Distributions

1. Introduction

This is the R track of the *Distributions* workshop prepared by Los Angeles County, ISAB. Section numbering is intended to be consistently referenced from the main workshop document available at

https://github.com/lacounty-isab/workshops/tree/master/distributions

1.1. Preparation

Since R is oriented toward statistics, no special preparation is required to work with distributions. You can obtain a list of *baked-in* distributions by entering

```
?Distributions
```

These are all part of the **stats** package which is available in every R installation.

1.2 Conventions

R has a consistent naming convention for functions that work with distributions - a single letter followed by the name of the distribution. The four single letters are

- `d` - density function
- `p` - percent point function (CDF)
- `q` - inverse of CDF
- `r` - random sampler

If we take the binomial distribution as an example, then `dbinom` is a *binomial density function.*

```
dbinom(4, 10, 0.3)
```

```
## [1] 0.2001209
```

This gives the probability of getting 4 successes after 10 attempts where each success has a probability of 0.3; which is about 20 percent.

2 Discrete Distributions

2.1 Binomial Distribution

The binomial distribution has a `binom` suffix for each of its standard methods. We'll assume 100 trials with 20% chance of success at each trial. The value of the random variable is the number of successes. Let's start with the density function `dbinom`. This is equation (7) of the math supplement.

$$f_X(i) = \sum_{i=0}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

Here we consider the case of $n = 100$ and $p = 0.2$. We suspect that the highest probability will be around 20 successes since the probability of each success is 0.2 and there are 100 trials. The `dbinom` function accepts numeric vectors, so we'll check `10`, `20`, and `30` in one go.

```r
dbinom(c(10, 20, 30), 100, 0.2)
```

```
## [1] 0.003362820 0.099300215 0.005189643
```

Sure enough, the density functon shows a 9.9% chance of obtaining exactly 20 successes. The probability for 10 and 30 are each less than 1%.

Now let's check the associated cumulative distribution function, or CDF. The CDF at a point is the cumulation of probability from the density function for all points equal or less. Since we expect the average to be around 20, the CDF should be close to 0.5 at 20.

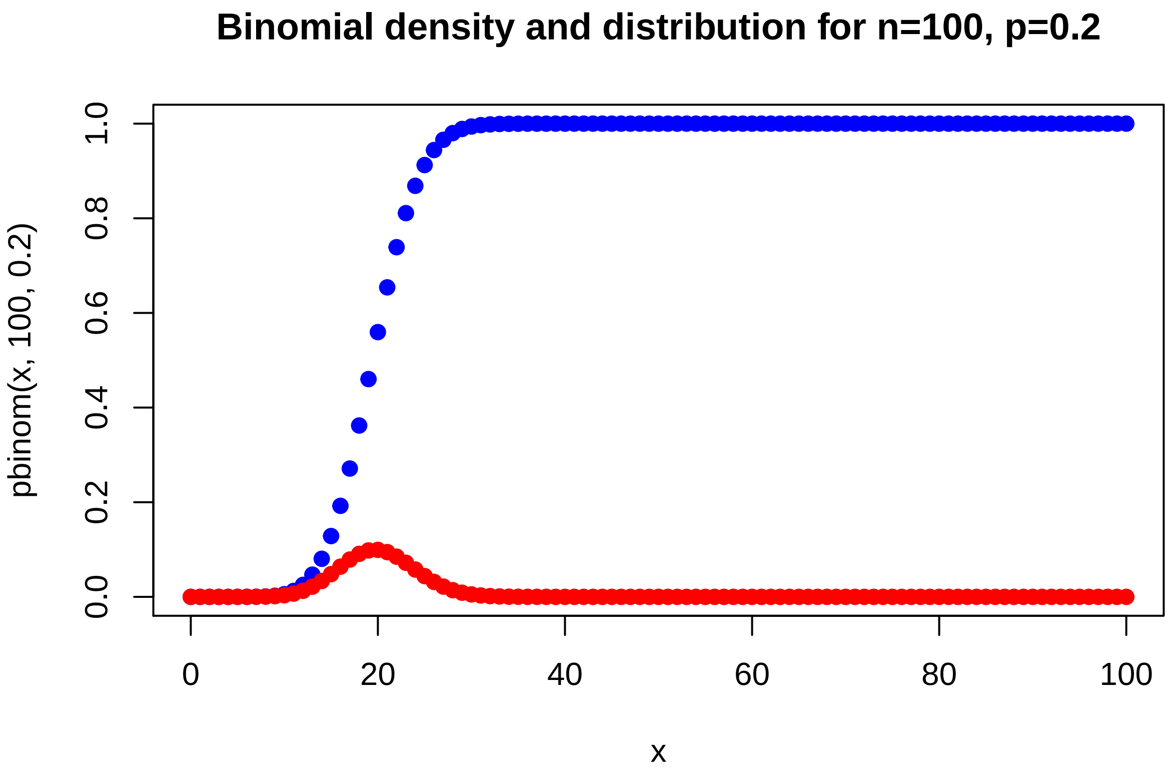
```r
pbinom(c(10, 20, 30), 100, 0.2)
```

```
## [1] 0.005696381 0.559461585 0.993940665
```

This shows that the probability of having *less than or equal to 10 successes* is 0.57%. The probability of having less than or equal to 30 successes is 99.4%.

