R handout: Monte Carlo Method and Application

## Methodology

### 1. Motivation

In practice, we may meet some hard integrations which is really hard to solve. For example, in one of my working papers, we have this integrand with the domain

where , and are constants. is the c.d.f. of the standard normal distribution. is the quantile function the standard normal distribution. is the quantile function of with degrees of freedom.

The analytical solution probably is desperate, but we can use other methods to obtain the numerical solution. This handout will show a way to solve integrations numerically.

### 2. Example. where . Find out

#### (a) Method 1: Direct Integration (By hand)

As we know, the p.d.f. of

where is the p.d.f. of the standard normal distribution. The expectation

#### (b) Method 2: Direct Integration (In R)

* R code
* # define the p.d.f. of Y  
  Y.pdf <- function(y) 2 \* dnorm(y)  
    
  # define the integrand of the expecation of Y  
  E.Y <- function(y) y \* Y.pdf(y)  
    
  # Integrate the integrand  
  integrate(E.Y, lower = 0, upper = Inf)  
  # Result: 0.7978846

#### (c) Monte Carlo Method

Suppose is a random variable with a p.d.f. and a c.d.f. over a domain . Also, we have a sample from with observations.

Besides, because of Central Limit Theorem, asymptotically follows where is the mean of and is the variance of . So, when , converges to in probability. Hence, intuitively, the result is more precise with a larger sample size.

##### i. Method 3: Monte Carlo Method with Transformation 1

* Motivation

Our target distribution may be hard to be found, but the original distribution may be not. Hence, we simulate a sample from the original distribution and transform it to our target, and then, calculate the expectation.

* The analytical part

Because

where is the number of simulation and is a sample from .

* R code
* # set seed to make this process repeatable  
  set.seed(12345)  
    
  # set the number of simulation  
  n <- 1000  
    
  # simulate a sample from the standard   
  # normal distribution  
  X.s <- rnorm(n)  
    
  # calculate the mean of the absoluate values  
  mean(abs(X.s))  
  # Result: 0.7944

##### ii. Method 4: Monte Carlo Method with Transformation 2

* Motivation

Both of our target distribution and the original distribution may be hard to be found, so we use other relatively simple distributions which have same domains as the ones of our target distribution or the original distribution, to simulate our target, and then, calculate the expectation.

* The analytical part

Because has a same domain as ,

where is the number of simulation and is a sample from .

* R code
* # set seed to make this process repeatable  
  set.seed(12345)  
    
  # set the number of simulation  
  n <- 1000  
    
  # define the p.d.f. of Y  
  Y.pdf <- function(y) 2 \* dnorm(y)  
    
  # simulate a sample from the standard   
  # normal distribution  
  Y.s <- rexp(n)  
    
  # calculate the mean of the absoluate values  
  mean(Y.s \* Y.pdf(Y.s) / dexp(Y.s))  
  # Result: 0.7729

### 3. Summary

#### (a) Comparison of these methods

## Method1 Method2 Method3 Method4  
## Results 0.7979 0.7979 0.7944 0.7729

where Method3 and Method4 have 1000 simulations.

#### (b) Steps of Monte Carlo Method

1. Check the domain
2. Simulate an appropriate sample over the domain
3. Calculate the mean via an appropriate transformation

### 4. Practice

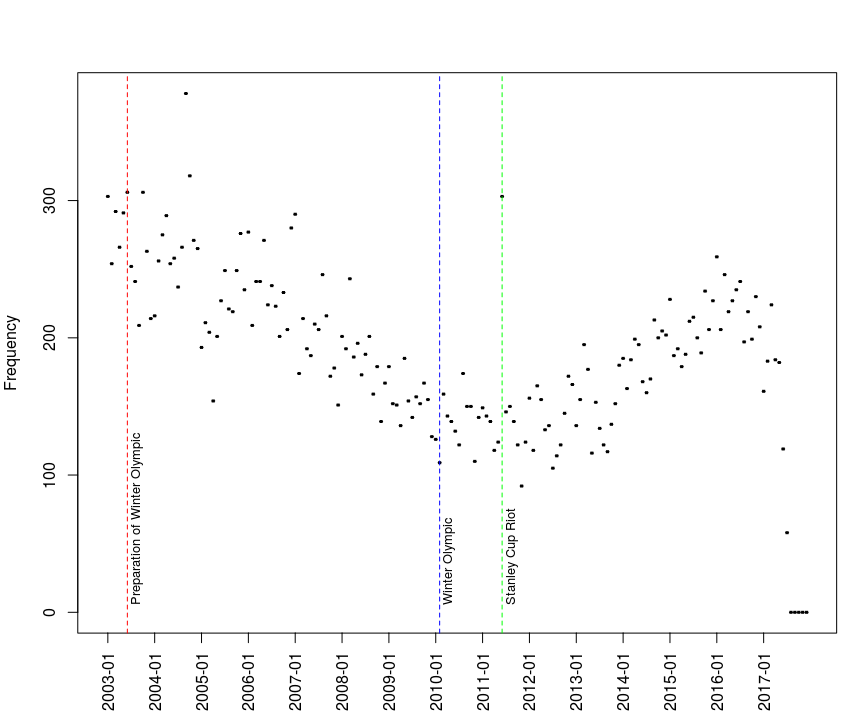
Suppose , where . Use Monte Carlo Method to find .

