Random Variables

Random Variables in R

R has built in functions for working with several common distributions:

- Normal
- Uniform
- Gamma
- Binomial
- Chi-square
- Poisson
- Multinomial

For a more complete list, type `help(Distributions)`.

Random Variables in R

For every one of these distributions, R has at least 4 functions that

- Compute the density
- Compute the CDF
- Compute the inverse CDF
- Simulate a random variable

Normal Random Variables

- Compute the density `dnorm`
- Compute the CDF `pnorm`
- Compute the inverse CDF `qnorm`
- Simulate a random variable `rnorm`

General Random Variables

Different distributions have a corresponding suffix, e.g. `norm` for normal, and their density, CDF, inverse CDF, and random variable generation functions can be obtained by combining the appropriate letter - `d`, `p`, `q`, or `r` - with the corresponding suffix.

Early Random Number Generation

- Coins, dice, spinning a cardboard disc
- Tables/books of random numbers based on
  - Experimental data combined with statistical theory
  - Randomly selecting numbers from tables/books of other numbers

Even when these methods work well, they take a lot of effort and require importing sequences of random numbers onto a computer.

Deterministic Random Number Generation

- Middle Square Method (MSM):
  - Start with a number x with $2a$ digits and square it to get a number with $4a$ digits (pad with 0's if needed)
  - Retain the middle $2a$ digits of the square
  - Iterate
- Digits of transcendental numbers

MSM method can yield sequences that get stuck at 0 or have a short "cycle," try starting from 2500 or starting from any two digit number.

Digits of transcendental numbers are too expensive to obtain and store and hard to study mathematically.

What does R use?

You can read about what R uses and other options if you type `help(RNG)`.

R uses the "Mersenne-Twister" from 1998!

All of the options start from a specified seed, which is a number that you provide or the computer picks for you based on features of the session.

You need to fix the seed to get replicable results, using `set.seed`.

```
set.seed(372132)
```

All Roads Lead Back to the Uniform

- A sequence of positive numbers can be transformed to a distribution on $(0, 1)$ by adding a decimal point
- Any random variable can be obtained using uniform variables on $(0, 1)$
- By definition, applying a CDF to identically distributed random variables yields uniform variables on $(0, 1)$ - we can reverse this!

Normal Random Variables from Uniform

```r
u <- runif(100000)
```

```r
z <- qnorm(u)
```

```r
hist(z, freq = FALSE)
curve(dnorm, add = TRUE)
```

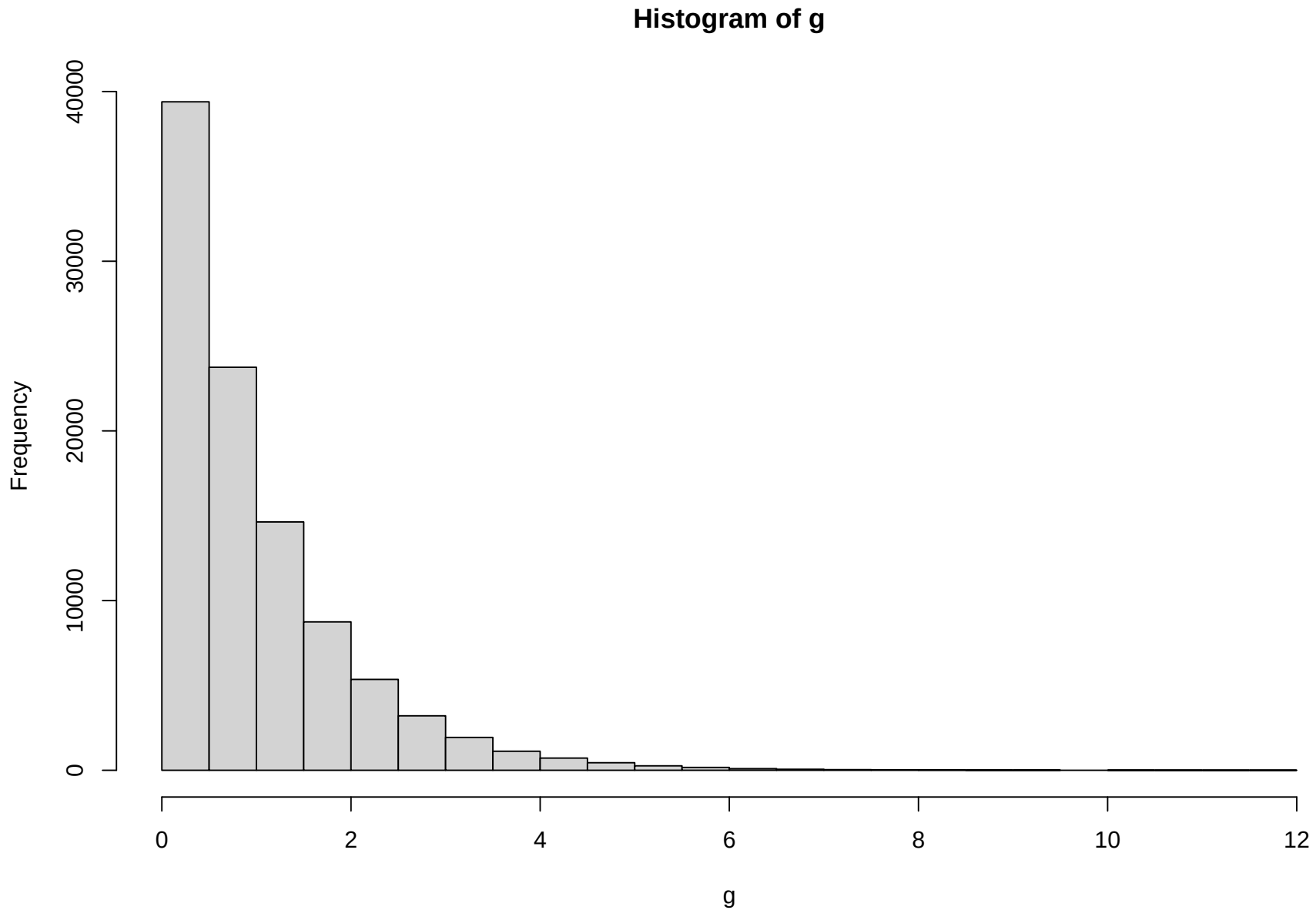
Normal Random Variables from Uniform

Normal Random Variables from Uniform

We generated standard normal random variables...how do we generate normal random variables with mean μ and variance σ^2?

Gamma Variables from a Uniform

```
g <- qgamma(u, 1, 1)
```



Gamma Variables from a Uniform

```
g <- qgamma(u, 0.0001, 1)
```

```
table(g == 0)
```

```
FALSE  TRUE
 7144 92856
```

Sometimes the inversion method for simulating from a distribution can be numerically unstable.