Statistics Software Lab Report - 4 (Outputs file)

Name of the Student: Shatansh Patnaik
Roll No: 20MA20067

IIT Kharagpur
Statistics Software Lab

Output for Exercise-1

```
> ##########################################################s##########
> # a) Generating random permutations
> set.seed(67)
>
> generate_random_permutation <- function (n, p) {
+   k <- n
+
+   while(k>=1){
+     u <- runif(1, 0, 1)
+     I <- floor(k*u) + 1
+     temp <- p[I]
+     p[I] <- p[k]
+     p[k] <- temp
+     k <- k-1
+   }
+
+   return(p)
+ }
>
> n <- 10
> p <- seq(1, n, 1)
> cat("Initial Permutation is: \n")
Initial Permutation is:
> cat(p)
1 2 3 4 5 6 7 8 9 10> q <- generate_random_permutation(n, p)
> cat("\nFinal Permutation is: \n")

Final Permutation is:
> cat(q, "\n")
3 1 5 6 8 10 7 4 2 9
> ##########################################################s##########
```

Output for Exercise-2

```
> ##########################################################s##########
> # b) Generation of a stationery Poisson Process
> generate_stationery_poisson <- function(T, S, lambda){
+   t <- 0
+   I <- 0
+
+   while(t < T){
+     u <- runif(1, 0, 1)
+     t <- t - (1/lambda)*log(u)
+     if (t>T)
+       break
+     I <- I + 1
+     S[I] <- t
+   }
+   return(S)
```

```
+ }
>
> T <- 10
> S <- c()
> lambda <- 5
>
> S <- generate_stationery_poisson(T, S, lambda)
> cat("The resultant array is as follows: ", S)
The resultant array is as follows:  0.7132501 0.7247624 0.8298674 0.8608822
    0.9322894 1.131573 1.303386 1.321246 1.346822 1.6533 1.675817 1.771621
    1.966995 2.240367 2.383362 2.680529 3.166485 3.385787 4.295339 4.316184
    4.42878 4.837133 5.159777 5.199202 5.23906 5.704794 5.862096 6.163606
    6.165663 6.323427 6.386015 6.58073 6.595207 6.65816 7.061228 7.22715
    7.594829 7.730231 7.88249 8.211276 8.418683 8.518424 8.970141 9.040554
    9.438016 9.59068 9.604993 9.774426
> ########################################################s###########
```

Output for Exercise-3a

```
> ########################################################s###########
> # c)
> # Algorithm - 1 : Generation of a non-stationery Poisson Process
> intensity_function <- function (t, lambda) {
+   return(lambda*exp(-t))
+ }
>
> generate_non_stationery_poisson_algo_1 <- function(T, S, lambda_upper_
    bound, fn){
+   t <- 0
+   I <- 0
+
+   while(t < T){
+     u <- runif(1, 0, 1)
+     t <- t - (1/lambda)*log(u)
+     if (t>T)
+       break
+     if(u <= fn(t, lambda)/lambda_upper_bound){
+       I <- I + 1
+       S[I] <- t
+     }
+   }
+   return(S)
+ }
>
> T <- 1000
> S <- c()
> lambda <- 40
>
> S <- generate_non_stationery_poisson_algo_1(T, S, lambda, intensity_
    function)
> cat("The resultant array is as follows: ", S)
```

```
The resultant array is as follows:  0.01238433 0.0170233 0.05997008
    0.07643351 0.08891763 0.09480655 0.1288624 0.139562 0.2000077 0.2714118
    0.3086383 0.3227021 0.3651778 0.4184447 0.459363 0.5116289 0.5465375
    0.5834553 0.6343951 0.6982244 0.7944398 0.8333842 0.8835149 0.9542637
    1.001949 1.099725 1.178052 1.252361 1.326366 1.409252 1.468056 1.570278
    1.628812 1.740452 1.990323 2.208586 2.278075 2.551064 3.279962 3.800597
> #########################################################s###########
```

Output for Exercise-3b

```
> #########################################################s###########
> # Algorithm - 2 : Generation of a non-stationery Poisson Process
> intensity_function <- function (t, lambda) {
+   return(lambda*exp(-t))
+ }
>
> generate_non_stationary_poisson_algo2 <- function(fn, intervals, l, k){
+   t <- 0
+   J <- 1
+   I <- 0
+   S <- numeric(0)
+   flag <- FALSE
+
+   while(flag == FALSE){
+     u1 <- runif(1)
+     X <- (-(1/l[J])*log(u1))
+     while(TRUE){
+       if(t+X <= intervals[J]){
+         t<- t+X
+         u2 <- runif(1)
+         if(u2 <= fn(t,l[J])/l[J]){
+           I <- I+1
+           S[I] <- t
+         }
+         break
+       }
+       if(J == k+1){
+         flag <- TRUE
+         break
+       }
+       X <- (X-intervals[J]+t)*l[J]/l[J+1]
+       J <- J + 1
+     }
+   }
+   return(S)
+ }
>
> intervals <- seq(10,100,5)
> k <- length(intervals)-1
> l <- numeric(0)
>
```

```
> for(i in 0:k+1){
+   l[i] = runif(1,1,50)
+ }
> getAns <- generate_non_stationary_poisson_algo2(intensity_function,
    intervals, l, k)
> print(getAns)
 [1] 0.02709049 0.04462291 0.05856059 0.07032225 0.13296169 0.16152925
     0.19201378 0.31355503 0.38690543 0.47411528 0.49332752 0.73934279
[13] 0.83090932 0.93424072 0.99721979 0.99859952 1.09092378 1.21280807
    1.22631339 1.22697635 1.29093896 1.31301272 1.34646459 1.38252903
[25] 1.50846175 1.54086721 1.78652062 1.82721212 1.97178329 2.16137535
    2.91757142 3.53572290 4.41556647
> ######################################################s###########
```