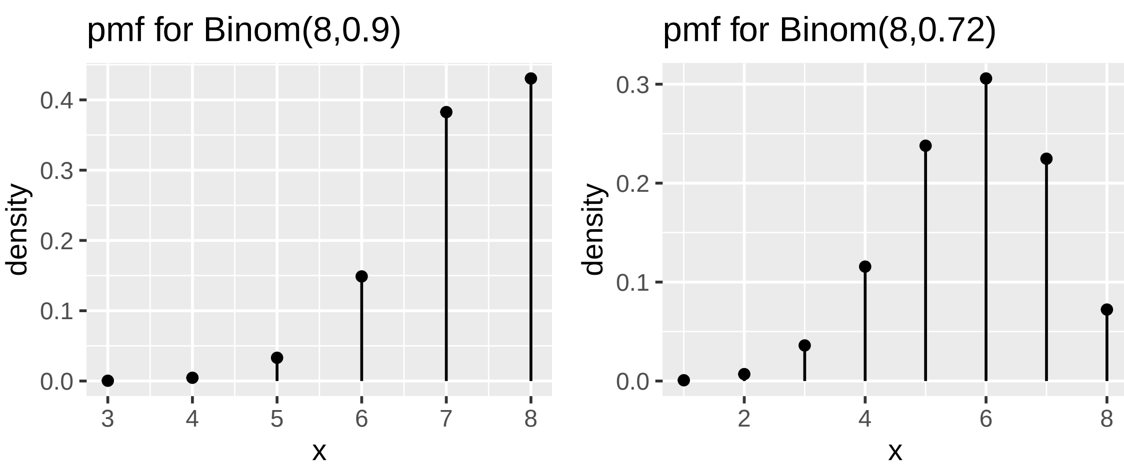
```r
gf_dist("binom", size=5, prob=0.35, kind="cdf")
```




### Other binomial distributions

The shape of the pmf for $\text{Binom}(5, 0.35)$ was right-skewed. In fact, the same might be said of any binomial distribution $\text{Binom}(n, p)$ when $p < 0.5$. On the other hand, if $p > 0.5$, a binomial distribution is generally left-skewed, as these graphs with $n = 8$ and $p = 0.9$, $p = 0.72$ show.

```
p1 = gf_dist("binom", size=8, prob=0.9) |> gf_labs(title="pmf for Binom(8,0.9)")
p2 = gf_dist("binom", size=8, prob=0.72) |> gf_labs(title="pmf for Binom(8,0.72)")
gridExtra::grid.arrange(p1, p2, ncol=2)
```




However, many binomial distributions are only slightly skewed. In fact, if

$$np \geq 10 \qquad \text{and} \qquad n(1-p) \geq 10,$$

you could just as well say that $\text{Binom}(n, p)$ has a (near) *symmetric* distribution.

## Working with geometric distributions in R

Like binomial distributions, $X \sim \text{Geom}(p)$ strings together independent Bernoulli trials, not a set number of them, but continuing until the first success. The sample space for $X$ is $S = 0, 1, 2, 3, \ldots$, another discrete variable, since $X$ is the count of leading "failed" trials. The pmf for $X$ is

$$\text{P}(X = x) = (1 - p)^x p,$$

implemented by the R function `dgeom()`.

### Special instance: $X \sim$ Geom$(0.35)$

When our Bernoulli trials produce successes with probability $p = 0.35$, the chances that the first success comes on the 3rd trial (i.e., $\text{P}(X = 2)$) is

```
(1-0.35)^2 * 0.35
```

```
[1] 0.147875
```

```
dgeom(2, 0.35)
```

```
[1] 0.147875
```

There is a `pgeom()` command for evaluating the cdf.

