Foundations of Probability in R

The binomial distribution

- Note that:

```
pbinom(q, size, prob) = 1 - pbinom(q, size,  prob, lower.tail=FALSE)
```

- Estimating probablity with `rbinom`: Note that `rbinom` produces a list of results, and when we write `new_list <- list == n`, with n being a constant, we get another list of boolean values indicating whether each element in the first list equals n. Taking the mean of `new_list` gives us the probability density, because each element in new list is either `TRUE` or `FALSE`, which R interprets as 1 or 0.

```
# This gives the mean, which is approximately 5
mean(rbinom(1000, 10, .5))
```

```
## [1] 4.899
```

```
# While this gives the probability density
mean(rbinom(1000, 10, .5) == 5)
```

```
## [1] 0.253
```

- Variance:

$$\sigma^2 = Var(X) = size \cdot \pi \cdot (1 - \pi)$$

- For the binomial distribution, the mathematical symbol denoting probability is π.
- When an individual trial has only 2 possible outcomes, it is called a _Bernoulli random variable___.

Laws of probability

1. Multiplying a random variable:

Given $X \sim Binomial(10, 5)$:

```
x <- rbinom(100000, 10, .5)
mean(x)
```

```
## [1] 4.99829
```

```
var(x)
```

```
## [1] 2.503812
```

```
y <- 3 * x
mean(y)
```

```
## [1] 14.99487
```

```
var(y)
```

```
## [1] 22.53431
```

- When multiplying a random variable by k (a constant), all of the values of that random variable are also multiplied by k.
- When multiplying a random variable by k (a constant), we also multiply the expected value by k.
- When multiplying a random variable by k, we multiply the variable by k^2.

So:

$$E[k \cdot X] = k \cdot E[X]$$
$$var(k \cdot X) = k^2 \cdot var(X)$$

2. Adding 2 random variables

$$E[X + Y] = E[X] + E[Y]$$

(Even if X and Y are not independent)

$$Var[X + Y] = Var[X] + Var[Y]$$

(Only if X and Y are independent)

Conditional Probability & Bayes' Theorem

First, we need to review a few probability rules:

- Completeness Rule: $P(B \cap A) + P(B \cap A^c) = P(B)$, in which $P(A^c)$ is the probability that A does NOT happen.
- Conditional Probability: $P(A \cap B) = P(A \mid B)\, P(B) = P(B \mid A)\, P(A)$, or also denoted as: $P(A|B) = \frac{P(A \cap B)}{P(B)}$ and $P(B|A) = \frac{P(B \cap A)}{P(B)}$

Bayes' Theorem:

$$\begin{aligned} P(A|B) &= \frac{P(A \textit{ and } B \textit{ both happen})}{P(B \textit{ happens})} \\ &= \frac{P(B \cap A)}{P(B)} \\ &= \frac{P(B \cap A)}{P(B \cap A) + P(B \cap notA)} \\ &= \frac{P(B|A) \cdot P(A)}{P(B|A) \cdot P(A) + P(B|notA) \cdot P(notA)} \end{aligned}$$

In which:

- $P(A)$ and $P(B)$ is the probabilities of observing A and B. They are also called the **prior probabilities**.
- $P(B|A)$ is the conditional probability of B given A, and can also be interpreted as the *likelihood* of A given a fixed B.
- $P(A|B)$ is the conditional probability of A given B, or the **posterior probablity** of A given B.

Bayes' Theorem can be used to estimate the probability of an event A, given that we have a prior probability (prior for short) $P(A)$, which is the first estimate we have for such event. This can either be our personal belief, or the data we have through past observation. Maybe we want to update this probability (so that it becomes more useful), and intuitively, we would want more evidence.

By also observing event B (besides A), we get (B). We then include these priors into our calculation in order to produce a new probability for A but this time given that B happens, which is $P(A|B)$.

We can also express this process by asking: if we are given $P(B|A)$, how can we "reverse" it and get $P(A|B)$?

