Section 2: identification and simulation (solutions)

Advanced Quantitative Methods (PLSC 504)

Fall 2017

The first part of this handout is a review problem that we will solve together in section. Please try and make some progress on your own before we meet. The second part of the handout reviews some basic distributions and how to use them in `R`. The remainder of the handout contains some notes on simulation with some illustrations in `R`.

Review Problem

Suppose that D_i and Z_i are binary[1]. Let

- $\mathbb{E}[Y_i|D_i = 1, Z_i = 1] = 1.00$
- $\Pr[D_i = 1, Z_i = 1] = 0.25$
- $\mathbb{E}[Y_i|D_i = 0, Z_i = 1] = 0.50$
- $\Pr[D_i = 0, Z_i = 1] = 0.50$
- $\mathbb{E}[Y_i|D_i = 0, Z_i = 0] = 0.50$
- $\Pr[D_i = 0, Z_i = 0] = 0.25$
- $\Pr[D_i = 1, Z_i = 0] = 0.00$

Assumptions:

1. $Y_i = Y_i(1)D_i + Y_i(0)(1 - D_i)$.
2. $D_i = D_i(1)Z_i + D_i(0)(1 - Z_i)$
3. $(Y_i(0), Y_i(1)) \perp\!\!\!\perp D_i$
4. $(Y_i(0), Y_i(1), D_i(1), D_i(0)) \perp\!\!\!\perp Z_i$
5. $\text{Supp}[Y_i] = [\alpha, \beta]$

Define $\tau_i := Y_i(1) - Y_i(0)$. The following tool (Law of Iterated Expectations) will be necessary:

For r.v. X and Y, if f is a function of X and Y, then $\mathbb{E}[f(X, Y)] = \mathbb{E}\left[\mathbb{E}[h(X, Y)|X]\right]$

Exercise:

We are told the *joint distribution* of of D_i and Z_i in the setup. This gives us a great deal of information,

$$f(D_i, Z_i) = \begin{cases} 0.25 & \text{if } D_i = 1, Z_i = 1; \\ 0.50 & \text{if } D_i = 0, Z_i = 1; \\ 0.25 & \text{if } D_i = 0, Z_i = 0; \\ 0 & \text{otherwise.} \end{cases}$$

Recall that: $f(y|x) = \frac{f(y,x)}{f(x)}$. We can deduce from the joint distribution facts about the marginal and conditional distributions,

[1] This example is stolen from an old midterm exam question in Peter Aronow's PLSC 503 class

$$f(D_i) = \begin{cases} 0.25 & \text{if } D_i = 1; \\ 0.75 & \text{if } D_i = 0; \\ 0 & \text{otherwise.} \end{cases}$$

$$f(Z_i) = \begin{cases} 0.75 & \text{if } Z_i = 1; \\ 0.25 & \text{if } Z_i = 0; \\ 0 & \text{otherwise.} \end{cases}$$

$$f(D_i|Z_i = 1) = \begin{cases} 0.25/0.75 = 1/3 & \text{if } D_i = 1; \\ 0.50/0.75 = 2/3 & \text{if } D_i = 0; \\ 0 & \text{otherwise.} \end{cases}$$

$$f(D_i|Z_i = 0) = \begin{cases} 0.25/0.25 = 1 & \text{if } D_i = 1; \\ 0 & \text{otherwise.} \end{cases}$$

The manifestation of the Law of Iterated Expectations (LIE) here is of the simple discrete form: $\mathbb{E}[Y] = \mathbb{E}\left[\mathbb{E}[Y|X]\right] = \sum_x \mathbb{E}\left[Y|X=x\right] f(x)$. This is used throughout.

(a) Under assumptions (1) and (5), compute a sharp upper bound for $\mathbb{E}[Y_i(1)|Z_i = 0]$.

$$\begin{aligned} \mathbb{E}\left[Y_i(1)|Z_i = 0\right] &= \mathbb{E}\left[Y_i(1)|D_i = 1, Z_i = 0\right] \cdot \Pr\left(D_i = 1|Z_i = 0\right) + \mathbb{E}\left[Y_i(1)|D_i = 0, Z_i = 0\right] \cdot \Pr\left(D_i = 0|Z_i = 0\right) \\ &= \mathbb{E}\left[Y_i(1)|D_i = 1, Z_i = 0\right] \cdot 0 + \mathbb{E}\left[Y_i(1)|D_i = 0, Z_i = 1\right] \cdot 1 \\ &= ? \cdot 0 + \beta \cdot 1 = \beta \end{aligned}$$

The first line follows from the application of LIE. It is sufficient to simply note that outcomes are undefined when $D_i = 1$ and $Z_i = 0$ so that, without additional assumptions, the upper bound for $\mathbb{E}[Y_i(1)|Z_i = 0]$ is $\max\{\text{Supp}[Y_i]\} = \beta$.

(b) Under assumption (1), compute $\mathbb{E}[Y_i(0)|Z_i = 0]$.

$$\begin{aligned} \mathbb{E}\left[Y_i(0)Z_i(0)\right] &= \mathbb{E}\left[Y_i(0)|D_i = 0, Z_i = 0\right] \cdot \Pr\left(D_i = 0|Z_i = 0\right) + \mathbb{E}\left[Y_i(0)|D_i = 1, Z_i = 0\right] \cdot \Pr\left(D_i = 1|Z_i = 0\right) \\ &= \mathbb{E}[Y_i(0)|D_i = 0, Z_i = 0] \cdot 1 + \mathbb{E}[Y_i(0)|D_i = 1, Z_i = 0] \cdot 0 \\ &= \mathbb{E}[Y_i|D_i = 0, Z_i = 0] \text{ (by Assumption i.)} \\ &= 0.50 \end{aligned}$$

(c) Under assumptions (1) and (3), compute $\mathbb{E}[Y_i(1) - Y_i(0)]$.

$$\begin{aligned}
\mathbb{E}[Y_i(1) - Y_i(0)] &= \mathbb{E}[Y_i(1)] - \mathbb{E}[Y_i(0)] \text{ (linearity of expectations)} \\
&= \mathbb{E}[Y_i|D_i = 1] - \mathbb{E}[Y_i|D_i = 0] \text{ (by Assumption i.)} \\
&= \mathbb{E}\left[Y_i|D_i = 1, Z_i = 0\right] \cdot \Pr(Z_i = 0|D_i = 1) + \mathbb{E}\left[Y_i|D_i = 1, Z_i = 1\right] \cdot \Pr(Z_i = 1|D_i = 1) \\
&\qquad \left[\mathbb{E}\left[Y_i|D_i = 0, Z_i = 0\right] \cdot \Pr(Z_i = 0|D_i = 0) + \mathbb{E}\left[Y_i|D_i = 0, Z_i = 1\right] \cdot \Pr(Z_i = 1|D_i = 0)\right] \text{ (by LIE)} \\
&= \mathbb{E}\left[Y_i|D_i = 1, Z_i = 0\right] \cdot \Pr(Z_i = 0|D_i = 1) + 1 \cdot \Pr(Z_i = 1|D_i = 1) \\
&\qquad - 0.5 \cdot \Pr(Z_i = 0|D_i = 0) - 0.5 \cdot \Pr(Z_i = 1|D_i = 0) \\
&= \mathbb{E}[Y_i|D_i = 1, Z_i = 0] \cdot \frac{\Pr(D_i = 1, Z_i = 0)}{\Pr(D_i = 1)} + 1 \cdot \frac{\Pr(D_i = 1, Z_i = 1)}{\Pr(D_i = 1)} \\
&\qquad - 0.5 \cdot \frac{\Pr(D_i = 0, Z_i = 0)}{\Pr(D_i = 0)} - 0.5 \cdot \frac{\Pr(D_i = 0, Z_i = 1)}{\Pr(D_i = 0)} \\
&= \mathbb{E}[Y_i|D_i = 1, Z_i = 0]\frac{0}{0.25} + 1 \cdot \frac{0.25}{0.25} - 0.5 \cdot \frac{0.5}{0.75} - 0.5\frac{0.25}{0.75} \\
&= 0 + 1 - \left(\frac{1}{2}\right)\frac{2}{3} - \left(\frac{1}{2}\right)\frac{1}{3} = \frac{1}{2}
\end{aligned}$$

