Project 04

STAT 41600

On Simulating Random Variables

Recall the following Theorem

Theorem: *Suppose that a continuous RV X has a CDF $F(x)$, then*

$$U \equiv F(X) \sim U(0,1) \qquad \text{and in particular} \qquad X = F^{-1}(U) \sim F(\cdot).$$

we will use it to simulate (generate) continuous type random variables.

The Exponential distribution $\mathcal{E}xp(\beta)$

- For an exponential distribution with rate $\lambda \equiv 1/\beta$, the *pdf* and *cdf* are, for $x > 0$,

$$f(x) = \lambda e^{-\lambda x} \quad \text{and} \quad F(x) = 1 - e^{-\lambda x}.$$

- It is easy to check that the inverse cdf is

$$F^{-1}(u) = -\log(1-u)/\lambda, \quad u \in (0,1).$$

- Therefore, to sample X from an exponential distribution:
  1. Sample $U \sim \mathsf{Unif}(0,1)$.
  2. Set $X = -\log(1-U)/\lambda$.
- This can be easily "vectorized" to get samples of size n.
- The built-in R function is `rexp`.

Example: Generate 100 ranom values from the $\mathcal{E}xp(5)$ distribution

```
n<-100
lambda<-0.2

xx<--log(1-runif(n))/lambda ## Note: this is the same as xx<-replicate(n, -log(1-runif(1))/lambda)

summary(xx)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##  0.03528  1.37466  3.11084  4.63177  6.51141 25.34439
```

To simulate the $\mathcal{E}xp(5)$ distribution

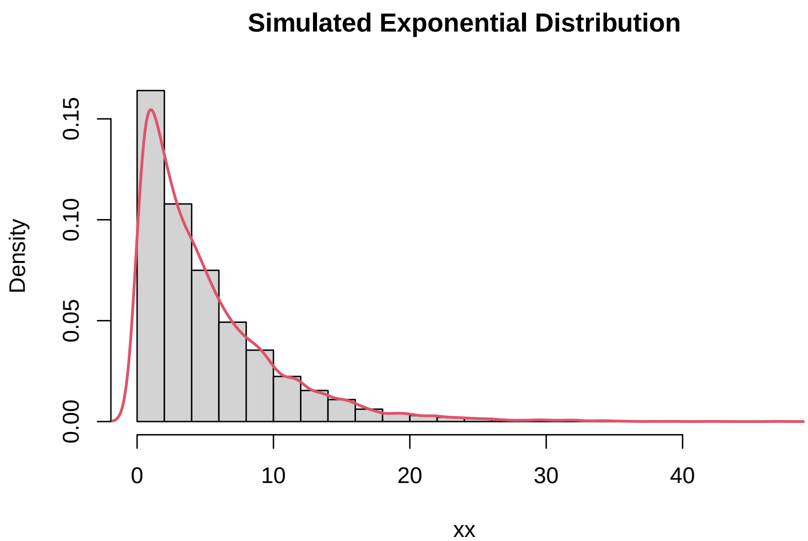
```r
n<-10000
lambda<-0.2

xx<--log(1-runif(n))/lambda

hist(xx, freq=FALSE, nclass=30, main='Simulated Exponential Distribution')

#Standard Density Estimate
est.den<-density(xx)
x0<-est.den$x
y0<-est.den$y

lines(x0, y0, lwd=2, col=2)
```



The Gamma distribution, $\mathcal{G}amma(k, \beta)$

We can use the above procedure to simulate the Gamma distribution, in the special case of $\alpha \equiv k$, an integer. This procedure is based on the fact that if $X_i \sim \mathcal{E}xp(\beta)$, $i = 1, \ldots, k$ and independent, than

$$Y \equiv X_1 + X_2 + \ldots, +X_k \sim \mathcal{G}amma(k, \beta).$$

- The built-in R function is `rgamma`.