For example:

- We have a coin, which can either be: fair (which gives 50% chance of head), or biased (which gives 75% chance of head).
- We estimate that the coin has a 10% probability of being biased. This is our prior.
- By tossing the coin many times, and observe the amount of time it lands on head, we can update the prior to a posterior probability. In this example, we toss the coin 20 times, in which the coin lands on head 14 times out of 20.

```
# P(14 heads given fair)
prob_14_fair <- dbinom(14, 20, .5) * 0.9

# P(14 heads given biased, or not fair)
prob_14_biased <- dbinom(14, 20, .75) * 0.1

# P(14 heads)
prob_14 <- prob_14_fair + prob_14_biased
```

- The posterior probability $P(biased\ given\ 14\ heads)$ is:

```
prob_14_fair/prob_14
```

```
## [1] 0.6636486
```

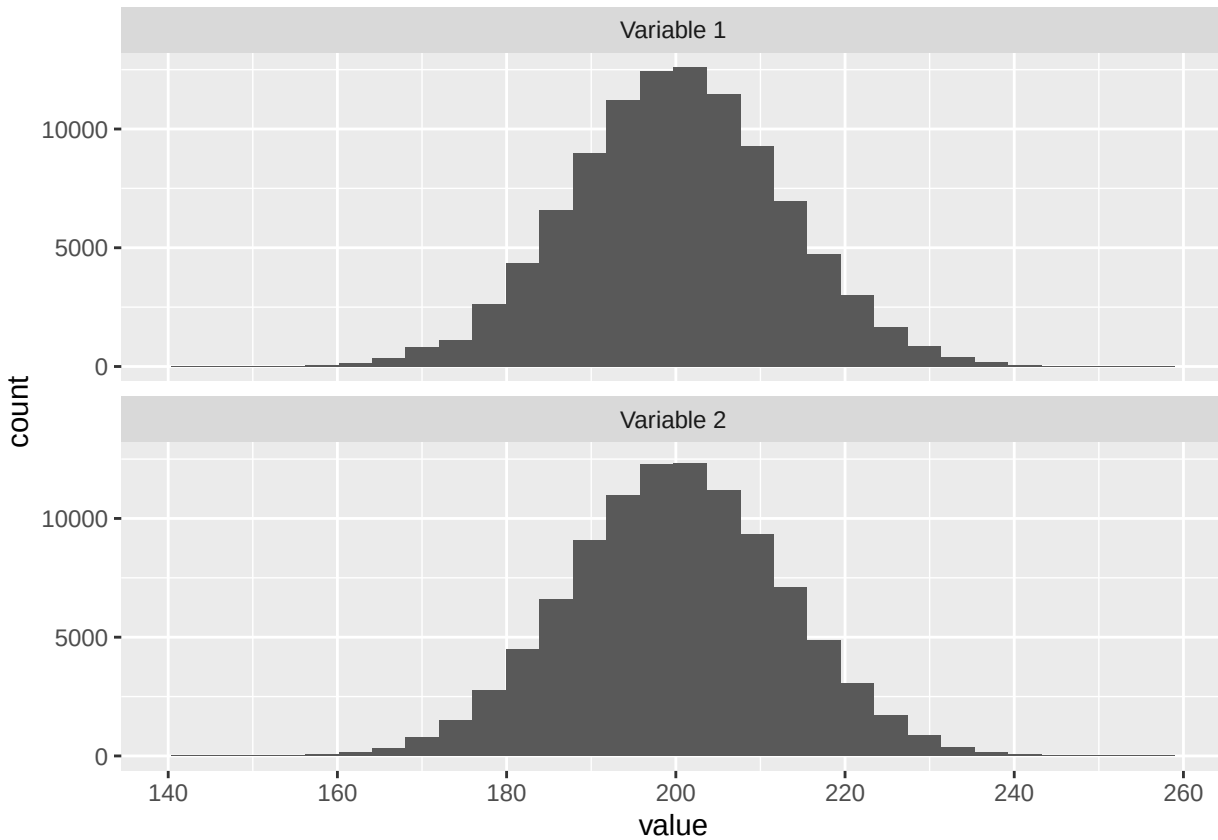
Related distributions

The normal distribution

Note that:

