Continuous RVs and University of Uniform

Becky Tang

3/31/25

Continuous RVs in `R`

Built into `R` are some functions that work with the named continuous distributions. Recall we have the following types of functions:

For example:

- `dnorm(x, mu, sigma)` will evaluate the PDF of a $N(\mu, \sigma^2)$ distribution at the value `x`.
- `pnorm(q, mu, sigma)` will evaluate the CDF of a $N(\mu, \sigma^2)$ distribution at the value `q`
- `qnorm(p, mu, sigma)` will evaluate the inverse-CDF of a $N(\mu, \sigma^2)$ distribution at the value `p`
- `rnorm(n, mu, sigma)` will generate `n` random variables from the $N(\mu, \sigma^2)$ distribution

NOTE: for the `_norm()` functions in R, the functions expect standard deviation as input, not variance!!

```r
# X ~ N(0, 4) -> What is P(X <= 1)?
pnorm(1, 0, sqrt(4))
```

```
[1] 0.6914625
```

The different distributions in R all follow the same format: `d<dist>()` or `p<dist>()`, and you specify the specific inputs and parameters.

```r
# Generate 10 random variables from the Unif(0,1) distribution:
runif(10, min = 0, max = 1)
```

```
 [1] 0.58337852 0.19442343 0.47478946 0.17898227 0.94950207 0.79544064
 [7] 0.16657604 0.45360168 0.05066744 0.46992893
```

```r
# X ~ Exp(2). What is f(1)?
dexp(2)
```

```
[1] 0.1353353
```

```r
# Obtain median of standard normal
qnorm(0.5, 0, 1)
```

```
[1] 0
```

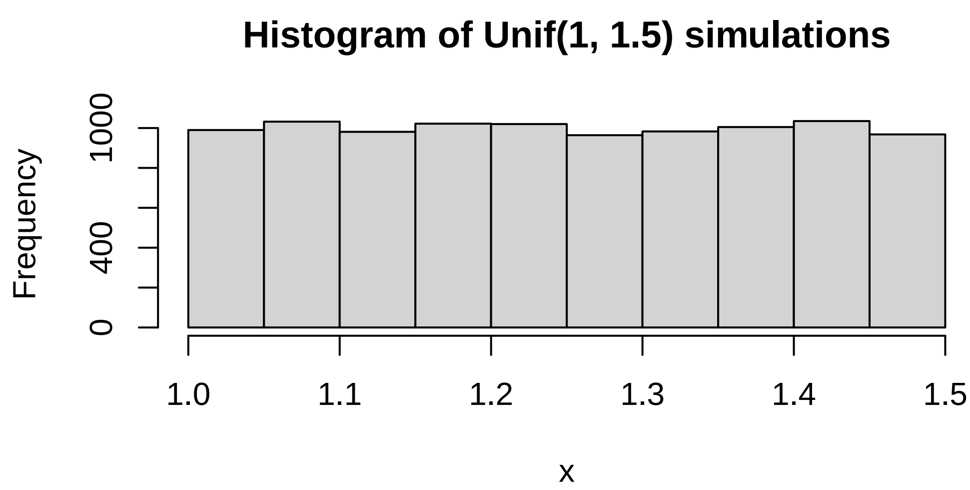
Visualizing densities

Sometimes it can be helpful to visualize the density of a distribution. There are a couple ways we can do this. Let's do this example for the $Unif(1, 1.5)$ distribution.

