Contents

Generating continuous Random variables

Abhinav Gupta
(150123001)

February 9, 2017

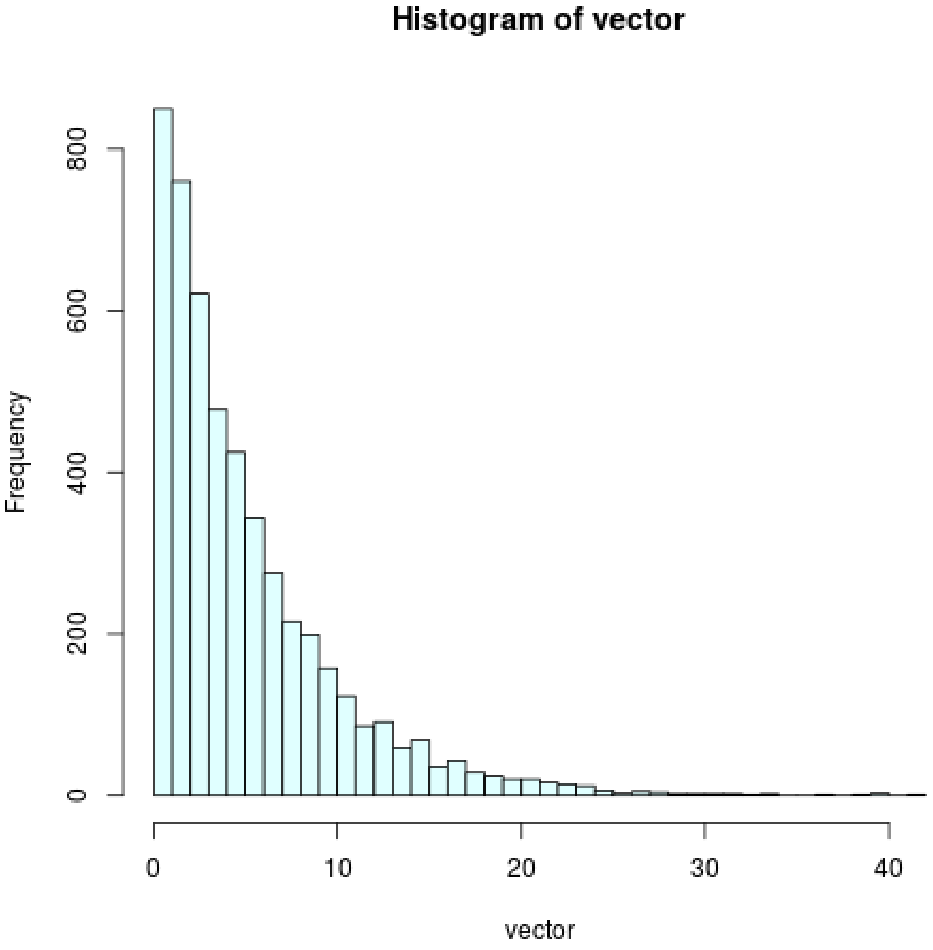
1 Short Summary

2 Question-1

2.1 Observation:

- By **"The Inverse transform algorithm"**, we can see that our approximation is almost correct and resembles the exponential distribution which can be further proved by the graph of the sample variables generated vs the frequency graphs.
- **R Code and Histogram :** Mean: 4.902336 Maximum: 47.89849 Minimum: 0.00195348

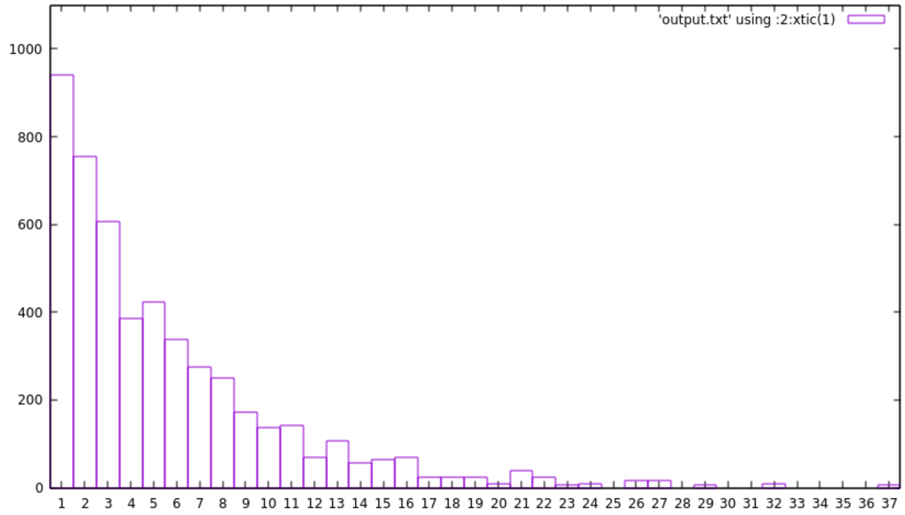
```
mean<-0
max<- -2147483646
min<- 2147483646
vector<-c()
for(i in 1:5000){
        u<-runif(1)
        sample<- -5*log(1-u)
        vector[i]<-sample
        cat("Sample Values are: ",sample)
        cat("\n")
        mean<-mean+sample
        if(max<=sample)
                max<-sample
        if(min>=sample)
                min<-sample
}
png("question1.png")
hist(vector, breaks=50 , col="light cyan",plot=TRUE)
cat("Mean: ",mean/5000)
cat("\nMaximum: ",max)
cat("\nMinimum: ",min)
cat("\n")
```