(d) Under assumptions (1), (2) and (4), compute $\Pr[D_i(1) > D_i(0)]$, $\Pr[D_i(1) = D_i(0) = 1]$, $\Pr[D_i(1) = D_i(0) = 0]$, and $\Pr[D_i(1) < D_i(0)]$.

$\Pr(D_i = 1, Z_i = 0) = 0$. So $D_i(0) = 0\ \forall i$. This is a very helpful fact, and implies the following,

$$\begin{aligned}
&\Pr\left(D_i(1) = D_i(0) = 1\right) = 0 \\
&\Pr\left(D_i(1) < D_i(0)\right) = 0
\end{aligned}$$

The first line above states that the probability of observing $D_i(1) = 0 = 1$ is zero, which is true because of what we know about $D_i(0)$. The second line above states that the probability that $D_i(1) < 0$ is zero. Again, follows directly from what we know about $D_i(0)$. Next,

$$\begin{aligned}
\Pr(D_i(1) > D_i(0)) &= \Pr(D_i(1) > 0) = \Pr(D_i(1) = 1) \text{ (since } D_i \text{ is either 0 or 1)} \\
&= \Pr(D_i(1) = 1|Z_i = 1) \text{ (by Assumption iv.)} \\
&= \Pr(D_i = 1|Z_i = 1) \text{ (by Assumption ii.)} \\
&= \frac{\Pr(D_i = 1, Z_i = 1)}{\Pr(Z_i = 1)} \\
&= \frac{0.25}{0.75} = \frac{1}{3}
\end{aligned}$$

Finally, putting it all together we have,

$$\begin{aligned}
\Pr(D_i(1) = D_i(0) = 0) &= \Pr(D_i(1) = 0) \\
&= 1 - \left[\Pr\left(D_i(1) = 0 = 1\right) + \Pr\left(D_i(1) < 0\right)\right) + \Pr(D_i(1) > 0)] \\
&= 1 - \left[0 + 0 + \frac{1}{3}\right] = \frac{2}{3}
\end{aligned}$$

(e) Under assumptions (1), (2) and (4), compute $\mathbb{E}[Y_i(1) - Y_i(0)|D_i(1) > D_i(0)]$.

This is an application of the famous LATE (Local Average Treatment Effect) Theorem, discusssed in more detail in the next section.

$$\mathbb{E}[Y_i(1) - Y_i(0)|D_i(1) > D_i(0)] = \frac{\mathbb{E}[Y_i|Z_i = 1] - \mathbb{E}[Y_i|Z_i = 0]}{\mathbb{E}[D_i|Z_i = 1] - \mathbb{E}[D_i|Z_i = 0]}$$

Starting with the numerator (applying LIE),

$$\begin{aligned}\mathbb{E}[Y_i|Z_i=1] &= \mathbb{E}\left[Y_i|Z_i=1,D_i=1\right]\cdot\frac{\Pr(Z_i=1,D_i=1)}{\Pr(Z_i=1)}+\mathbb{E}\left[Y_i|Z_i=1,D_i=0\right]\cdot\frac{\Pr(Z_i=1,D_i=0)}{\Pr(Z_i=1)}\\ &= 1\cdot\frac{0.25}{0.75}+0.5\cdot\frac{0.5}{0.75}=\frac{1}{3}+\frac{1}{3}=\frac{2}{3}\end{aligned}$$

$$\begin{aligned}\mathbb{E}[Y_i|Z_i=0] &= \mathbb{E}\left[Y_i|Z_i=0,D_i=1\right]\cdot\frac{\Pr(Z_i=0,D_i=1)}{\Pr(Z_i=0)}+\mathbb{E}\left[Y_i|Z_i=0,D_i=0\right]\cdot\frac{\Pr(Z_i=0,D_i=0)}{\Pr(Z_i=0)}\\ &= 0+0.5\cdot\frac{0.25}{0.25}=\frac{1}{2}\end{aligned}$$

Now for the denominator,

$$\begin{aligned}\mathbb{E}[D_i|Z_i=1] &= \Pr(D_i=1|Z_i=1)=\frac{1}{3}\\ \mathbb{E}[D_i|Z_i=0] &= \Pr(D_i=1|Z_i=0)=0\end{aligned}$$

Putting it all together,

$$\frac{\mathbb{E}[Y_i|Z_i=1]-\mathbb{E}[Y_i|Z_i=0]}{\mathbb{E}[D_i|Z_i=1]-\mathbb{E}[D_i|Z_i=0]}=\frac{2/3-1/2}{1/3-0}=\frac{1/6}{1/3}=\frac{1}{2}$$

Summary notes about IV

Typical assumptions:

1. SUTVA: $Y_i=Y_i(1)D_i+Y_i(0)(1-D_i)$
   - the potential treatments and outcomes $(D_i(0),D_i(1),Y_i(0),Y_i(1))$ are stable
2. Exclusion restriction: $D_i=D_i(1)Z_i+D_i(0)(1-Z_i)$
   - the instrument Z_i is *excluded* from the equation determining Y_i
3. Monotonicity: $D_i(1)\geq D_i(0)$
4. Encouragement (non-zero effect):

$$\begin{aligned}&\mathbb{E}[D_i(1)-D_i(0)]>0\\ &\mathbb{E}[D_i|Z_i=1]>0\ \text{ provided (2) holds}\end{aligned}$$

5. Strong ignorability of instrument:

$$(Y_i(0),Y_i(1),D_i(0),D_i(1))\perp\!\!\!\perp Z_i$$

Provided the above assumptions hold, the Local Average Treatment Effect (LATE), also called the Complier Average Causal Effect (CACE), is point identified. This is the average treatment effect for units that *would take up the treatment if assigned, but would not take up the treatment if not assigned*:

$$\mathbb{E}[\tau_i|D_i(1)>D_i(0)]=\frac{\mathbb{E}[Y_i|Z_i=1]-\mathbb{E}[Y_i|Z_i=0]}{\mathbb{E}[D_i|Z_i=1]-\mathbb{E}[D_i|Z_i=0]}$$

Under ignorability, the numerator is called the *intention to treat* (ITT) effect of $Z_i on Y_i$ and the denominator is the *first-stage effect* of Z_i on D_i.