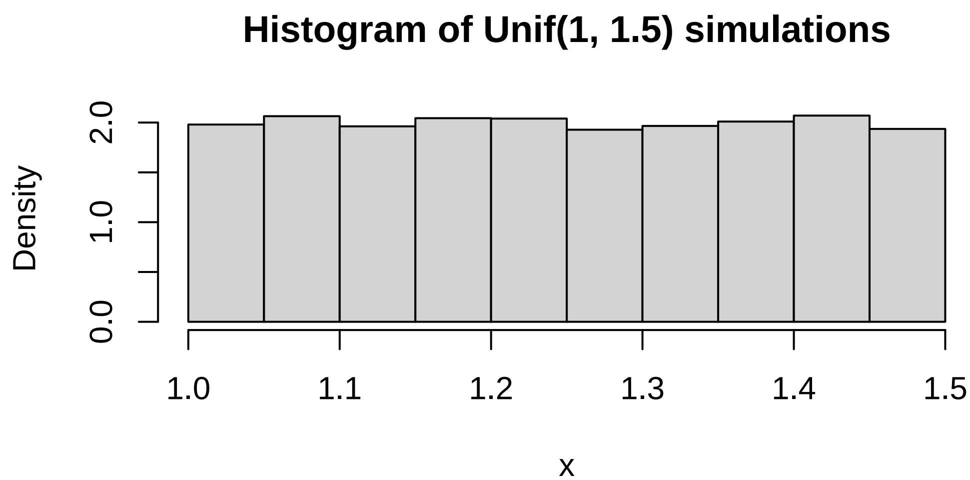
Option 1

If we can randomly sample from the distribution, we can generate lots and lots of random variables from that distribution and make a histogram of them!

```r
# simulate lots and lots of Unif(1, 1.5) rvs
sims <- runif(10000, min = 1, max = 1.5)

# make a histogram of the simulations
hist(x= sims, xlab = "x", main = "Histogram of Unif(1, 1.5) simulations")
```



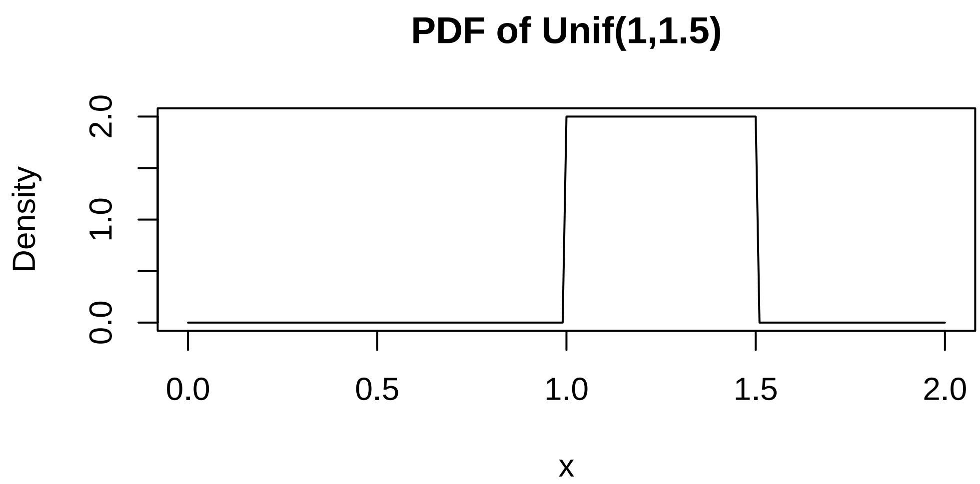
```
# turn y-axis into a density
hist(x = sims,  xlab = "x", main = "Histogram of Unif(1, 1.5) simulations",
     freq = F)
```



Option 2

If we have access to the PDF directly (which we do for the named distributions), we can simply graph the function

```
# create a sequence of values from 0 and 2 at equally-size 0.01 increments
x_seq <- seq(0, 2, 0.01)

# evaluate PDF of interest at each value in x_seq
f <- dunif(x_seq, min = 1, max = 1.5)

# type = "l" turns into lines
plot(x = x_seq, y = f, xlab = "x", ylab = "Density", main = "PDF of Unif(1,1.5)",
     type = "l")
```