- **C++ Code and Histogram :** Mean is : 5.098216 Maximum is :36.905397 Minimum is :0.002559

```
#include <stdio.h>
#include <iostream>
#include <stdlib.h>
#include <math.h>
#include <limits.h>
#include <algorithm>
int main(){
        long long int x;
        int a,i,k;
        double m,sample[5000],mean,maximum=INT_MIN ,minimum=INT_MAX;
        x=23;
        a=16807;
        m=44947;
        int count=5000;
        float u,j;
        for(i=0;i<count;i++){
                u=x/m;
                x=(x*a);
                x%=(long long int)m;
                sample[i]=-5*log(1-u);
                mean+=(sample[i]/count);
                if(sample[i]>maximum)
                        maximum=sample[i];
                if(sample[i]<minimum)
                        minimum=sample[i];
        }
        int frequency[(int)ceil(maximum)]={0};
        for(i=0;i<5000;i++){
                frequency[(int)ceil(sample[i])-1]++;
        }
        for(i=0;i<(int)ceil(maximum);i++){
                printf("%d %d\n",i+1,frequency[i]);
        }
        printf("Mean is : %lf\n",mean);
        printf("Maximum is :%lf\n",maximum);
        printf("Minimum is :%lf\n",minimum);

}
```



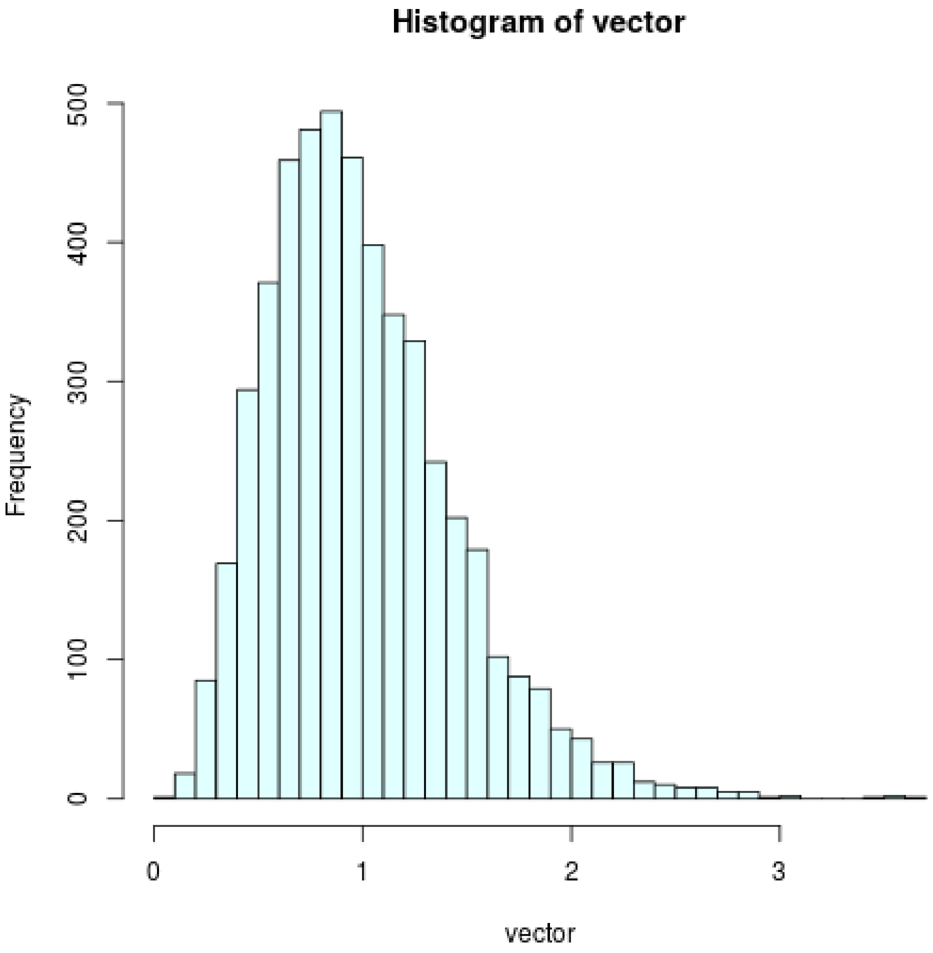
3 Question-2

3.1 Observation:

- By using the relation between **a gamma variable and the sum of** *alpha* **exponential variables** and also **"The Inverse transform algorithm"**, we can see that our approximation is almost correct and resembles the gamma distribution which can be further proved by the graph of the sample variables generated vs the frequency graphs.

