Discrete Distributions in R

We love working in R because it's made for statisticians! All of the "named" distributions have been implemented in R. We will first see how to evaluate the PMF and CDF of some named distributions, and then how to simulate observations for these distributions.

Binomial distribution

Suppose $X \sim \text{Binom}(n = 5, p = 0.2)$.

Evaluating the PMF

Recall that the PMF of a discrete RV X is $f_X(x) = P(X = x)$, where $x \in \mathbb{R}$.

I might want to know, what is $P(X = 3)$? We will use the function `dbinom()`, which takes in three arguments:

```
# x: where we'd like to evaluate the PMF at
# size: n in Binom(n, p)
# prob: p in Binom(n, p)
dbinom(x = 3, size = 5, prob = 0.2)
```

```
[1] 0.0512
```

A quick confirmation using the Binomial PMF:

```
choose(5,2) * 0.2^3 * (1-0.2)^(5-3)
```

```
[1] 0.0512
```

Evaluating the CDF

What if I want to know $F_X(2) = P(X \leq 2)$? We will use the function `pbinom()`, which also takes in three arguments, in the same order as in `dbinom()`, but we just remember that we'll be obtaining the CDF, not PMF:

```r
# q: where we'd like to evaluate the CDF at
# size: n in Binom(n, p)
# prob: p in Binom(n, p)
pbinom(q = 2, size = 5, p = 0.2)
```

```
[1] 0.94208
```

A quick confirmation:

```r
choose(5,0) * 0.2^0 * (1-0.2)^(5-0) +
  choose(5,1) * 0.2^1 * (1-0.2)^(5- 1) +
  choose(5,2) * 0.2^2 * (1-0.2)^(5-2)
```

```
[1] 0.94208
```

Simulating/generating random variables

Suppose I want to simulate a single observation of X (e.g. flipping a coin 5 times with probability 0.2 of Heads, and counting how many Heads occurred). We can do that with `rbinom()`:

```r
# first argument: number of observations to generate
# size: n in Binom(n, p)
# prob: p in Binom(n, p)
rbinom(1, size = 5, p = 0.2)
```

```
[1] 1
```

I can change how many times I observe this process by changing the first argument. So if I instead I want to repeat the experiment 10 times:

```r
rbinom(10, size = 5, p = 0.2)
```

```
 [1] 0 4 1 3 1 2 1 2 2 2
```

Each of the numbers above is the number of successes in 5 trials.

Hypergeometric Distribution

The nice thing about R is the the syntax of `d<dist>()`, `p<dist>()`, and `r<dist>()` is consistent! We just have to remember the shorthand for the `<dist>`. So if $X \sim \text{HyperGeom}(4, 3, 2)$, we have the following:

```
# P(X = 2)
# here, m = number of white balls (w), n = number of black balls (b),
# and k = number of balls drawn (n)
dhyper(x = 2, m = 4, n = 2, k = 3)
```

```
[1] 0.6
```

```
# P(X <= 2)
phyper(q = 2, m = 4, n = 2, k = 3)
```

```
[1] 0.8
```

```
# simulate 1 observation of X (i.e. number many white balls drawn)
rhyper(1, m = 4, n = 2, k = 3)
```

```
[1] 3
```

Discrete Uniform

The Discrete Uniform doesn't need a fancy R function because we can get the PMF and CDF extremely easily. To generate observations from this distribution, we use the `sample()` function because it, by default, treats all objects in a vector as equally likely to be randomly sampled.

Suppose $X \sim \text{DiscreteUnif}(\{1, 2, 4, 10\})$ and I'd like to sample 1 observation from this distribution.

```
sample(c(1,2,4,10), size = 1)
```

```
[1] 10
```

If I want to sample 3 observations, I need to specify that I'd like to sample with replacement:

```r
sample(c(1,2,4,10), size = 3, replace = T)
```

```
[1]  2 10  4
```

Sampling any discrete distribution with finite support

We can also use the `sample()` function to generate observations from any discrete distribution with finite support by changing a single argument. Remember that a PMF just specifies the probabilities associated with each $x \in S_X$. Suppose I have a discrete RV with the following PMF:

$$P(X = x) = \begin{cases} 0.5 & \text{if } x = 0 \\ 0.4 & \text{if } x = 1 \\ 0.1 & \text{if } x = 2 \\ 0 & \text{otherwise} \end{cases}$$

We can sample from this distribution by simply specifying these probabilities in the `prob` argument:

```r
sample(c(0,1,2), size = 10, replace = T, prob = c(0.5, 0.4, 0.1))
```

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

Note that the order of probabilities provided must match the order of the values in the support!