MA - 226 Assignment Report

Ayush Sharma	(150123046)
Shashank Oberoi	(150123047)
Ojasvin Sood	(150123048)
Yugansh Bhatnagar	(150123050)
Remidi Nishanth Reddy	(150123051)

12 April 2017

Q 1. $F(x)$ be General Exponential Distribution, i.e. $F(x) = (1-e^{-\lambda x})^{\alpha}$. Generate random numbers from distribution $F_1(x) = (1+\lambda)F(x) - \lambda F^2(x)$, where $\lambda \in [-1,1]$, using :

- Inverse Transform
- Acceptance Rejection

Solution

- **Inverse Transform Method.**

  Generating random number from the given distribution $F_1(x)$ using Inverse Transform Method:

$$
\begin{aligned}
& F_1(x) = (1+\lambda)F(x) - \lambda F^2(x) = u, \text{ where } u \sim \mathcal{U}(0,1). \\
\Rightarrow & \lambda F^2(x) - (1+\lambda)F(x) + u = 0 \\
\Rightarrow & F(x) = \frac{(1+\lambda) - \sqrt{(1+\lambda)^2 - 4\lambda u}}{2\lambda} = q \qquad \{\text{see NOTE}\} \\
\Rightarrow & F(x) = (1-e^{-\lambda x})^{\alpha} = q \\
\Rightarrow & (1-e^{-\lambda x}) = q^{\frac{1}{\alpha}} \\
\Rightarrow & e^{-\lambda x} = (1-q^{\frac{1}{\alpha}}) \\
\Rightarrow & -\lambda x = \log(1-q^{\frac{1}{\alpha}}) \\
\Rightarrow & x = -\frac{1}{\lambda}\log(1-q^{\frac{1}{\alpha}})
\end{aligned}
$$

NOTE: Here we have only considered $F(x) = \frac{(1+\lambda)-\sqrt{(1+\lambda)^2-4\lambda u}}{2\lambda}$ and not $F(x) = \frac{(1+\lambda)+\sqrt{(1+\lambda)^2-4\lambda u}}{2\lambda}$ because

$$
\begin{aligned}
& u \sim U(0,1) \Rightarrow u < 1 \\
\Rightarrow & (1+\lambda)^2 - 4\lambda u \ > \ (1+\lambda)^2 - 4\lambda \\
\Rightarrow & \sqrt{(1+\lambda)^2 - 4\lambda u} \ > \ |1-\lambda| \\
\Rightarrow & \sqrt{(1+\lambda)^2 - 4\lambda u} \ > \ 1-\lambda \quad \textit{given} \quad \lambda \in [-1,1] \\
\Rightarrow & (1+\lambda) - \sqrt{(1+\lambda)^2 - 4\lambda u} \ < \ (1+\lambda+\lambda-1) = 2\lambda \\
\Rightarrow & \frac{(1+\lambda) - \sqrt{(1+\lambda)^2 - 4\lambda u}}{2\lambda} < 1 \\
\Rightarrow & F(x) < 1 \\
\Rightarrow & F(x) = (1-e^{-\lambda x})^{\alpha} \ \ and \ \ 1 - e^{-\lambda x} \leq 1 \ , \ \lambda \geq 0
\end{aligned}
$$

Algorithm 1 Generating random number from the distribution by inverse transform method.

Generate $U \sim \mathcal{U}(0,1)$.

Generate Q from the relation $Q = \frac{(1+\lambda)-\sqrt{(1+\lambda)^2-4\lambda U}}{2\lambda}$.

Generate X from the relation $X = -\frac{1}{\lambda}\log(1-Q^{\frac{1}{\alpha}})$.

Return X.

Code for R

```
args<-commandArgs(TRUE)

gen_inv <- function(lambda, alpha, n) {
   U <- runif(n, 0, 1);
   Q <- vector(length = n);
   X <- vector(length = n);
   Q <- ((1 + lambda) - sqrt((1 + lambda)^2 - (4 * lambda * U))) / (2 *
       lambda);
   X <- - log(1 - Q^(1 / alpha)) / lambda;

   pdf("1a.pdf");
   hist(X, breaks = 50, main = "");
   legend('topright', legend = c(paste("lambda =", lambda), paste("alpha =
       ", alpha), paste("sample size =", n)), lty = 0, col = "white", bty =
        'n');

   cat("The sample mean and variance, given lambda =", lambda, ", alpha ="
       , alpha, ", and sample size ", n, ", are estimated to be", mean(X),
       ", and", var(X), ", respectively.\n")
}

lambda = as.numeric(args[1]);
alpha = as.numeric(args[2]);
n = as.integer(args[3]);

set.seed(1);

gen_inv(lambda, alpha, n);
```