Hint: You need to show the analytical part and the coding part in R, separately. Also, you may need to use and in R

## Application

This data comes from the Vancouver Open Data Catalogue. It has 530,652 records regarding to different types of crimes between Jan 01, 2003 and July 13, 2017. The original data set contains coordinates in UTM Zone 10 with Latitude and Longitude. (Kaggle 2017)

Suppose our target is to predict the monthly frequency of the type of crimes “Break and Enter Commercial” in the future and we are not very interested in the exact locations of crimes. The frequency during the period between Jan 01, 2003 and July 13, 2017 can be showed as the following plot:



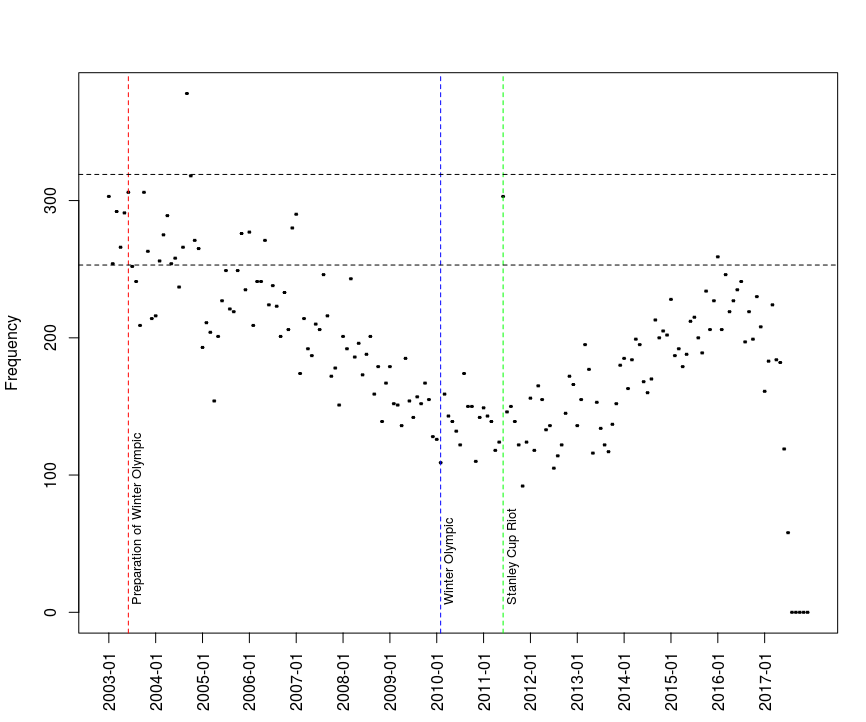
where on July 2, 2003, Vancouver won the bid to host the Winter Olympic (the vertical red dashed line). The game was hosted from February 12 to 28, 2010 (the vertical blue dashed line). So I called the period between July 2, 2003 and February, 28, 2010 as the preparation of Winter Olympic. Also, on June 15, 2011, there was a riot, 2011 Vancouver Stanley Cup riot (the vertical green dashed line). It is obvious for us to observe that the frequency has a convex shape and one of possible reasons causing this phenomenon is the Winter Olympic. Besides, the fraction of data after 2017 may be not trustful, because the trend of this fraction is supposed to be increasing or maintain the same level.

Intuitively, I will guess the frequency of crimes are increasing until reaching the situation before the preparation of Winter Olympic. Hence, I will use the fraction of the data before July 2003, as my training data, to predict the frequency after 2017.