With binary Z_i, D_i and SUTVA, we have four possible 'types':

1. *always takers*: $D_i(1) = D_i(0) = 1$
2. *never takers*: $D_i(1) = D_i(0) = 0$
3. *compliers*: $D_i(1) = 1,\ D_i(0) = 0 := D_i(1) > D_i(0)$
4. *defiers*: $D_i(1) = 0,\ D_i(0) = 1 := D_i(1) < D_i(0)$

The LATE can also be written as:

$$\frac{\mathbb{E}[\tau_i | D_i(1) > D_i(0)] \cdot \Pr\left(D_i(1) > D_i(0)\right) - \mathbb{E}[\tau_i | D_i(1) < D_i(0)] \cdot \Pr\left(D_i(1) < D_i(0)\right)}{\Pr[D_i(1) > D_i(0))] - \Pr[D_i(1) < D_i(0))]}$$

Monotonicity implies,

$$\frac{\mathbb{E}[\tau_i | D_i(1) > D_i(0)] \Pr\left(D_i(1) > D_i(0)\right)}{\Pr\left(D_i(1) > D_i(0)\right)} = \mathbb{E}[\tau_i | D_i(1) > D_i(0)]$$

Proof (tedious):

$$\begin{aligned}
\mathbb{E}[D_i | Z_i = 1] &= \mathbb{E}[D_i | Z_i = 1, D_i(1) = D_i(0) = 1] \cdot \Pr\left(D_i(1) = D_i(0) = 1\right) \\
&\quad + \mathbb{E}[D_i | Z_i = 1, D_i(1) = D_i(0) = 0] \cdot \Pr\left(D_i(1) = D_i(0) = 0\right) \\
&\quad + \mathbb{E}[D_i | Z_i = 1, D_i(1) > D_i(0)] \cdot \Pr\left(D_i(1) > D_i(0)\right) \\
&\quad + \mathbb{E}[D_i | Z_i = 1, D_i(1) < D_i(0)] \cdot \Pr\left(D_i(1) < D_i(0)\right) \\
&= 1 \cdot \Pr\left(D_i(1) = D_i(0) = 1\right) + 0 \cdot \Pr\left(D_i(1) = D_i(0) = 0\right) \\
&\quad + 1 \cdot \Pr\left(D_i(1) > D_i(0)\right) + 0 \cdot \Pr\left(D_i(1) < D_i(0)\right) \\
&= \Pr\left(D_i(1) = D_i(0) = 1\right) + \Pr\left(D_i(1) > D_i(0)\right) \\
&= \text{always-takers + compliers}
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[D_i | Z_i = 0] &= \mathbb{E}[D_i | Z_i = 0, D_i(1) = D_i(0) = 1] \cdot \Pr\left(D_i(1) = D_i(0) = 1\right) \\
&\quad + \mathbb{E}[D_i | Z_i = 0, D_i(1) = D_i(0) = 0] \cdot \Pr\left(D_i(1) = D_i(0) = 0\right) \\
&\quad + \mathbb{E}[D_i | Z_i = 0, D_i(1) > D_i(0)] \cdot \Pr\left(D_i(1) > D_i(0)\right) \\
&\quad + \mathbb{E}[D_i | Z_i = 0, D_i(1) < D_i(0)] \cdot \Pr\left(D_i(1) < D_i(0)\right) \\
&= 1 \cdot \Pr\left(D_i(1) = D_i(0) = 1\right) + 0 \cdot \Pr\left(D_i(1) = D_i(0) = 0\right) \\
&\quad + 0 \cdot \Pr\left(D_i(1) > D_i(0)\right) + 1 \cdot \Pr\left(D_i(1) < D_i(0)\right) \\
&= \Pr\left(D_i(1) = D_i(0) = 1\right) + \Pr\left(D_i(1) < D_i(0)\right) \\
&= \text{always-takers + defiers}
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbb{E}[D_i | Z_i = 1] - \mathbb{E}[D_i | Z_i = 0] &= \Pr\left(D_i(1) > D_i(0)\right) - \Pr\left(D_i(1) < D_i(0)\right) \\
&= \text{compliers - defiers}
\end{aligned}$$

From assumption 5, the numerator is:

$$
\begin{aligned}
\mathbb{E}[Y_i|Z_i=1] = {} & \mathbb{E}[Y_i(1)|D_i(1)=D_i(0)=1]\cdot \Pr\left(D_i(1)=D_i(0)=1\right) \\
& + \mathbb{E}[Y_i(0)|D_i(1)=D_i(0)=0]\cdot \Pr\left(D_i(1)=D_i(0)=0\right) \\
& + \mathbb{E}[Y_i(1)|D_i(1)>D_i(0)]\cdot \Pr\left(D_i(1)>D_i(0)\right) \\
& + \mathbb{E}[Y_i(0)|D_i(1)<D_i(0)]\cdot \Pr\left(D_i(1)<D_i(0)\right) \\
\mathbb{E}[Y_i|Z_i=0] = {} & \mathbb{E}[Y_i(1)|D_i(1)=D_i(0)=1]\cdot \Pr\left(D_i(1)=D_i(0)=1\right) \\
& + \mathbb{E}[Y_i(0)|D_i(1)=D_i(0)=0]\cdot \Pr\left(D_i(1)=D_i(0)=0\right) \\
& + \mathbb{E}[Y_i(0)|D_i(1)>D_i(0)]\cdot \Pr\left(D_i(1)>D_i(0)\right) \\
& + \mathbb{E}[Y_i(1)|D_i(1)<D_i(0)]\cdot \Pr\left(D_i(1)<D_i(0)\right)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\mathbb{E}[Y_i|Z_i=1]-\mathbb{E}[Y_i|Z_i=0] = {} & \mathbb{E}[Y_i(1)|D_i(1)>D_i(0)]\cdot \Pr\left(D_i(1)>D_i(0)\right) \\
& + \mathbb{E}[Y_i(0)|D_i(1)<D_i(0)]\cdot \Pr\left(D_i(1)<D_i(0)\right) \\
& - \mathbb{E}[Y_i(0)|D_i(1)>D_i(0)]\cdot \Pr\left(D_i(1)>D_i(0)\right) \\
& - \mathbb{E}[Y_i(1)|D_i(1)<D_i(0)]\cdot \Pr\left(D_i(1)<D_i(0)\right) \\
= {} & \mathbb{E}[\tau_i|D_i(1)>D_i(0)]\cdot \Pr\left(D_i(1)>D_i(0)\right) \\
& - \mathbb{E}[\tau_i|D_i(1)<D_i(0)]\cdot \Pr\left(D_i(1)<D_i(0)\right)
\end{aligned}
$$

What are the *key* assumptions here?

- The instrument is strongly ignorable. (Randomly assigned, at least conditional on covariates.)
- The exclusion restriction: the instrument does not directly affect the outcome *except through the increased take-up of the treatment.*

Nonzero effect of the assignment on the treatment is important but empirically verifiable – is $\mathbb{E}[D_i|Z_i = 1] - \mathbb{E}[D_i|Z_i = 0] > 0$? Monotonicity and SUTVA aren't as important, but it's still nice for them to hold. When monotonicity fails, you get this weighted average (with negative weights) of effects for compliers and defiers. When SUTVA fails, you get an effect that marginalizes over all sorts of different varieties of the assignment and interference.