Results:

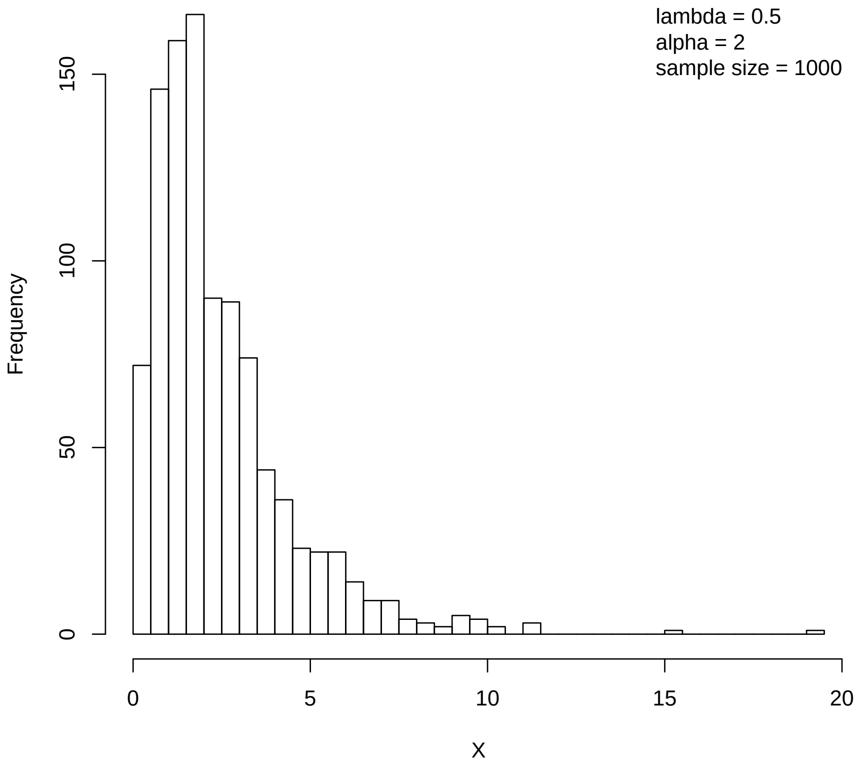

Figure 1: *Histogram for generated random numbers using Inverse Transform Method*

The sample mean and variance, given λ 0.5, α 2, and sample size 1000, are estimated to be 2.441197, and 3.944559, respectively.

- **Acceptance Rejection Method.**

  For generating random number from the distribution $F_1(x)$ by acceptance-rejection method, the distribution $F(x)$ is chosen as our candidate distribution.

  We can write the probability density function of distribution $F_1(x)$ as

$$f_1(x) = (1+\lambda)f(x) - \lambda F(x)f(x),$$

  where $f(x)$ is the probability density function of distribution $F(x)$.

  Also, we know that $F(x) = (1-e^{-\lambda x})^{\alpha}$.

  To apply inverse transform method we equate

$$u_1 = F(y) = (1-e^{-\lambda y})^{\alpha} \Rightarrow y = -\frac{1}{\lambda}\log(1-u_1^{\frac{1}{\alpha}}),$$

  where $u_1 \sim \mathcal{U}(0,1)$.

  We choose c maximizing

$$\frac{f_1(y)}{f(y)} = (1+\lambda) - 2\lambda F(y),$$

  where $-1 \leq \lambda \leq 1$ and $0 \leq F(y) \leq 1$.

  The maximum value attained will be $(1+|\lambda|)$ which is to be taken as c.

  Here,the rejection test $u_2 > \frac{f_1(y)}{cf(y)}$ can be implemented as

$$u_2 > \frac{(1+\lambda)-2\lambda F(y)}{1+|\lambda|} = \frac{(1+\lambda)-2\lambda u_1}{1+|\lambda|},$$

  where $u_2 \sim \mathcal{U}(0,1)$.