Suppose is from my training data and this data follows a Poisson distribution with parameter . Find out the 2.5% and 97.5% quantile. The p.d.f. of

By m.l.e, , so the 2.5% and 97.5% quantile

They can be showed on the plot



where the upper horizontal dashed line is the 97.5% quantile, 319 and the lower horizontal dashed line is the 2.5% quantile, 253.

Because is a sample, shall have its own distribution. Suppose . Find out the 2.5% and 97.5% quantile.

The p.d.f. of given

And the p.d.f. of

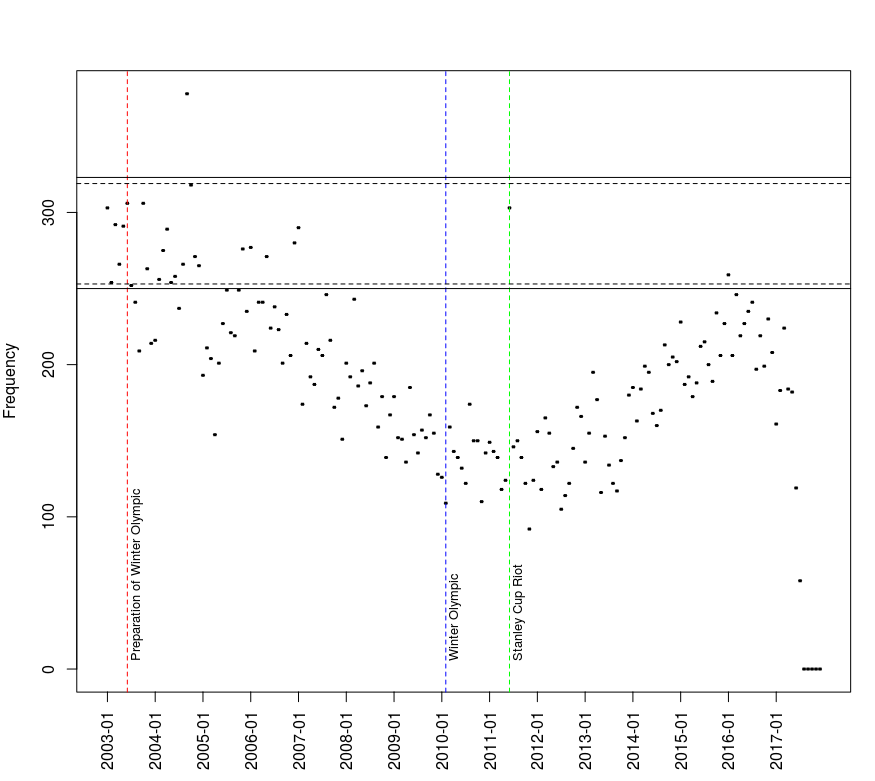
Then

So, the 2.5% and 97.5% quantile

The analytical solution is complicated, but it is relatively easy if we use numerical methods. By Monte Carlo Method,

where is a sample following . Also, the equation can be expressed as the following (the root searching form)

where and . The quantiles and are roots of the equation. After the computation, we can show the quantiles on the plot:



where the upper horizontal solid line is the 97.5% quantile, 323, and the lower horizontal solid line is the 2.5% quantile, 250.

* R code for the whole process
* #################################################  
  # describe BEC during the period  
  #################################################  
  # load the new data  
  addr <- paste('https://raw.githubusercontent.com/bolus123',   
  '/R-handout/master/MCandApp/BEC.csv', sep = '')  
  BEC.monthly.freq <- read.csv(file = addr)[, -1]  
    
  # add a new column combining YEAR with MONTH  
  BEC.monthly.freq <- cbind(BEC.monthly.freq,   
  paste(BEC.monthly.freq$YEAR,   
  substr(  
  as.character(  
  as.numeric(  
  BEC.monthly.freq$MONTH) +   
  100), 2, 3)  
  , sep = '-'))  
    
  # build a basic scatter frequency plot   
  # during the period  
  plot(BEC.monthly.freq[, 5], BEC.monthly.freq[, 4],   
  xaxt="n", ylab = 'Frequency')  
  # define the tick of x-axis  
  labs <- sort(BEC.monthly.freq[, 5])[rep(c(T, F, F,   
  F, F, F, F, F, F, F, F, F), 15)]  
  for (i in 1:15){  
    
  axis(1, at = (12 \* (i - 1) + 1),   
  labels = labs[i], las = 2)  
    