Distributions

First, let's explore some common distributions and see how to take draws from them in R. We will explore their probability density function (PDF), or probability mass function (PMF) in the discrete case, and their cumulative distribution function (CDF). You don't need to memorize any of these. They are all on Wikipedia.

Uniform distribution

A continuous random variable (r.v.) U has a Uniform distribution on the interval (a, b) if its PDF is,

$$
f(x) = \begin{cases} \frac{1}{b-a} & \text{if } a < x < b \\ 0 & \text{otherwise} \end{cases}
$$

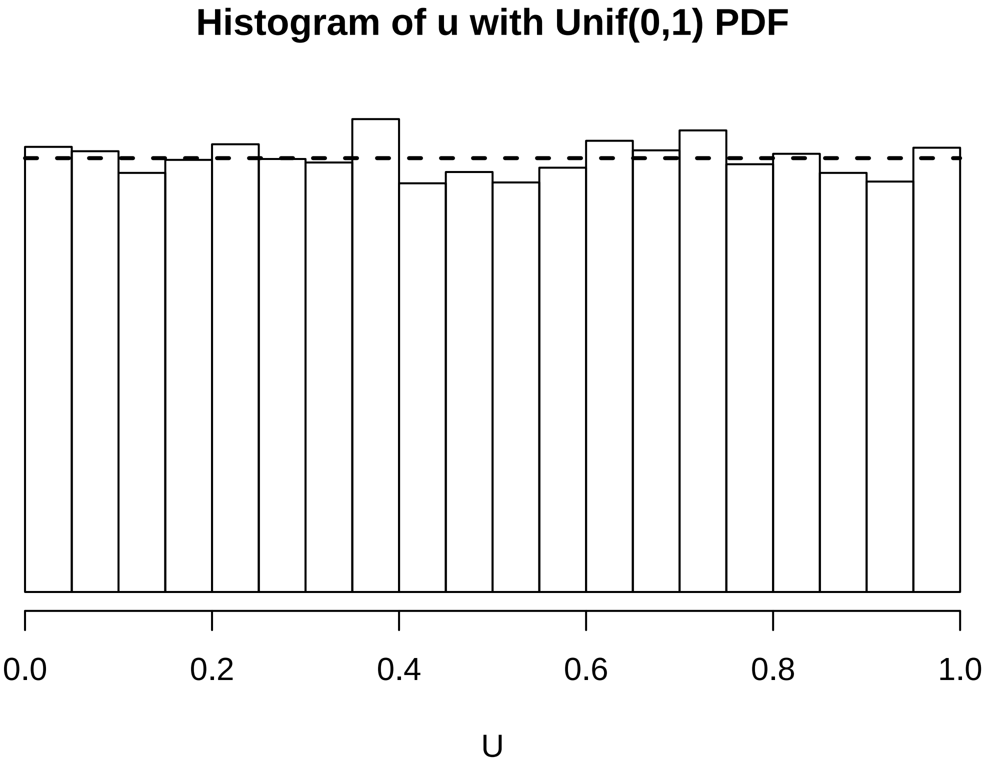
We write this $U \sim \text{Unif}(a, b)$. The $\sim$ means "is distributed as". Its CDF is,

$$
F(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{b-a} & \text{if } a < x < b \\ 1 & \text{if } x \geq b \end{cases}
$$

The standard Uniform, Unif$(0,1)$, is frequently used. The PDF is $f(x) = 1$ and the CDF is $F(x) = x$ for $0 < x < 1$.

Let's generate 10,000 draws from the standard uniform with the `runif()` function in `R` and plot the histogram,

```r
U <- runif(10000)
hist(U, yaxt = "n", ylab = "", prob = TRUE,
     main = "Histogram of u with Unif(0,1) PDF")
curve(dunif(x), lty = 2, lwd = 2, add = TRUE)
```



The standard Uniform distribution is incredibly useful for the following reason: we can use it to construct an r.v. with *any* continuous distribution that we want. This is property is called "Universality of the Uniform"[2] and it works like this.

1. Let $U \sim \text{U}(0,1)$ and $X = F^{-1}(U)$. Then X is an r.v. with CDF F.
2. Let X be an r.v. with CDF F. Then $F(X) \sim \text{Unif}(0,1)$.

Applicaton (from Blitzstein and Hwang p. 207):

The Logistic CDF is

$$F(x) = \frac{e^x}{1 + e^x}$$

for $x \in \mathbb{R}$. We can use $U \sim \text{Unif}(0,1)$ to generate a Logistic r.v.,

First, we know that $F^{-1}(U) \sim \text{Logistic}$. Next, invert the CDF to get F^{-1} by solving $u = F(x)$ for x in terms of u:

[2] See section 5.3 of Blitzstein and Hwang for more details

$$
\begin{aligned}
u &= \frac{e^x}{1+e^x} \\
u &= e^x - ue^x = e^x(1-u) \\
e^x &= \frac{u}{1-u} \\
x &= \log\left(\frac{u}{1-u}\right) \\
F^{-1}(u) &= \log\left(\frac{u}{1-u}\right)
\end{aligned}
$$

Then let $U = u$,

$$
F^{-1}(U) = \log\left(\frac{U}{1-U}\right)
$$

Now $\log\left(\frac{U}{1-U}\right) \sim$ Logistic. Why?

$$
\begin{aligned}
\Pr\left(\log\left(\frac{U}{1-U}\right) \leq x\right) &= \Pr\left(\frac{U}{1-U} \leq e^x\right) \\
&= \Pr\left(U \leq e^x(1-U)\right) \\
&= \Pr\left(U \leq \frac{e^x}{1+e^x}\right) \\
&= \frac{e^x}{1+e^x}
\end{aligned}
$$

We will illustrate this with simulation in the next section.

Bernoulli distribution

A r.v., X has a Bernoulli distribution with parameter p if $\Pr(X = 1) = p$ and $\Pr(X = 0) = 1-p$ for $0 < p < 1$. You will often see this written as $X \sim \text{Bern}(p)$.

This particular distribution is very useful because **ANY** event has a Bernoulli r.v. associated with it, called the *indicator* r.v., which equals 1 if the event happens and 0 otherwise. This is simple yet extremely useful for a variety of computational problems. To formalize this, let W be the r.v. which equals 1 if W occurs and 0 otherwise. You will often see this as I_W or $I(W)$ or $\mathbb{1}\{W\}$. It follows from the previous definition of the Bernoulli that $\mathbb{1}\{W\} \sim \text{Bern}(p)$ where $p = \Pr(W) = \mathbb{E}[\mathbb{1}\{W\}]$.