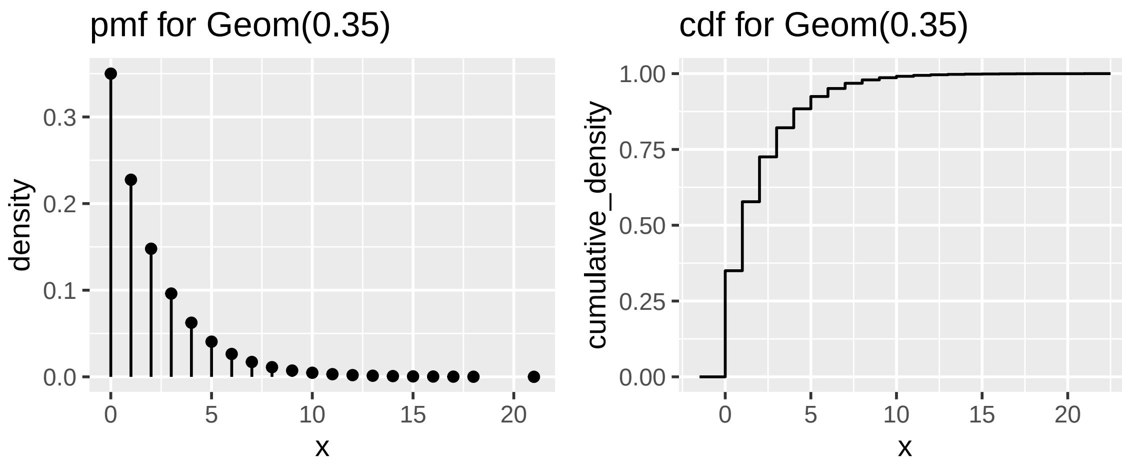
```
pgeom(3, 0.35)      # P(X <= 3), i.e. 3 or fewer failures before the first success
```

```
[1] 0.8214938
```

```
1 - pgeom(1, 0.35)      # P(X > 1), i.e. at least 2 failures before the first success
```

```
[1] 0.4225
```

```
p1 = gf_dist("geom", prob=0.35) |> gf_labs(title="pmf for Geom(0.35)")
p2 = gf_dist("geom", prob=0.35, kind="cdf") |> gf_labs(title="cdf for Geom(0.35)")
gridExtra::grid.arrange(p1, p2, ncol=2)
```




## Simulating Bernoulli trials (and related)

A single Bernoulli trial can be simulated easily. Here I do so while taking $p = 0.7$:

```
bag = c(0,1)
sample(bag, prob=c(0.3, 0.7), size=1)
```

```
[1] 0
```

When the output is a 1, that represents a success; 0 represents failure. Naturally, the success rate over many trials should be approximately 0.7:

```
N = 10000
manyRuns = replicate(N, sample(bag, prob=c(0.3, 0.7), size=1))
prop(~(manyRuns == 1))
```

```
prop_TRUE
   0.7024
```

### Simulating binomial outcomes

A random variable $X$ Binom$(10, 0.7)$ counts successes in not just one Bernoulli trial, but 10 of them. All one needs is a slight modification, employing `resample()` instead of `sample()` and summing the resulting vector, to simulate a single result for $X$:

```
bag = c(0,1)
sum(resample(bag, prob=c(0.3, 0.7), size=10))
```

```
[1] 7
```

It's as if a 70% free-throw shooter took 10 shots while you weren't looking, then reported to you the number he made.

Using `replicate()`, it is easy to produce many values of this binomial r.v.:

```
N=10000
manyRunsBinom = replicate(N, sum(resample(bag, prob=c(0.3, 0.7), size=10)))
```

What the vector `manyRunsBinom` contains is akin to the same 70% free-throw shooter taking 10 shots and counting successes, and keeping a running log of this for 10000 days, straight.

