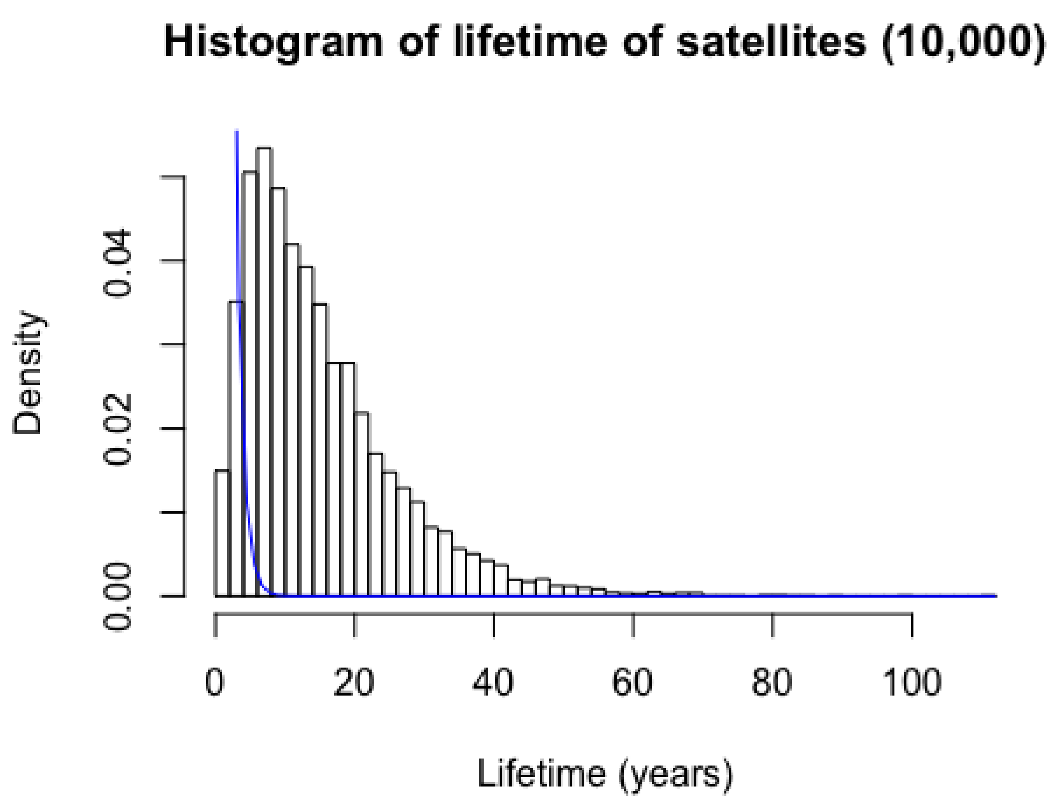
Poisson & Monte Carlo

Part 1: simulate a Poisson distribution and use Monte Carlo sampling to validate result.

```
#     i. simulate one draw:
      rexp(1,1/15)
```

```
## [1] 8.715887
```

```
#     ii. simulate 10,000 draw:
      x <- rexp(10000,1/15)
      x <- pmax(rexp(10000, 0.1), rexp(10000,0.1))
#     iii. create histogram:
      hist(x, main="Histogram of lifetime of satellites (10,000)",
           xlab="Lifetime (years)",
           probability = T,
           breaks=50)
      curve(dexp(x),add=T,col=4)
```




```
#     iv. estimate E(T):
      mean(x)
```

```
## [1] 14.96165

#     v. estimate P(T>15):
      #length(x[x>15])/length(x)
      mean(x>15)

## [1] 0.3967

      #for (i in c(1:5)) {
      x <- rexp(100000,1/15)
      mean(x)

## [1] 15.03576

      mean(x>15)

## [1] 0.36781

      #}

      sim.fun <- function(nsim, lambda.A = 0.10, lambda.B = 0.10){
        x <- replicate(nsim, max(rexp(1, lambda.A), rexp(1, lambda.B)))
        result <- c(mean = mean(x), prob = mean(x > 15))
        return(result)
      }
      round(replicate(5, sim.fun(1000)), 3)

##        [,1]   [,2]   [,3]   [,4]   [,5]
## mean 15.627 14.883 14.641 15.005 14.988
## prob  0.422  0.384  0.378  0.401  0.405

      round(replicate(5, sim.fun(10000)), 3)

##        [,1]   [,2]   [,3]   [,4]   [,5]
## mean 15.010 14.927 15.005 15.095 14.846
## prob  0.391  0.394  0.395  0.404  0.395
```

Part 2: use Monte Carlo to estimate pi.

```
# generate random points:
  i <- runif(10000)
  j <- runif(10000)

# check points fall in circle with radius 0.5, center (0.5, 0.5)
  points <- ((i-0.5)^2 + (j-0.5)^2) <= (0.5)^2

# estimate pi=area/(r^2):
  mean(points)/(0.5)^2

## [1] 3.1656
```