- The normal distribution is an approximation of the binomial distribution, particularly for large sample size (large n) and/or π is close to 1/2.

```
library(ggplot2)
# Draw a random sample of 100,000 from the Binomial(1000, .2) distribution
binom_sample <- rbinom(100000, 1000, .2)

# Draw a random sample of 100,000 from the normal approximation
normal_sample <- rnorm(100000, 1000*0.2, sqrt(1000*0.2*0.8))

# Compare the two distributions with the compare_histograms function
compare_histograms <- function(variable1, variable2) {
    x <- data.frame(value = variable1, variable = "Variable 1")
    y <- data.frame(value = variable2, variable = "Variable 2")
    ggplot(rbind(x, y), aes(value)) + geom_histogram(bins=30) + facet_wrap(~variable,
        nrow = 2)
}

compare_histograms(binom_sample, normal_sample)
```



- Comparing the cumulative density of the binomial:

```
# Simulations from the normal and binomial distributions
binom_sample <- rbinom(100000, 1000, .2)
normal_sample <- rnorm(100000, 200, sqrt(160))

# Use binom_sample to estimate the probability of <= 190 heads
mean(binom_sample<= 190)
```

```
## [1] 0.22643
```

```
# Use normal_sample to estimate the probability of <= 190 heads
mean(normal_sample<= 190)
```

```
## [1] 0.21494
```

```
# Calculate the probability of <= 190 heads with pbinom
pbinom(190, 1000, .2)
```

```
## [1] 0.2273564
```

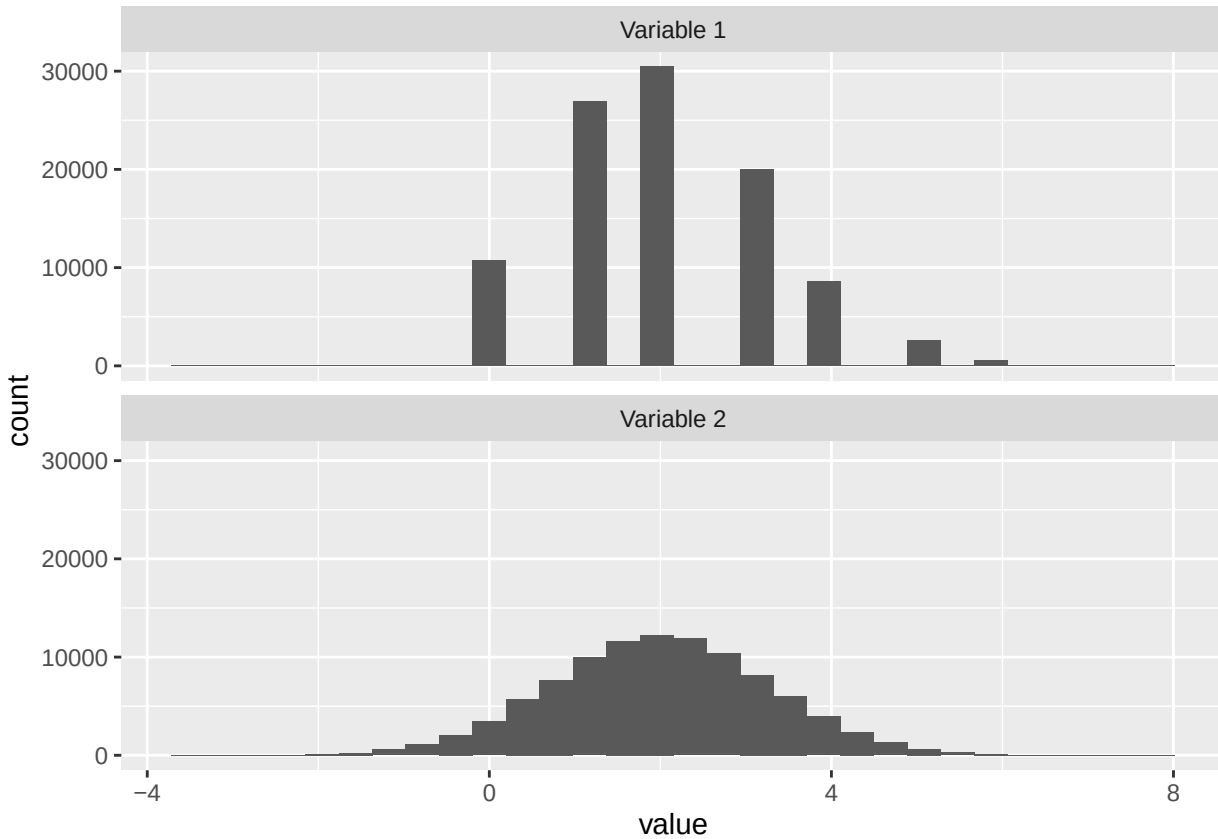
```
# Calculate the probability of <= 190 heads with pnorm
pnorm(190, 1000*0.2, sqrt(1000*0.2*0.8))
```

```
## [1] 0.2145977
```