Binomial distribution

Suppose that n independent Bernoulli trials are performed, each with probability p. Let Y denote the number of "successes" (e.g. cases where we observe 1). $Y \sim \text{Bin}(n, p)$ means that Y has a Binomial distribution with parameters p and n, where n is a positive integer and $0 < p < 1$. The PMF is,

$$
\Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}
$$

for $k = 0, 1, \ldots, n$ and $\Pr(X = k) = 0$ otherwise[3]

We can generate random draws from the Binomial in R with the function rbinom(). The function takes arguments n for the number of draws to simulate, size for the number of trials (zero or more), and p for the success probability. It returns a vector of length n that contains independent draws from the Binomial. For example, we can simulate 10 independent random draws from a Binomial with size 5 and probability of 0.5 with,

```
rbinom(n = 10, size = 5, prob = 0.5)
```

```
##  [1] 3 2 2 4 3 2 1 1 4 3
```

Each element in the vector is the number of "successes" in 5 trials. Note that the Bernoulli is equivalent to a Binomial with $n = 1$, i.e. $\text{Bin}(1, p)$. So the Bernoulli is a special case of the Binomial. There isn't a function to simulate Bernoulli's in R but we take advantage of this fact and just use rbinom(). For example, we can simulate a sequence of 10 independent Bernoulli trials, each with success probability $p = 0.30$, in R via

```
rbinom(n = 10, size = 1, prob = 0.3)
```

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

So if we want to simulate Bernoulli trials we just set size equal to 1. This is almost always what we want to do in practice.

Standard Normal distribution

A continuous r.v. Z has a standard Normal distribution if its PDF φ is

$$\varphi(z) = \frac{1}{\sqrt{2\pi}} e^{-z^2/2}$$

for $z \in \mathbb{R}$. This is typically written as $Z \sim \mathcal{N}(0, 1)$ since it has mean 0 and variance 1. The standard Normal CDF Φ is, per usual, the integral of the PDF, e.g.

$$\phi(z) = \int_{-\infty}^{z} \varphi(t) dt = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

gives us the accumulated area under the PDF from $-\infty$ to z, which we might denote $F_Z(z) = \Pr(Z \leq z)$. But the standard Normal is so important that it has its own special symbols, e.g. $\Phi(z) = \Pr(Z \leq z)$. We leave it as an integral because there is no closed form solution. For example, what's the probability of seeing a value less than or equal to 0 in the standard Normal? This is the integral from $-\infty$ to 0. Luckily R can handle this for us with the pnorm() function,

```
pnorm(0)
```

```
## [1] 0.5
```

As expected, we get 0.5 (Why?). Suppose we want to work with the PDF to determine the probability of observing z between -1 and 1 (remember the 68-95-99.7 rule?). We could write a function for this,

```
normal_pdf <- function(z) {
  1/sqrt(2*pi)*exp(-z^2/2)
}
```

And then use the integrate() function in R,

[3] Where does this come from? Well, n independent Bernoulli trials is a sequence of successes and failures (kind of like academic life, right?). The probability of any specific sequence of k successes and $n - k$ failures is $p^k(1-p)^{n-k}$. We have $\binom{n}{k}$ such sequences. Why? Well, we just need to select where the *successes* are.

```r
integrate(normal_pdf, -1, 1)
```

```
## 0.6826895 with absolute error < 7.6e-15
```

But we don’t need to do this, because we have the `dnorm()` function,

```r
integrate(dnorm, -1, 1)
```

```
## 0.6826895 with absolute error < 7.6e-15
```

The general **Normal Distribution**, with parameters μ and σ^2, has an important relationship to the standard Normal. If $Z \sim \mathcal{N}(0, 1)$ then

$$X = \mu + \sigma Z$$

, which we write $X \sim \mathcal{N}(\mu, \sigma^2)$. If we start with the general Normal, we can also convert the other way or “standardize”. The standardized verison of $X \sim \mathcal{N}(\mu, \sigma^2)$ is

$$\frac{X - \mu}{\sigma} \sim \mathcal{N}(0, 1)$$

The Normal CDF is

$$F(x) = \Phi\left(\frac{x - \mu}{\sigma}\right)$$

Why? Because

$$F(x) = \Pr(X \leq x) = \Pr\left(\frac{X - \mu}{\sigma} \leq \frac{x - \mu}{\sigma}\right) = \Phi\left(\frac{x - \mu}{\sigma}\right)$$

And the Normal PDF is

$$f(x) = \varphi\left(\frac{x - \mu}{\sigma}\right)\frac{1}{\sigma}$$

Why? Because[4]

$$f(x) = \frac{d}{dx}\Phi\left(\frac{x - \mu}{\sigma}\right) = \varphi\left(\frac{x - \mu}{\sigma}\right)\frac{1}{\sigma} = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Suppose we want to generate 10,000 random deviates from the Normal with $\mu = 2,\ \sigma^2 = 4$. We can do this in two ways. First, we could use the fact from above about the relationship between the standard Normal and the general Normal. Second, we can specify the mean and variance directly in `R` via the `rnorm()` function.

```r
set.seed(123)

## Method 1
Z <- rnorm(10000)
X1 <- 2 + sqrt(4)*Z

## Method 2
X2 <- rnorm(n = 10000, mean = 2, sd = sqrt(4))

mean(X1)
```

```
## [1] 1.995257
```

```r
mean(X2)
```

```
## [1] 1.981787
```

```r
var(X1)
```

```
## [1] 3.989101
```

[4] remember the chain rule? $D\{f(g(x))\} = f'(g(x))g'(x)$

```
var(X2)
```

```
## [1] 4.012549
```

We can convince ourselves that the mean and variance are the same with math,

$$\mathbb{E}[\mu + \sigma Z] = \mathbb{E}[\mu] + \sigma \mathbb{E}[Z] = \mu$$
$$\text{Var}\left(\mu + \sigma Z\right) = \text{Var}(\sigma Z) = \sigma^2 \text{Var}(Z) = \sigma^2$$

Suppose $X \sim \mathcal{N}(-2, 4)$. What is $\Pr\left(|X| < 4\right)$? We can epxress this in terms of the standard Normal: $Z = \frac{X-(-2)}{2}$. Then,

$$\Pr\left(-4 < X < 4\right) = \Pr\left(\frac{-4-(-2)}{2} < \frac{X-(-2)}{2} < \frac{4-(-2)}{2}\right) = \Pr\left(-1 < Z < 3\right) = \Phi(3) - \Phi(-1)$$

Which is easy to calculate in R via the standard Normal CDF,

```
pnorm(3) - pnorm(-1)
```

```
## [1] 0.8399948
```

Which, as expected, gives the same result as,

```
pnorm(4, mean = -2, sd = 2) - pnorm(-4, mean = -2, sd = 2)
```

```
## [1] 0.8399948
```

Generating random deviates with inverse transform

Almost all programming languages are capable of generating pseudo random samples from the uniform distribution. With some statistical magic (a technique called "inverse transform sampling", which relies on the "universality of the uniform"), we can convert the uniform random variables to any other desired r.v., given its cumulative distribution function. All we need is the ability to generate draws from the unifrom distribution.

Let F_X denote the CDF for some arbitrary distribution that we want to sample from. Note that $F_X(x) = \Pr(X \leq x) = u$. The inverse of the CDF, F_X^{-1}, takes u as input and returns x. When the u's are uniformly distributed then we can sample from any F_X if we can write down F_X^{-1}. It is easy to sample from $u \sim \text{Unif}(0, 1)$, so we just take the values of u and feed them to F_X^{-1} to produce the x's since $F_X^{-1}(u) = x$.