Algorithm 2 Generating random number from the distribution by acceptance-rejection method.

1: Generate $U_1, U_2 \sim \mathcal{U}[0,1]$.
2: **if** $U_2 \leq \frac{(1+\lambda)-2\lambda U_1}{1+|\lambda|}$ **then**
3:     $X = -\frac{1}{\lambda}\log(1-U_1^{\frac{1}{\alpha}})$.
4:     Return X.
5: **else**
6:     return to step 1.
7: **end if**

Code for R

```r
args<-commandArgs(TRUE)

gen_ar <- function(lambda, alpha, n) {
  X <- vector(length = n);

  for (i in 1:n) {
    repeat {
      u<-runif(2, 0, 1);
      if (u[2] <= ((1+lambda) - (2*lambda*u[1]))/(1 + abs(lambda))) {
        X[i] = - (log(1 - u[1]^(1 / alpha)) / lambda);
        break;
      }
    }
  }

  pdf("1b.pdf");
  hist(X, breaks = 50, main = "");
  legend('topright', legend = c(paste("lambda =", lambda), paste("alpha = ", alpha), paste("sample size =", n)), lty = 0, col = "white", bty = 'n');

  cat("The sample mean and variance, given lambda", lambda, ", alpha", alpha, ", and sample size ", n, ", are estimated to be", mean(X), ", and", var(X), ", respectively.\n")
}

lambda = as.numeric(args[1]);
alpha = as.numeric(args[2]);
n = as.integer(args[3]);

set.seed(1);

gen_ar(lambda, alpha, n);
```

Results:

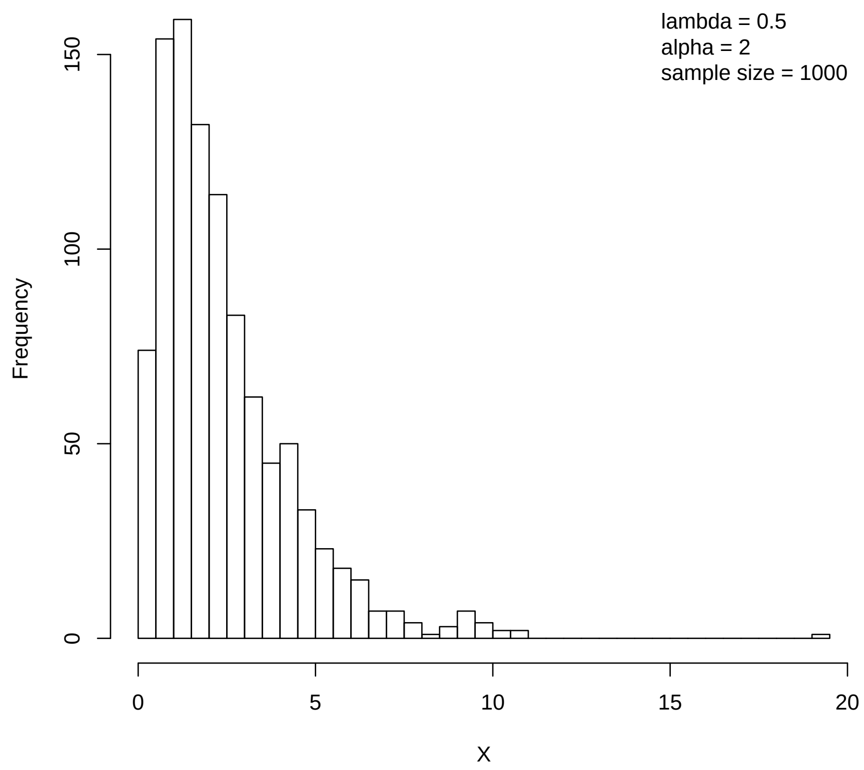

Figure 2: *Histogram for generated random numbers using Acceptance Rejection Method*

The sample mean and variance, given λ 0.5, α 2, and sample size 1000, are estimated to be 2.43525, and 3.771058, respectively.

Q 2. $F(x,y)$ be Bi-variate General Exponential Distribution(BVGE). Generate random numbers from distribution $F_1(x,y) = (1+\lambda)F(x,y) - \lambda F^2(x,y)$, where $\lambda \in [-1,1]$.

Solution

For generating random number from the distribution $F_1(x,y)$ by acceptance-rejection method, the distribution $F(x)$ is chosen as our candidate distribution.