But the makers of the software have provided the command `rbinom()` to streamline the content of the previous code chunk. The exact same process is achieved by

```
manyRunsBinom2 = rbinom(10000, 10, 0.7)
tally(~manyRunsBinom2, format="proportion")
```

```
manyRunsBinom2
     1      2      3      4      5      6      7      8      9     10
0.0001 0.0008 0.0103 0.0379 0.1015 0.2032 0.2675 0.2291 0.1196 0.0300
```

The relative frequencies produced by `tally()` are near matches for the actual binomial probabilities—compare:

```
dbinom(0:10, 10, 0.7)
```

```
 [1] 0.0000059049 0.0001377810 0.0014467005 0.0090016920 0.0367569090
 [6] 0.1029193452 0.2001209490 0.2668279320 0.2334744405 0.1210608210
[11] 0.0282475249
```

#### Simulating geometric outcomes

There is an `rgeom()` command, too, for simulating results of a geometric r.v.

```r
rgeom(1, 0.3)      # simulates one instance of X~Geom(0.3)
```

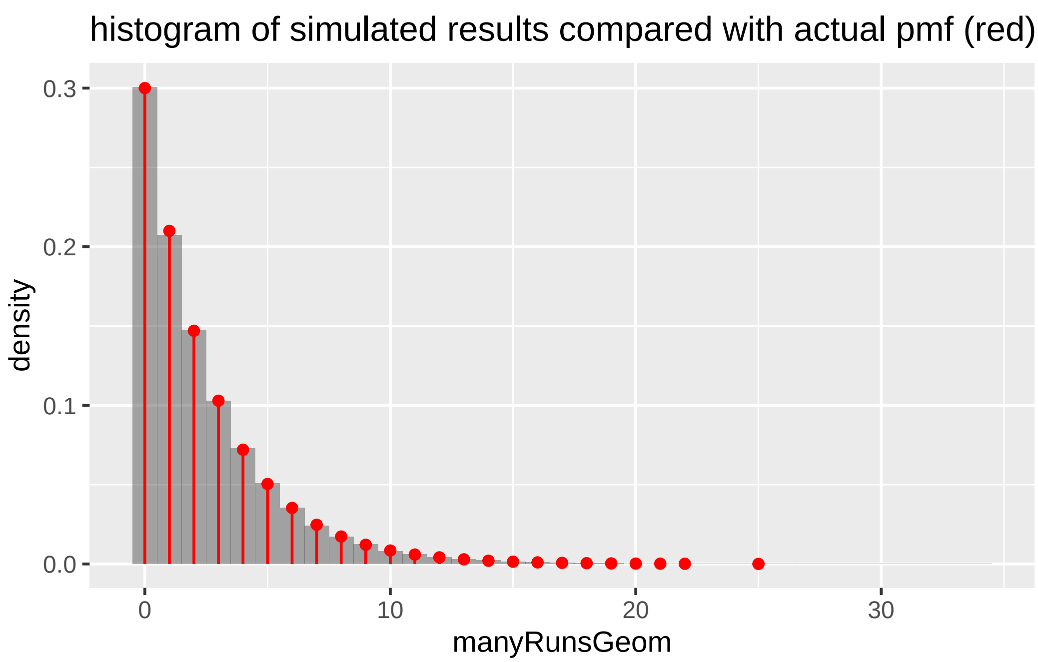
```
[1] 3
```

It's as if a friend—a very poor free-throw shooter at 30%—said "I'm going to shoot free-throws until I make one. I wonder how many I'll miss before one goes in?" On this occasion, 4 misses were endured and the 5th shot was made.

```r
manyRunsGeom = rgeom(50000, 0.3)    # simulates 50000 instances
head(manyRunsGeom)
```

```
[1]  2 12  8  6  0  0
```

```r
gf_dhistogram(~manyRunsGeom, binwidth=1) |> gf_dist("geom", prob=0.3, color="red") |>
  gf_labs(title="histogram of simulated results compared with actual pmf (red)")
```




Here, we see the results if that same friend, never improving from his 30% shooting, repeats 50000 times the experiment of counting the number of misses until he makes a free-throw. The histogram of his results closely mirrors the pmf for Geom(0.3).