- For low sample size (low n), the accuracy of the normal approximation decreases:

```
# Draw a random sample of 100,000 from the Binomial(10, .2) distribution
binom_sample <- rbinom(100000, 10, .2)
```

```r
# Draw a random sample of 100,000 from the normal approximation
normal_sample <- rnorm(100000, 10*.2, sqrt(10*.2*.8))

# Compare the two distributions with the compare_histograms function

compare_histograms(binom_sample, normal_sample)
```



The Poisson distribution

As mentioned above, we can use the normal distribution to approxmiate the binomial distribution under with certain conditions, one of which is that π is close to 1/2.

The Poisson distribution can also be used to approximate the binomial distribution when π is low and/or n is large ($n \geq 100$ and $p \leq 0.01$).

This is because:

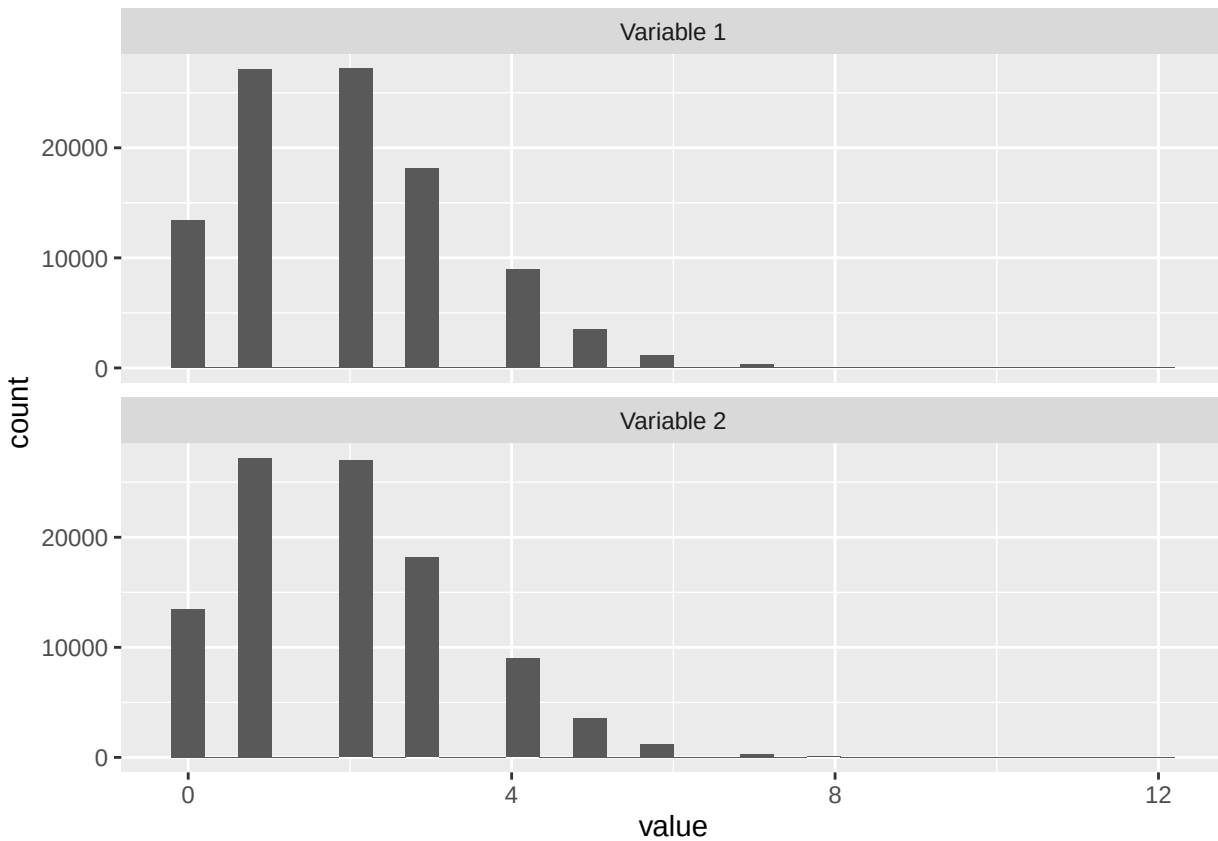
- When $n \to \infty$, $\pi \to 0$, $n\pi \to \lambda$, where λ is a constant.
- So: $Binom(n, \pi) \to Poisson(\lambda)$, i.e:

$$\frac{n!}{x!(n-x)!}\pi^x(1-\pi)^{n-x} \to \frac{e^{-\lambda}\lambda^x}{x!}$$

Intuitively, we can understand the approximation as: We use the Poisson distribution when we only care about the count of each rare event (and the mean), not the total size (in the way we would with the binomial distribution).

1. Simulating from a Poisson and a binomial

```r
# Draw a random sample of 100,000 from the Binomial(1000, .002) distribution
binom_sample <- rbinom(100000, 1000, .002)

# Draw a random sample of 100,000 from the Poisson approximation
poisson_sample <- rpois(100000, 2)

# Compare the two distributions with the compare_histograms function

compare_histograms(binom_sample, poisson_sample)
```



2. Density of the Poisson distribution

```r
# Simulate 100,000 draws from Poisson(2)
poisson_sample <- rpois(100000, 2)

# Find the percentage of simulated values that are 0
mean(poisson_sample==0)
```

```
## [1] 0.13906
```