We can write the probability density function of distribution $F_1(x,y)$ as

$$f_1(x,y) = (1+\lambda)f(x,y) - 4\lambda F(x,y)f(x,y),$$

where $f(x,y)$ is the probability density function of distribution $F(x,y)$.

Also, we know that $F(x,y) = (1-e^{-\lambda x})^{\alpha_1} \cdot (1-e^{-\lambda y})^{\alpha_2} \cdot (1-e^{-\lambda\ z})^{\alpha_3}$, where $z = min(x,y)$.

To apply inverse transform method

$$U_0 \sim GE(\alpha_1, \lambda) \qquad U_1 \sim GE(\alpha_2, \lambda) \qquad U_2 \sim GE(\alpha_3, \lambda)$$

$$\begin{aligned} P(max(U_0,U_1) \leq x, max(U_0,U_2) \leq y) &= P(U_0 \leq x, U_1 \leq x, U_0 \leq y, U_2 \leq y) \\ &= P(U_0 \leq min(x,y), U_1 \leq x, U_2 \leq y) \\ &= P(U_0 \leq min(x,y)) \cdot P(U_1 \leq x) \cdot P(U_2 \leq y) \\ &= (1-e^{-\lambda\ min(x,y)})^{\alpha_0} \cdot (1-e^{-\lambda x})^{\alpha_1} \cdot (1-e^{-\lambda y})^{\alpha_2} \end{aligned}$$

And hence if $(X_1, X_2) \sim$ BVGE $(\lambda, \alpha_1, \alpha_2, \alpha_3)$ then the joint CDF of (X_1, X_2) for $x_1 > 0, x_2 > 0$ is

$$F_{X_1,X_2}(x_1,x_2) = (1-e^{-\lambda x_1})^{\alpha_1} \cdot (1-e^{-\lambda x_2})^{\alpha_2} \cdot (1-e^{-\lambda z})^{\alpha_3},$$

where $z = min(x_1, x_2)$.

We choose c maximizing

$$\frac{f_1(x,y)}{f(x,y)} = (1+\lambda) - 4\lambda F(x,y),$$

where $-1 \leq \lambda \leq 1$ and $0 \leq F(x,y) \leq 1$.

The maximum value attained will be $(1+\lambda)$ which is to be taken as c.

Here,the rejection test $u > \frac{f_1(x,y)}{cf(x,y)}$ can be implemented as

$$u > \frac{(1+\lambda) - 4\lambda F(x,y)}{1+\lambda},$$

where $u \sim \mathcal{U}(0,1)$.

Algorithm 3 Generating random number from the distribution by acceptance-rejection method.

1: Generate $U_0 \sim GE(\alpha_1, \lambda), U_1 \sim GE(\alpha_2, \lambda), U_2 \sim GE(\alpha_3, \lambda)$.

2: Generate $X = (X_1, X_2)$ from the relation $X_1 = max\{U_0, U_1\}$ and $X_2 = max\{U_0, U_2\}$.

3: **if** $u \leq 1 - \frac{4\lambda F(X)}{1+\lambda}$, where $u \sim \mathcal{U}(0,1)$ **then**

4: Return X.

5: **else**

6: return to step 1.