Inversion sampling is the default procedure at work "behind the scenes" for many (all?) of the random number generators that come with base R, including the Normal. And we would never bother coding this method by hand to sample from a distribution like the Normal precisely because we can take advantage of `rnorm()`. However, this is potentially useful when we want to simulate from a distribution that doesn't live in base R, like the Laplace distribution example below.

Illustration: Biased coin flip

Suppose we are taking a statistics class and our TA asks us to simulate a coin tossing process where $p \neq 0.5$ is the probability of heads. Our TA is a sadist and says the only base R function we can use is `runif()`. Let $U \sim \text{Unif}(0, 1)$. We can write the Bernoulli r.v. X as,

$$X = \begin{cases} 1 & \text{if } U < p \\ 0 & \text{if } U \geq p \end{cases}$$

Let H denote "success" – the coin comes up heads. It follows that

$$\Pr(H) = \Pr(X = 1) = \Pr(U < p) = p$$

So $X \sim \text{Bern}(p)$. Suppose we want to simulate deviates for a biased coin with $p = 0.4$. This is easy to do in R[5].

```
set.seed(123)
p <- 0.4
U <- runif(1, min = 0, max = 1)
X <- as.numeric(U < p)
```

What's going on here? The support of U is $[0, 1]$. The function `runif()` returns a number from this interval. In this case $U =$`0.2875775`. In essence, we divide the interval $[0, 1]$ into two parts: one of length $p = 0.4$ and another of length $1 - p = 0.6$ in this example. The value of X, our r.v., is determined based on the realization of U, here `0.2875775`. Let's write a function for this,

```
gen_bernoulli <- function(p, nsims){
  X <- NULL
  for(i in 1:nsims){
    U <- runif(1, min = 0, max = 1)
    X[i] <- as.numeric(U < p)
  }
  return(X)
}
```

Now let's confirm it works by taking 10,000 draws from it with $p = 0.4$ and confirming that we get 'heads' 4 out of 10 times.

```
set.seed(123)
x <- gen_bernoulli(p = 0.4, nsims = 10000)
table(x)/10000
```

```
## x
##      0      1
## 0.6026 0.3974
```

Illustration: Laplace distribution

The double exponential (Laplace) distribution

$$f(x) = \frac{a}{2} \exp\{-a|x - b|\}$$

where $a > 0$ and b is any finite real number. If we let $a = a^{-1}$ then the pdf is the standard representation of the Laplace,

$$g(x) = \frac{1}{2a} \exp\left(-\frac{|x - b|}{a}\right)$$

[5] note that R does not have a built in function called 'rbern()' for generating random Bernoulli r.v.s; instead, we use the Binomial with $size = 1$. This can be also be accomplished with the function 'rbinom()'

Which has cdf,

$$G(x) = \begin{cases} \frac{1}{2}\exp\left(\frac{x-b}{a}\right) & x \in (-\infty, b]; \\ 1 - \frac{1}{2}\exp\left(-\frac{x-b}{a}\right) & x \in [a, \infty). \end{cases}$$

Substituting $a = a^{-1}$ we have,

$$F(x) = \begin{cases} \frac{1}{2}\exp\left(a(x-b)\right) & x \in (-\infty, b]; \\ 1 - \frac{1}{2}\exp\left(-a(x-b)\right) & x \in [b, \infty). \end{cases}$$

Solving each piece of $U = F(X)$ for X yields,

$$\begin{aligned} U &= \frac{1}{2}\exp\left(a(X-b)\right) \\ \log(2U) &= a(X-b) \\ X &= \frac{\log(2U)}{a} + b \\ &\text{and} \ldots \end{aligned}$$

$$\begin{aligned} U &= 1 - \frac{1}{2}\exp\left(-a(x-b)\right) \\ 2(1-U) &= \exp\left(-a(x-b)\right) \\ X &= b - \frac{1}{a}\left(\log\left(2(1-U)\right)\right) \end{aligned}$$

Therefore, for $a \in (0, \infty)$ and $b \in \mathbb{R}$

$$X = \left[b + \frac{1}{a}\log(2U)\right]\mathbb{1}\left(U < \frac{1}{2}\right) + \left[b - \frac{1}{a}\log\left(2U\right)\right]\mathbb{1}\left(U \geq \frac{1}{2}\right)$$

Has the double exponential (Laplace) distribution. The algorithm to generate a random draw is then,

1) Generate $U \sim \text{Unif}(0, 1)$
2) Set $X = b + \frac{1}{a}\log(2U)$ if $U < \frac{1}{2}$. Else $X = b - \frac{1}{a}\log\left(2(1-U)\right)$.

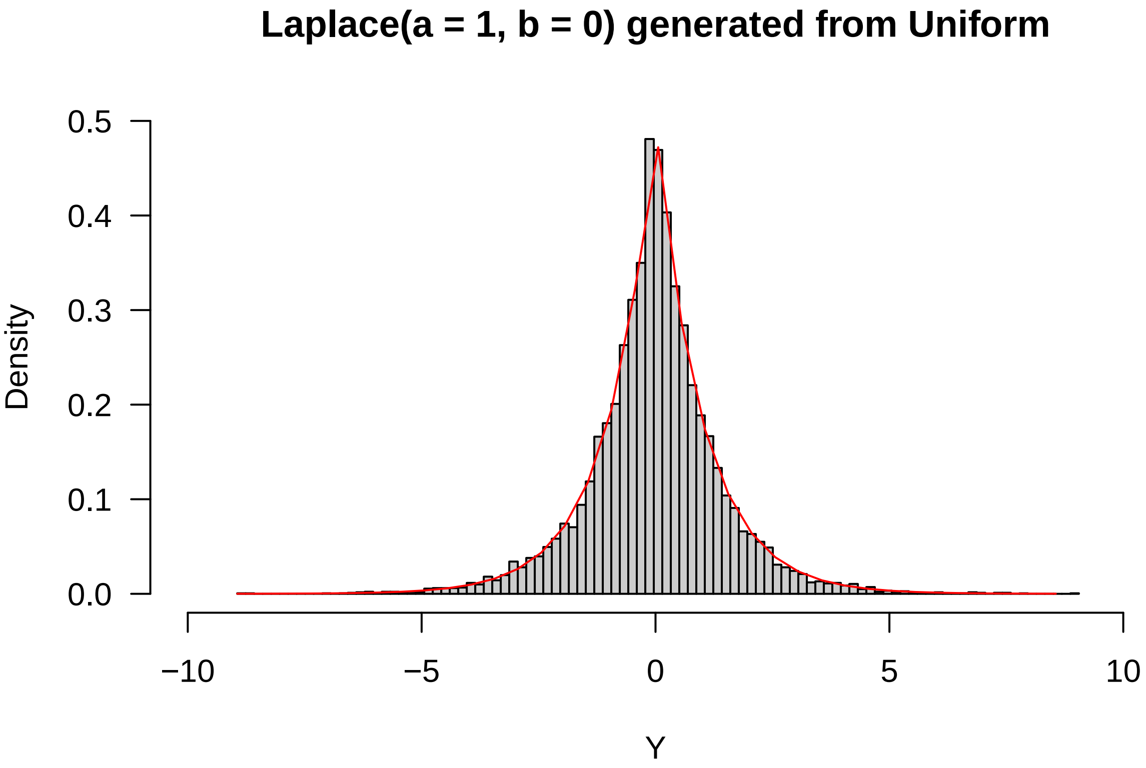
First, I create some functions:

```r
genlaplace <- function(a, b, nsims){
  X <- NULL
  for(i in 1:nsims){
    U <- runif(1)
    if(U < 0.5){
      X[i] <- (b + (1/a)*log(2*U))
    } else{
      X[i] <- (b - (1/a)*log(2*(1-U)))
    }
  }
  return(X)
}

laplace_pdf <- function(a, b, x){
  out <- NULL
  for(i in seq_along(x)){
    out[i] <- (a/2)*exp(-a*abs(x[i]-b))
  }
  return(out)
}
```