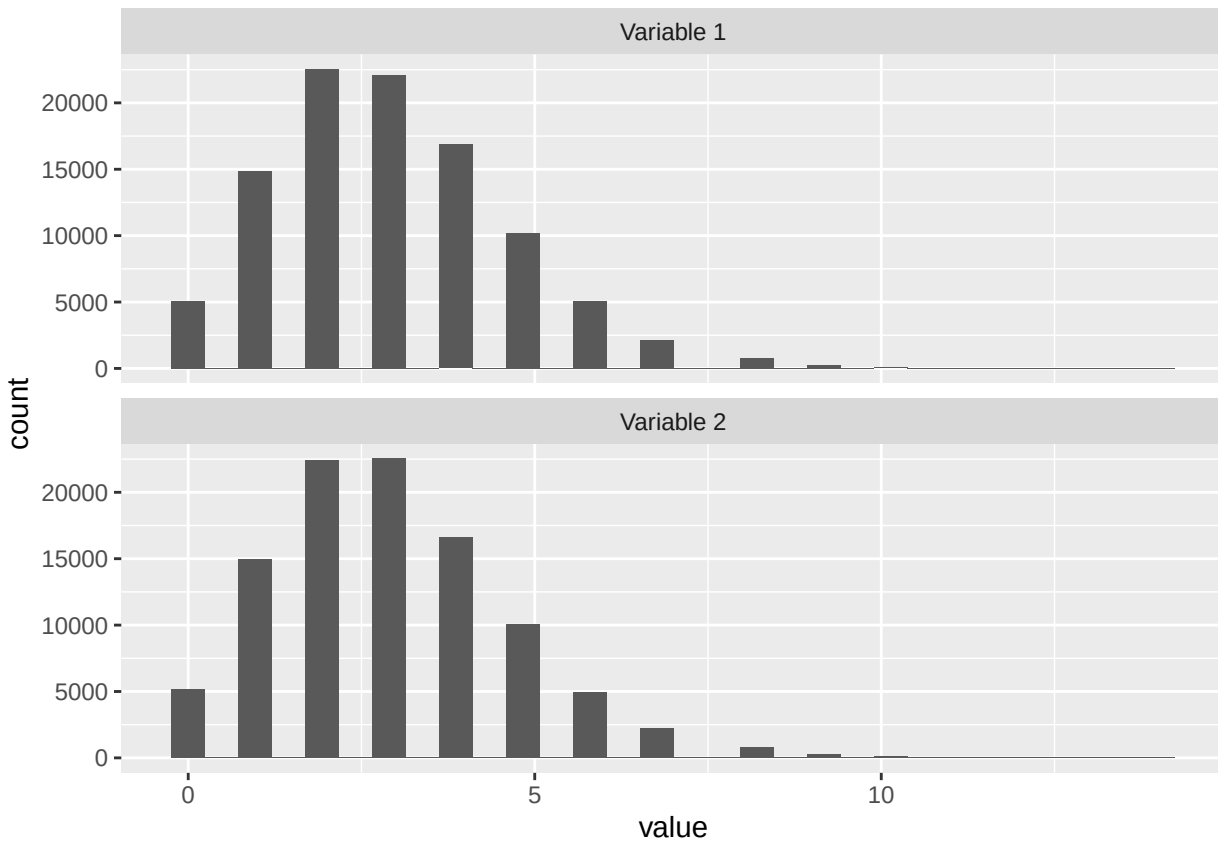
```r
# Use dpois to find the exact probability that a draw is 0
dpois(0, 2)
```

```
## [1] 0.1353353
```

3. Sum of two Poisson variables

One of the useful properties of the Poisson distribution is that when you add multiple Poisson distributions together, the result is also a Poisson distribution.

```
# Simulate 100,000 draws from Poisson(1)
X <- rpois(100000, 1)

# Simulate 100,000 draws from Poisson(2)
Y <- rpois(100000, 2)

# Add X and Y together to create Z
Z<- X+Y

# Use compare_histograms to compare Z to the Poisson(3)
compare_histograms(Z, rpois(100000, 3))
```