7: **end if**

Code for R

```r
args<-commandArgs(TRUE)

library(MASS) #For kde2d

#Function BVGEF Computes Value F(X)
BVGEF <- function(X, lambda, alpha) {
    return ((1 - exp(-lambda * X[1]))^(alpha[2]) * (1 - exp(-lambda * X[2]))^(
        alpha[3]) * (1 - exp(-lambda * min(X)))^(alpha[1]));
}

#Function GE generates n numbers from generalised exponential
GE <- function(n, lambda, alpha) {
    U <- runif(n,0,1);
    return (- log(1 - U^(1 / alpha)) / lambda);
}

#Function BVGE generates n numbers from Bivariate generalised exponential
BVGE <- function(n, lambda, alpha) {
    bvge <- matrix(0, nrow = n, ncol = 2);

    for(i in 1:n) {
        U <- vector(length = 3);
        U[1]<-GE(1,lambda,alpha[1]);
        U[2]<-GE(1,lambda,alpha[2]);
        U[3]<-GE(1,lambda,alpha[3]);

        bvge[i,] <- c(max(U[1], U[2]), max(U[1], U[3]));
#       bvge[i,1] <- max(U[1], U[2]);
#       bvge[i,2] <- max(U[1], U[3]);
```

```
    }
    return (bvge);
}

#Function TRUNC_BVGE generates n numbers from Truncated Bivariate generalised exponential
TRUNC_BVGE <- function(n, lambda, alpha) {
    trunc_bvge<-matrix(0, nrow = n, ncol = 2);

    for (i in 1:n) {
        repeat {
            u<-runif(1, 0, 1);
            X <- BVGE(1, lambda, alpha);
            F <- BVGEF(X, lambda, alpha);

            if (u <= (1 - (4 * lambda * F)/(1 + lambda))) {
                trunc_bvge[i,] <- X;
                break;
            }
        }
    }

    return(trunc_bvge);
}

n = as.integer(args[1]);
lambda = as.numeric(args[2]);
#alpha <- vector(length = 3);
alpha <- c(as.numeric(args[3]), as.numeric(args[4]), as.numeric(args[5]));

set.seed(1);

#X <- TRUNC_BVGE(1000,1,0.5,0.6,0.7);
X <- TRUNC_BVGE(n, lambda, alpha);
f <- kde2d(X[,1], X[,2], n = n); # Two dimensional kernel density approximation

cat("The sample means, variances, and covariance, given lambda", lambda, ", alpha", alpha, ", and sample size ", n, ", are estimated to be :","\nmean(X1) =", mean(X[,1]), "\nmean(X2) =", mean(X[,2]), "\nvariance(X1) =", var(X[,1]), "\nvariance(X2) =", var(X[,2]), "\ncorrelation(X1, X2) =", cor(X[,1], X[,2]), "\n");
```

```
pdf("2.pdf");
#contour(f, xlab = "X1", ylab = "X2", main = "")
persp(f, xlab = "X1", ylab = "X2", zlab = "Density", main = "", box = TRUE)
legend('topright', legend = c(paste("lambda =", lambda), paste("alpha =", alpha
    [1], ",", alpha[2], ",", alpha[3]), paste("sample size =", n)), lty = 0,
    col = "white", bty = 'n');
```

Results:

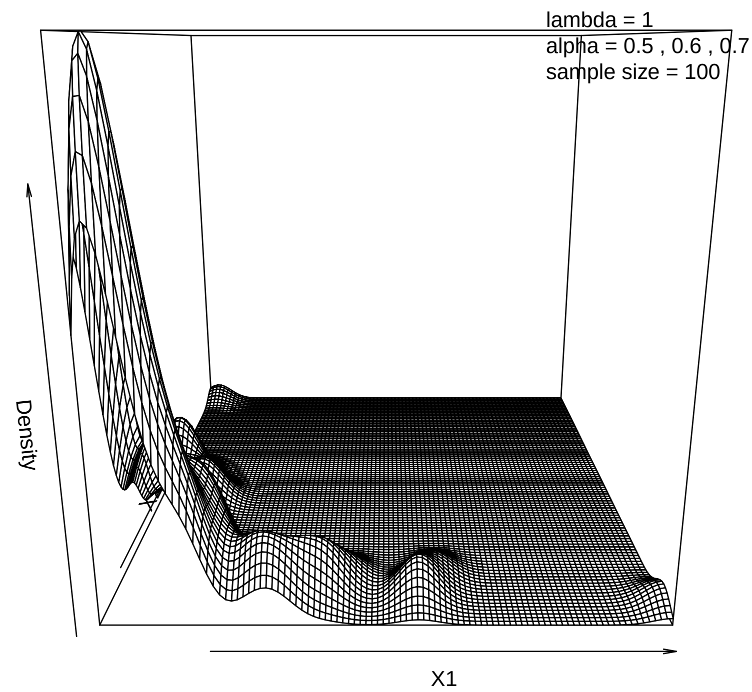

Figure 3: *Density Plot for generated bi-variate random numbers using Acceptance Rejection Method*

The sample means, variances, and covariance, given λ 1, α (0.5, 0.6, 0.7), and sample size 100, are estimated to be :

mean(X1) = 0.593018; mean(X2) = 0.6326948

variance(X1) = 0.5792707; variance(X2) = 0.5200241

correlation(X1, X2) = -0.137354