For example, simulating the $\mathcal{G}amma(3, 5)$ distribution

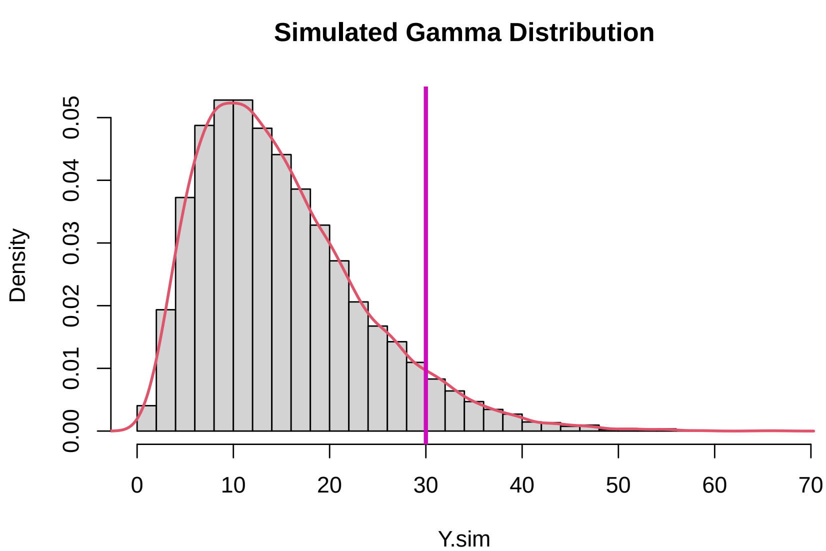
```
n<-10000
k<-3
lambda<-0.2

Y.sim<- replicate(n, -sum(log(1-runif(k))/lambda))

hist(Y.sim, freq=FALSE, nclass=30, main='Simulated Gamma Distribution')

#Standard Density Estimate
est.den<-density(Y.sim)
x0<-est.den$x
y0<-est.den$y

lines(x0, y0, lwd=2, col=2)
abline(v=30, col=6, lwd=3)
```



Approximated *cdf* and probabilities

- We may use the simulated data to approximate the *cdf* and probabilities (recall Project 01). For example, in the case of $Y \sim \mathcal{Gamma}(3,5)$, we can approximate

$$F_Y(30) \equiv Pr(Y \leq 30)$$

by

$$\hat{F}_Y(30) \approx \frac{\#\text{ of times } Y_i \leq 30}{n} \equiv \frac{1}{n}\sum_{i=1}^{n} I[Y_i \leq 30]$$

For the above data

```
Pr30<-sum(Y.sim<=30)/n
Pr30
```

```
## [1] 0.9371
```

```
# Compare this approximate probability with the one calculated using the built-in  R function
pgamma(30, 3, 0.2)
```

```
## [1] 0.9380312
```

The Standard Normal r.v.

- While normal RVs can, in principle, be generating using the *cdf* transform method, this requires evaluation of the standard normal inverse *cdf*, which is a non-trivial calculation.
- There are a number of fast and efficient alternatives for generating normal RVs.
- The one below, due to Box and Muller, is based on some trigonometric transformations.
- R has a number of algorithms to choose from – one is the Box-Muller method, although the default choice is the probability transform.
- The built-in R function is `rnorm`.

The Box-Muller method

- This method generates a pair of two independent normal RVs X and Y.
- The method is based on the following facts:
  - The cartesian coordinates (X, Y) are equivalent to the polar coordinates (Θ, R), and the polar coordinates have a *joint pdf*

$$r\exp\{-r^2/2\}/2\pi, \quad (\theta, r) \in [0, 2\pi] \times [0, \infty).$$

  - Then $\Theta \sim \mathsf{Unif}(0, 2\pi)$ and $R^2 \sim \mathsf{Exp}(2)$ are independent.
- So to generate independent normal X and Y:
  1. Sample $U, V \sim \mathsf{Unif}(0,1)$.
  2. Set $R^2 = -2\log(1-V)$ and $\Theta = 2\pi U$.
  3. Finally, take $X = R\cos\Theta$ and $Y = R\sin\Theta$.
- Take a linear function to get a normal r.v. with different mean and variance. For example

$$\mu + \sigma \cdot X \sim \mathcal{N}(\mu, \sigma^2)$$

Code to implement the Box-Muller method for normal RV generation

```
rnorm.bm <- function(n=1) {
  U <- runif(n)
  V <- runif(n)

  R <- sqrt(-2 * log(1-V))
  Theta <- 2 * pi * U

  X <- R * cos(Theta)
  Y <- R * sin(Theta)

  return(cbind(X,Y)) # or use return(X) to toss out the Y and to keep only the X
}
```

Task:

- Use the above procedures to generate $n = 10000$ random values from the $\mathcal{N}(10, 9)$ distribution.
- Obtain the summary statistics of these data. Do they conform with the theoretical distribution you meant to use?
- What proportion of your simulated data you **expect** to fall within two σ units from the mean?
- What proportion of your simulated data you actually **found** within two σ units from the mean?
- Use these simulated values to draw the approximate graph of the *pdf* of this distribution, appropriately labeled and marked.