The geometric distribution

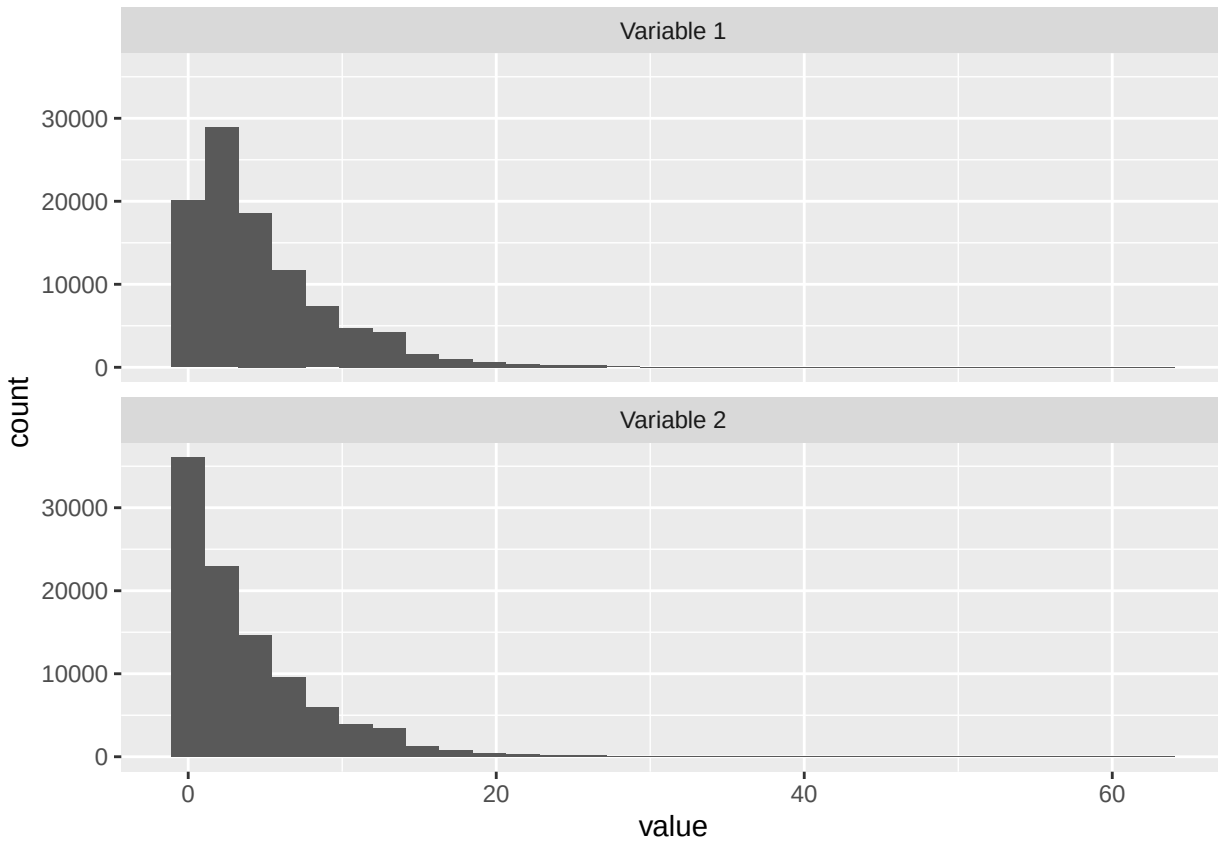
The geometric distribution is the probability distribution of the number X of Bernoulli trials needed to get one success, supported on the set $1, 2, 3, \ldots$.

Given $X \sim Geom(p)$, we have $E[X] = \frac{1}{p} - 1$.

We can simulate this process that gives the geometric distribution in R as below:

```
# Existing code for finding the first instance of heads
# which(rbinom(100, 1, 0.2) == 1)[1]

# Replicate this 100,000 times using replicate()
```

```r
replications <- replicate(100000, which(rbinom(100, 1, 0.2) == 1)[1])

# Generate 100,000 draws from the corresponding geometric distribution
geom_sample <- rgeom(100000, .2)

# Compare the two distributions with compare_histograms
compare_histograms(replications, geom_sample)
```



Example 1: A new machine arrives in a factory. This type of machine is very unreliable: every day, it has a 10% chance of breaking permanently. How long would you expect it to last?

```r
# Find the probability the machine breaks on 5th day or earlier

pgeom(4, .1, lower.tail=FALSE)
```

```
## [1] 0.59049
```

