GSI: Adam Rothman
Email: ajrothma@umich.edu
Office Hours: Monday 12-1 (441 WH), Thursday 11:30-1:30 (443 WH)

Course web page:

`http://www.stat.lsa.umich.edu/~kshedden/Courses/Stat406/`

Lab web page:

`http://www.stat.lsa.umich.edu/~ajrothma/Stat406/`

Simulated vs Exact Probabilities

Let Y be a standard Exponential random variable. Y is continuous with probability density function (pdf) $f_Y(y)$:

$$f_Y(y) = \begin{cases} e^{-y} & \text{if } y \geq 0 \\ 0 & \text{if } y < 0 \end{cases}$$

We can find the corresponding cumulative distribution function (cdf) $P(Y \leq t)$ by integrating the pdf:

$$\begin{aligned} P(Y \leq t) &= \int_{-\infty}^{t} f_Y(y)dy = \int_0^t e^{-y}dy = -e^{-y}|_{y=0}^{y=t} \\ &= 1 - e^{-t} \end{aligned}$$

If we wanted to find $P(a \leq Y < b)$, we can use the fact that for continuous random variables, $P(Y \leq t) = P(Y < t)$ and:

$$P(a \leq Y < b) = P(Y \leq b) - P(Y \leq a)$$

1. Use R to calculate $P(0.5 \leq Y < 1)$ exactly.

   Solution:

   ```
   a = 0.5
   b = 1
   the.prob  = pexp(b)-pexp(a)
   the.same.prob = 1-exp(-b) - (1-exp(-a))
   ```

2. Use R to simulate the value of $P(0.5 \leq Y < 1)$

   Solution:
   We must generate a large number of realizations(or draws) from a standard exponential distribution, then find the proportion of these realizations that are between 0.5 and 1.

(a) To start, generate `reps` realizations (or draws) from a standard Exponential distribution:

```
reps = 1e5
y.vec = rexp(reps)
```

(b) Count the number of realizations in `y.vec` that are between 0.5 and 1

```
## The Slow Way:
count = 0
for ( i in 1:reps )
{
  if( (a <= y.vec[i] ) & (y.vec[i] < b) )
  {
    count = count + 1
  }
}
count

## The Fast Way:
count = sum( (a <= y.vec ) & (y.vec < b) )
count
```

(c) Now divide this count by the number of replications to simulate the value of $P(0.5 \leq Y \leq 1)$

```
sim.prob = count / reps
sim.prob
```

The entire simulation problem can be done in two lines:

```
y.vec = rexp(1e5)
sim.prob = mean( (0.5 <= y.vec ) & (y.vec < 1) )
```

3. Building on the previous problem, fill in the following table:

	Exact	Simulation estimate
P$(0 \leq Y < 0.5)$		
P$(0.5 \leq Y < 1)$		
P$(1 \leq Y < 1.5)$		
P$(1.5 \leq Y < 2)$		

Solution:

```
## Start Vectors to store exact/simulated probs
exact.vec = NULL
sim.vec = NULL

## Generate 1e5 realizations from a standard Exponential distribution
y.vec = rexp(1e5)

## Create a vector of the left-cut points for the probs
a.vec = seq(0, 1.5, by=0.5)

for ( k in 1:length(a.vec) )
{
  a = a.vec[k]
  b = a + 0.5
  exact.vec[k] = pexp(b)-pexp(a)
  sim.vec[k] = mean( (a <= y.vec ) & (y.vec < b) )

  ## Print the result
  cat("P(", a, "<= Y <", b,")\t", exact.vec[k],
      "\t", sim.vec[k],  "\n", sep="")
}
```

4. Someone claims that if U is a standard uniform random variable ($U \sim \text{Uniform}(0, 1)$), then:

$$Y = \log(1/U)$$

will have a standard Exponential distribution. Using simulation, fill in the table from the last problem using this definition of Y, and assess whether it appears to behave like a standard Exponential random variable.

```
## Start Vectors to store exact/simulated probs
exact.vec = NULL
sim.vec = NULL

## Generate 1e5 realizations from a standard Uniform distribution
u.vec = runif(1e5)

## Make y.vec from the u.vec
y.vec = log(1/u.vec)

## Create a vector of the left-cut points for the probs
a.vec = seq(0, 1.5, by=0.5)

for ( k in 1:length(a.vec) )
```

```
{
  a = a.vec[k]
  b = a + 0.5
  exact.vec[k] = pexp(b)-pexp(a)
  sim.vec[k] = mean( (a <= y.vec ) & (y.vec < b) )

  ## Print the result
  cat("P(", a, "<= Y <", b,")\t", exact.vec[k],
      "\t", sim.vec[k],  "\n", sep="")
}
```

It turns out that her claim is true, and we can verify it mathematically by looking at the cdf:

$$\begin{aligned} P(\log(1/U) \leq t) &= P(e^{\log(1/U)} \leq e^t) = P(1/U \leq e^t) \\ &= P(U \geq e^{-t}) = 1 - P(U \leq e^{-t}) = 1 - e^{-t} \end{aligned}$$

which is the cdf of the standard Exponential.

Coin Flipping

One can simulate a fair coin flip in R using the uniform distribution.

$$F = \begin{cases} 1 & \text{if } U \geq 1/2 \\ 0 & \text{if } U < 1/2 \end{cases}$$

where $U \sim \text{Uniform}(0,1)$, and F is our coin flip random variable, where we say that 1=Heads and 0=Tails. In R we can generate a coin flip:

```
## single coin flip
flip = 1*(runif(1) > 0.5)

## 100 coin flips stored in a vector
flip.vec = 1*(runif(100) > 0.5)
```

1. A gambler bets on a sequence of fair coin flips. For each coin flip, he wins \$1 if heads appears and loses \$1 if tails appears. He stops when he has won 2 consecutive flips. Use R to simulate his average total winnings (assume that he executed this strategy many times and take the average of his total winnings). Note that negative winnings are losses.

```
## The number of simulated games.
reps = 1e4
winnings.vector = NULL

## Games loop.
```

```
for (r in 1:reps)
{
  ## To start a sequence of bets, set his winnings
  ## to zero
  winnings = 0
  last.flip = 0

  ## Start flipping until we win 2 times in a row
  while (1)
  {
    ## Flip the fair coin
    this.flip = 1*(runif(1) > 0.5) ## 1=Heads, 0=Tails

    ## Add 1 dollar to our winnings if we got heads,
    ##  and subtract 1 if we got tails
    winnings = winnings + 1 - 2*(this.flip == 0)

    ## Stop flipping when we have 2 consecutive heads
    if ((this.flip==1) & (last.flip == 1)) { break }
    last.flip = this.flip
  }
  winnings.vector[r] = winnings
}
mean(winnings.vector)
```

The expected value of the total winnings is zero with this strategy, and is zero if he choses to stop after winning any finite number of flips in a row. This is because N, the random variable corresponding to the number of flips he bets on, is a stopping time of a gambling strategy, and applying Wald's equation, the expected winnings is the product of the expected value of N and the expected value of his winnings on a single flip, which is zero, yielding expected winnings equal to zero.