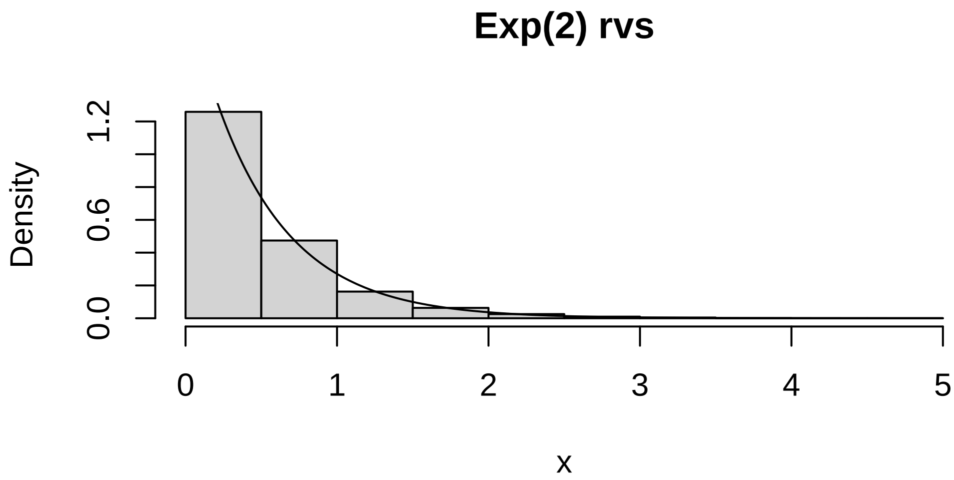
Universality of Uniform / Probability Integral Transform

Now, let's see the University of the Uniform in action! Suppose you've lost access to all the functions in `R` that allow you randomly generate rvs from all the named distribution *except* for the Uniform.

Example 1

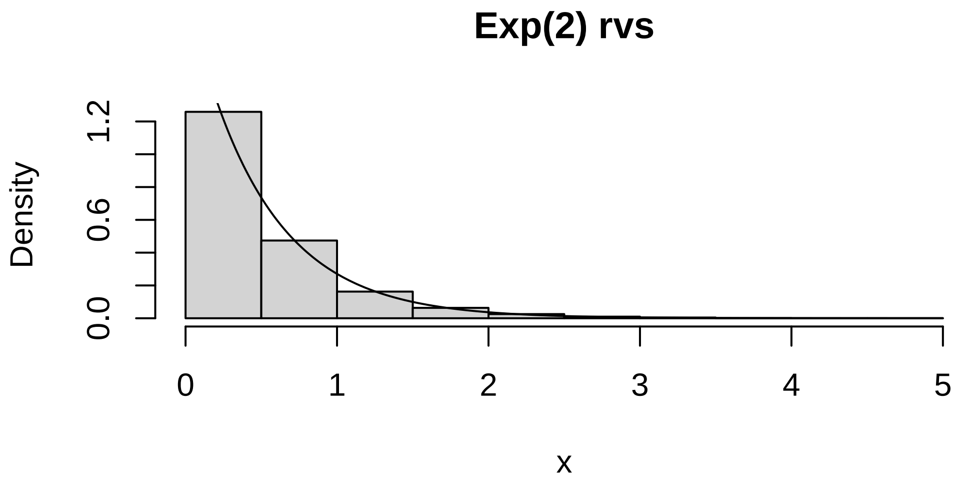
How can we simulate values from the Exp(2) distribution?

```
# generate lots of Unif(0,1) rvs
u <- runif(10000, min = 0, max = 1)

# use inverse CDF that we derived
lambda <- 2
x <- (-1/lambda) * log(1 - u)

# let's visualize them:
hist(x, xlab = "x", main = "Exp(2) rvs", freq = F)

# let's add the following to to double check
x_seq <- seq(0, 5, 0.01)
f <- dexp(x_seq, rate = lambda)
lines(x_seq, f, type = "l")
```



What if we didn't have access to the density function in R? No problem! We can make our own:

```
my_dexp <- function(x, lambda){
  n <- length(x)
  ret <- rep(0, n)
  for(i in 1:n){
      if(x[i] > 0){
        ret[i] <- x[i] * exp(-lambda * x[i])
      }
  }
}
f2 <- my_dexp(x_seq, lambda)
hist(x, xlab = "x", main = "Exp(2) rvs", freq = F)
lines(x_seq, f, type = "l")
```



Looks the same!

Example 2

Suppose we have a distribution whose PDF is

$$f_X(x) = \frac{e^{-x}}{(1+e^{-x})^2}$$

for $x \in (-\infty, \infty)$.

Write code to simulate 1000 random variables from this distribution, and visualize them as a density histogram.