Now I use the function `genlaplace()` to generate 10,000 deviates from the Laplace distribution with $a = 1,\ b = 0$ and plot them using a histogram with the true Laplace density (using the `laplace_pdf()` function I created above) overlaid as a red line:

```
set.seed(123)
Y <- genlaplace(a=1, b = 0, nsims=10000)
supp <- seq(min(Y), max(Y), 0.5)
hist(Y, breaks = seq(min(Y), max(Y), length.out=100), probability = TRUE,
     col = "grey80", las = 1, xlim = c(-10, 10),
     ylim = c(0,0.5), main = "Laplace(a = 1, b = 0) generated from Uniform")
lines(supp, laplace_pdf(x=supp, a=1, b=0), col = "red")
```



More simulation applications

Law of Large Numbers

Suppose we have i.i.d. $X_1, X_2, X_3, \ldots$ with finite mean μ and finite variance σ^2. For all positive integers n, let

$$\bar{X}_n = \frac{X_1 + \cdots + X_n}{n}$$

denote the sample mean of X_1 through X_n. The sample mean is an r.v. with mean μ and variance σ^2/n. Convince yourself that this is true by calculating $\mathbb{E}[\bar{X}_n]$ and $\text{Var}(\bar{X}_n)$.

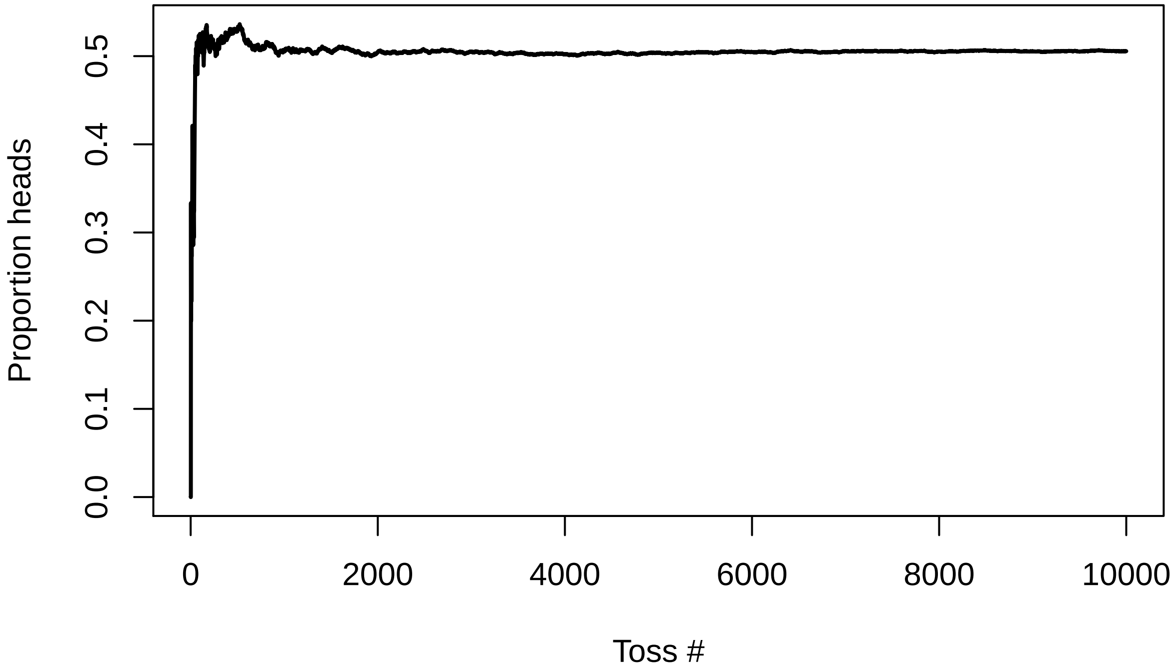
The law of large numbers (LLN) says that as n grows, the sample mean converges to the true mean μ. Formally, the (weak) LLN says: For all $\epsilon > 0$, $\Pr\left(|\bar{X}_n - \mu| > \epsilon\right) \to 0$ as $n \to \infty$. The relevant concept here is "convergence in probability". Another way to write this is $\bar{X}_n \xrightarrow{p} \mathbb{E}[X] = \mu$ where $\xrightarrow{p}$ means "converges in probability".

We could prove the (weak) LLN analytically [Hint: use the Chebyshev Inequality] but we can also simulate it. Let's simulate tossing a fair coin 10000 times to illustrate how the LLN works.

```r
set.seed(123)
n <- 10000
toss <- gen_bernoulli(p = 0.5, nsims = n)
a <- numeric(n + 1)
avg <- numeric(n)

for(i in 2:n + 1) {
  a[i] <- a[i-1] + toss[i-1]
  avg[i-1] <- a[i]/(i-1)
}

plot(1:n, avg[1:n], type = "l", lwd = 2, ylab = "Proportion heads", xlab = "Toss #")
```



Central Limit Theorem

The LLN tells us that as $n \to \infty$, $\bar{X}_n$ converges to the constant μ with probability 1. The central limit theorem (CLT) tells us what happens to its distribution along the way. The CLT states that for large n, the distribution of $\bar{X}_n$, after standardization, approaches the standard Normal distribution. Formally, for $\mathbb{E}[X] = \mu < \infty$ and $0 < \text{Var}(X) = \sigma^2 < \infty$,

$$\sqrt{n}\left(\frac{\bar{X}_n - \mu}{\sigma}\right) \xrightarrow{d} \mathcal{N}(0,1)$$

as $n \to \infty$. Where $\xrightarrow{d}$ means "converges in distribution". You can prove this if you want using moment generating functions, but it's hard. Instead, simulate it in R! Pick any distribution for X that you like (except the Normal!) that satisfies the necessary regularity conditions for the mean and variance.