  }  
    
  # specify special months  
  prepare.Winter.Olympic <- BEC.monthly.freq[, 5] == '2003-06'  
  Winter.Olympic <- BEC.monthly.freq[, 5] == '2010-02'  
  Stanley.Cup.riot <- BEC.monthly.freq[, 5] == '2011-06'  
    
  # show Preparation of Winter Olympic on the plot  
  abline(v = BEC.monthly.freq[prepare.Winter.Olympic, 5],   
  col = 'red', lty = 2)  
  text(BEC.monthly.freq[prepare.Winter.Olympic, 5], 5,   
  'Preparation of Winter Olympic', pos = 4,   
  srt = 90, cex = 0.8)  
    
  # show Winter Olympic on the plot   
  abline(v = BEC.monthly.freq[Winter.Olympic, 5],   
  col = 'blue', lty = 2)  
  text(BEC.monthly.freq[Winter.Olympic, 5], 5,   
  'Winter Olympic', pos = 4, srt = 90, cex = 0.8)  
    
  # show Stanley Cup Riot on the plot   
  abline(v = BEC.monthly.freq[Stanley.Cup.riot, 5],   
  col = 'green', lty = 2)  
  text(BEC.monthly.freq[Stanley.Cup.riot, 5], 5,   
  'Stanley Cup Riot', pos = 4, srt = 90, cex = 0.8)  
    
  # set the maximun date of the training data  
  x1.max.date <- which(  
  BEC.monthly.freq[  
  order(BEC.monthly.freq[, 5]), 5] == '2003-06')  
    
  # cut it off from the original data  
  x1 <- BEC.monthly.freq[  
  order(BEC.monthly.freq[, 5]), 4][1:x1.max.date]  
    
  # fit a poisson model for the training data  
  lambda1 <- mean(x1)  
    
  # show the 2.5% and 97.5 quantiles on the plot  
  abline(h = qpois(0.975, lambda1), lty = 2)  
  abline(h = qpois(0.025, lambda1), lty = 2)  
    
  #################################################  
  # find the gamma distribution for lambda   
  # by the nonparametric bootstrap  
  #################################################  
  set.seed(12345)  
    
  # the number of times for bootstrapping  
  n <- 100000  
  # set a vector to carry the means  
  xs.means <- rep(NA, n)  
    
  for (i in 1:n){  
  # resample from the training data  
  xs <- sample(x1, 6, replace = T)  
  # calculate their means  
  xs.means[i] <- mean(xs)  
  }  
    
  # calculate the grand mean  
  mu <- mean(xs.means)  
  # calculate the grand variance  
  sigma2 <- var(xs.means)  
    
  # fit a gamma distribution by the method of moment  
  alpha <- mu^2 / sigma2  
  beta <- sigma2 / mu  
    
  # check the gamma distribution  
  #x <- 1:1000   
  #plot(x, dgamma(x, alpha, 1/beta), type = 'l')  
    
  #################################################  
  # fit a new poisson distribution   
  # with a parameter lambda   
  # which is a random variable  
  #################################################  
  # set a seed make this process repeatable  
  set.seed(12345)  
    
  # define a user-defined function  
  # p is P(X <= x)  
  # alpha is the alpha for gamma(alpha, beta)  
  # beta is the beta for gamma(alpha, beta)  
  # interval is the range of searching the quantile  
  # rnum is the number of simulations  
  X.quantile <- function(p, alpha, beta,   
  interval = c(100, 500), rnum = 10000){  
    
  root.finding <- function(x, p, lambda){  
  # The root searching form  
  p - mean(ppois(x, lambda))  
  }  
    
  # simulate a sample from gamma distribution  
  lambda <- rgamma(rnum, alpha, scale = beta)  
    
  # search the root by the bisection method  
  uniroot(root.finding, interval = interval,   
  p = p, lambda = lambda)$root  
    
  }  
    
  # calcualte the 2.5% quantile  
  q0025 <- X.quantile(p = 0.025,   
  alpha = alpha, beta = beta)  
  # calcualte the 97.5% quantile  
  q0975 <- X.quantile(p = 0.975, alpha = alpha, beta = beta)  
    
  # add horizontal lines on the plot  
  abline(h = q0025)  
  abline(h = q0975)

# Reference

Kaggle. 2017. “Crime in Vancouver.” 2017. <https://www.kaggle.com/wosaku/crime-in-vancouver>.