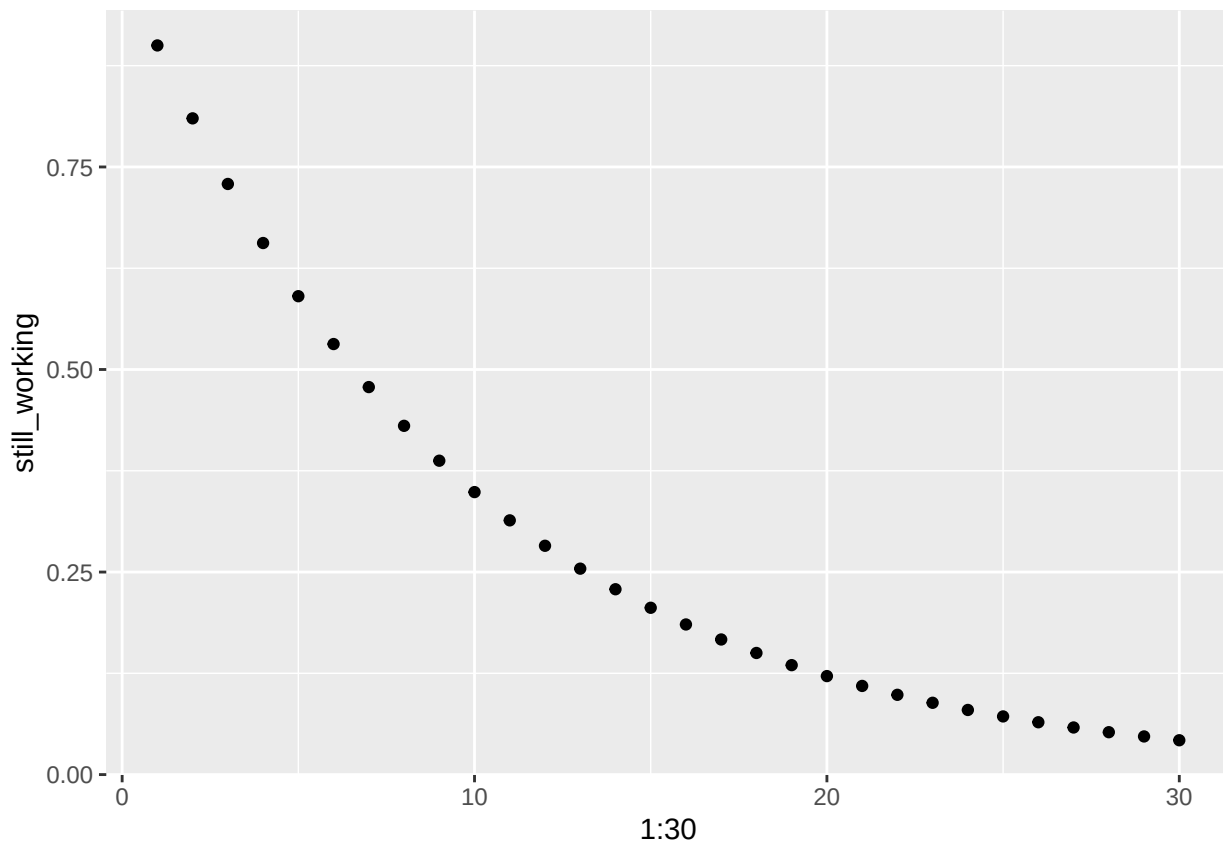
```r
# Find the probability the machine is still working on 20th day

pgeom(19, .1, lower.tail=FALSE)
```

```
## [1] 0.1215767
```

Note that:

- The cumulative distribution function gives the probability that the machine *might breaks on day X or earlier*.
- For the first question, we have: $P(breaks\ on\ 5^{th}\ day\ or\ earlier)$ is given by the CDF.

- For the second question, we have: $P(lasts\ for\ 19\ days) = 1 - P(breaks\ on\ day\ 20^{th}\ or\ earlier) = 1 - CDF$.

Example 2: Graphing the probability that a machine still works in 30 days.

```
# Calculate the probability of machine working on day 1-30
still_working <- pgeom(0:29, .1, lower.tail=FALSE)

# Plot the probability for days 1 to 30
qplot(1:30, still_working)
```