To better see the relationship between the density and cumulative distributions, we can plot them.

```r
x <- seq(0, 100)
plot(x, pbinom(x, 100, 0.2), col="blue", pch=19)
points(x, dbinom(x, 100, 0.2), col="red", pch=19)
title(main="Binomial density and distribution for n=100, p=0.2")
```



Quantiles are the inverse of the CDF function. For any probability, a quantile function tells us the value of the random variable for which the CDF is equal to that probability.

How many successes can we expect to have 25%, 50%, and 75% of the time?

```r
qbinom(c(0.25, 0.5, 0.75), 100, 0.2)
```

```
## [1] 17 20 23
```

The reason these come out to exact integers is because the binomial random variable only takes integer values. The `qbinom` can be interpreted as *the smallest integer number of successes with probability greater than or equal to the argument*. So it's not a strict inverse.

```
pbinom(c(16, 17, 19, 20, 22, 23), 100, 0.2)
```

```
## [1] 0.1923376 0.2711890 0.4601614 0.5594616 0.7389328 0.8109128
```

Sometimes it's useful to generate samples from a distribution. In other words, the numbers are generated as if they came from random variable values associated with outcomes of the experiement. The `rbinom` function is used for this.

```
b_sample <- rbinom(1000, 100, 0.2)
b_sample[1:100]
```

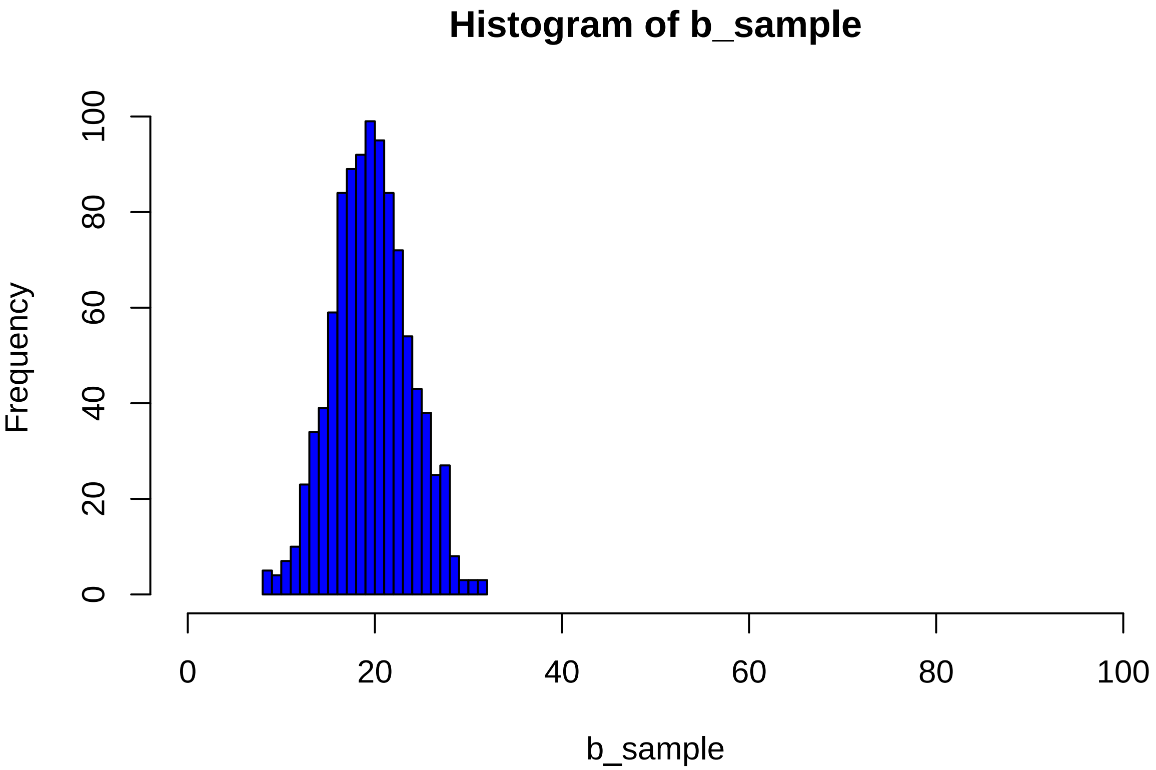
```
##   [1] 25 20 23 19 22 19 21 23 17 18 27 19 14 11 26 20 18 18 15 17 17 25 16
##  [24] 23 17 24 17 19 17 24 18 22 21 23 22 17 22 20 12 23 24 31 27 15 15 18
##  [47] 18 18 16 16 17 26 23 21 19 18 25 17 17 17 22 22 29 19 20 18 21 18 16
##  [70] 15 21 17 17 12 17 25 17 18 20 17 23 21 23 16 26 12 10 18 19 21 17 19
##  [93] 24 18 14 19 19 18 13 26
```

This simulates the arduous task of

1. flipping an unfair coin (with 20% chance of heads) 100 times,
2. noting the number of heads,
3. repeating steps 1 and 2 for 1,000 times

It's fun to plot a histogram of our sample. It should have a shape "approaching" our density function.

```
hist(b_sample, breaks=30, xlim=c(0, 100), col="blue")
```


Histogram of b_sample
Frequency
b_sample