Exercise: Write a function called `sim_clt()` that takes arguments `X`, `r` and `n` where `X` is a vector of random deviates from your chosen distribution, `r` is the number of repititions (use atleast 10,000), and `n` is the number of samples in each sample set (use atleast 100). The output of this function should be a vector of length `r` that contains the sample means for each repitition. Your function should also standardize by subtracting μ, the expected value of $\bar{X}_n$, and dividing by $\sigma/\sqrt{n}$, the standard deviation of $\bar{X}_n$. Store the result in a vector called `Z`.

```
## this function computes the r samples of the sample mean from the
## r*n original samples
sim_clt <- function(X, r, n) {
  sample_means <- rowMeans(matrix(X, nrow=r, ncol=n))
  Z <- (sample_means - mean(X))/(sd(X)/sqrt(n))
  return(Z)
}
```

Now verify that it works! First try the standard uniform distribution,

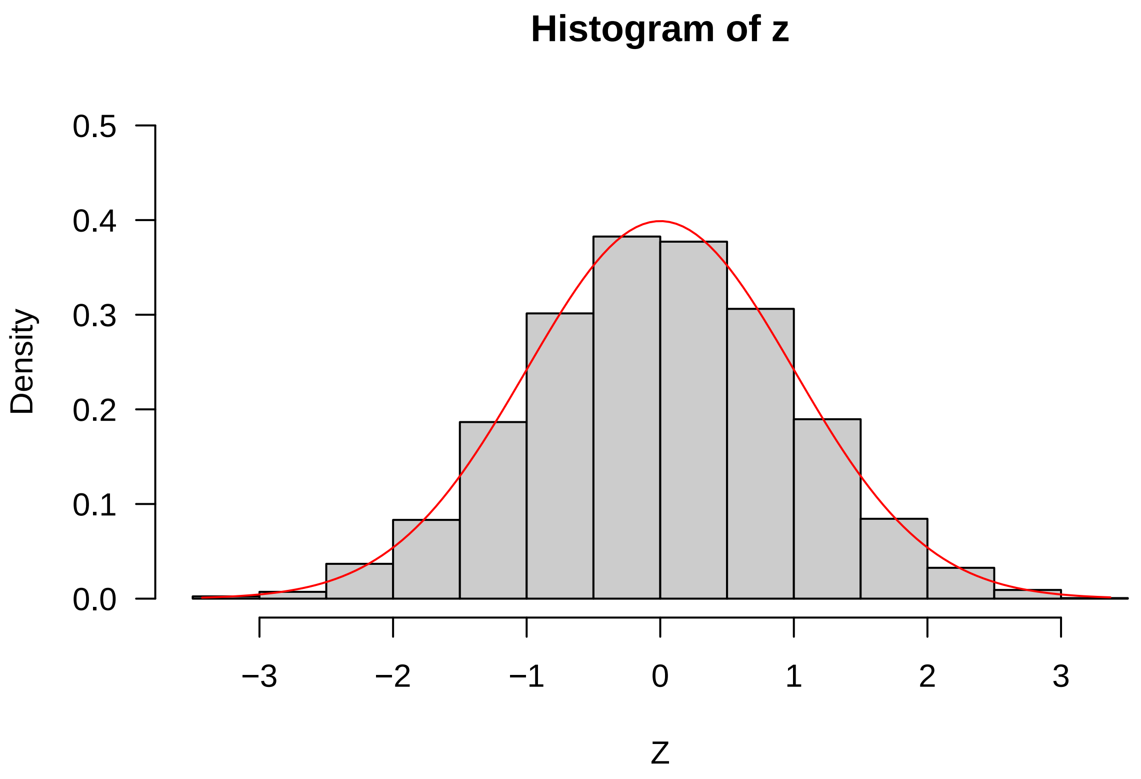
```
set.seed(123)
r <- 10000
n <- 200
Z <- sim_clt(X = runif(r*n), r = r, n = n)
mean(Z)
```

```
## [1] 1.962006e-16
```

```
sd(Z)
```

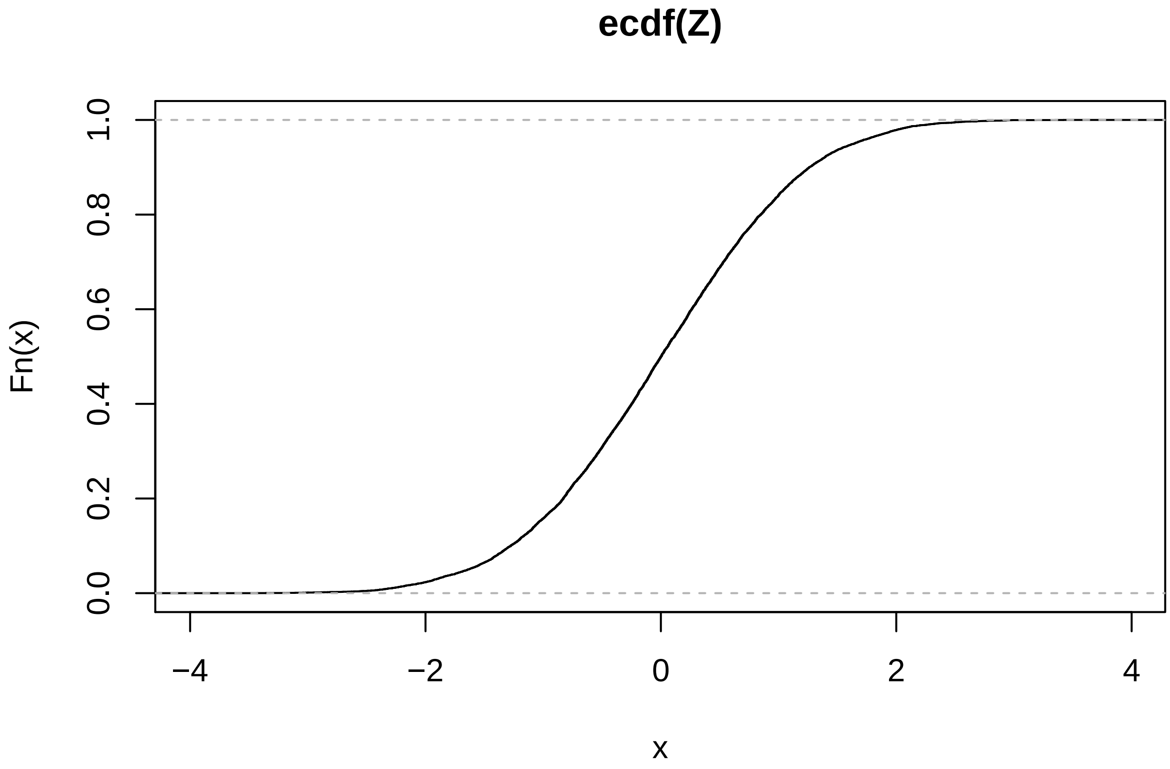
```
## [1] 0.9914193
```

```
supp <- seq(min(Z), max(Z), 0.05)
hist(Z, probability = TRUE, col = "grey80", las = 1, ylim = c(0, 0.5),
     main = "Histogram of z")
lines(supp, dnorm(x=supp), col = "red")
```



We can also plot the empirical CDF and see if it looks like it should,

```r
plot(ecdf(Z))
```



Another way to check if this worked (besides visual checking of implications) is to use the Kolmogorov-Smirnov test implemented with the `ks.test()` function in `R`. The null hypothesis here is that Z comes from a standard

Normal. We can use this for a variety of distributions besides the standard Normal (see `?ks.test()` for details),

```
ks.test(Z, "pnorm")
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  Z
## D = 0.0047157, p-value = 0.9793
## alternative hypothesis: two-sided
```