- **R Code and Histogram :** Mean: 0.9921659 Maximum: 3.441478 Minimum: 0.04518859

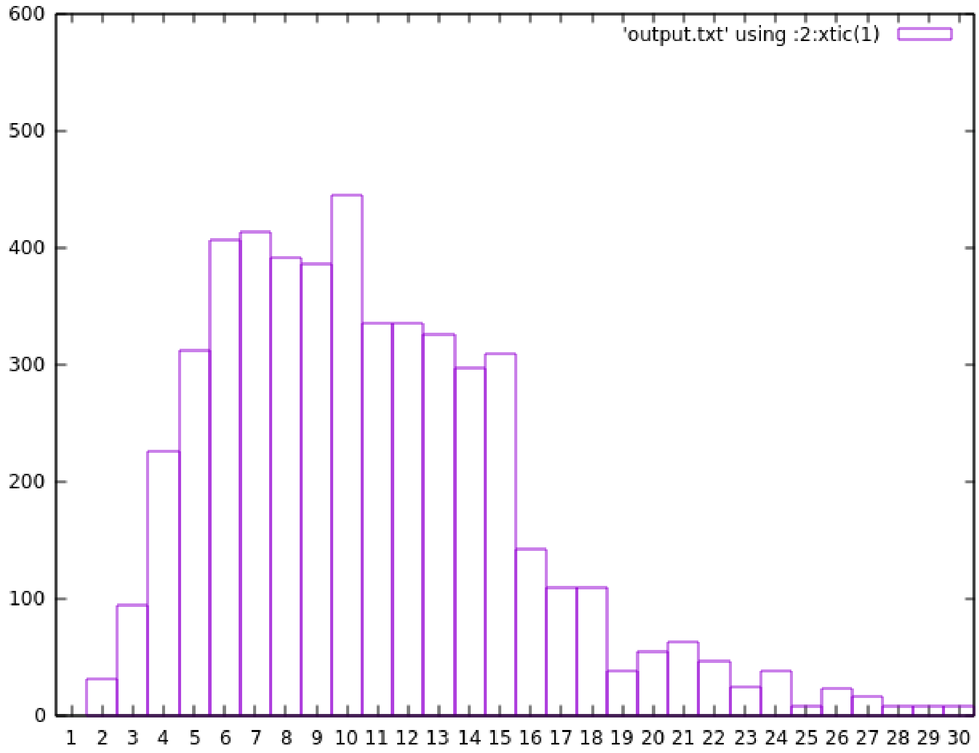
```
mean<-0
max<- -2147483646
min<- 2147483646
vector<-c()
for(i in 1:5000){
        u1<-runif(1)
        u2<-runif(1)
        u3<-runif(1)
        u4<-runif(1)
        u5<-runif(1)
        sample1<- -1*log(1-u1)/5
        sample2<- -1*log(1-u2)/5
        sample3<- -1*log(1-u3)/5
        sample4<- -1*log(1-u4)/5
        sample5<- -1*log(1-u5)/5
        sample =sample1+sample2+sample3+sample4+sample5
        vector[i]<-sample
        cat("Sample Values are: ",sample)
        cat("\n")
        mean<-mean+sample
        if(max<=sample)
                max<-sample
        if(min>=sample)
                min<-sample
}
png("question2.png")
hist(vector, breaks=50 , col="light cyan",plot=TRUE)
cat("Mean: ",mean/5000)
cat("\nMaximum: ",max)
cat("\nMinimum: ",min)
cat("\n")
```



- **C++ Code and Histogram :** Mean is : 1.021375 Maximum is :2.984813 Minimum is :0.162331

```
#include <stdio.h>
#include <iostream>
#include <stdlib.h>
#include <math.h>
#include <limits.h>
#include <algorithm>
int main(){
        long long int x;
        int a,i,k;
        double m,sample[25000],mean,maximum=INT_MIN ,minimum=INT_MAX,main_sample[5000];
        x=23;
        a=16807;
        m=44947;
        int count=25000;
        float u,j;
        for(i=0;i<count;i++){
                u=x/m;
                x=(x*a);
                x%=(long long int)m;
                sample[i]=((-1*log(1-u))/5);
        }
        for(i=0;i<5000;i++){
                main_sample[i]=sample[i]+sample[i+5000]+sample[i+10000]+sample[i+15000]+sample[

                if(main_sample[i]>maximum)
                        maximum=main_sample[i];
                if(main_sample[i]<minimum)
                        minimum=main_sample[i];
                mean+=(main_sample[i]/5000);
        }
        int frequency[(int)ceil(maximum*10)]={0};
        for(i=0;i<5000;i++){
                frequency[(int)ceil(main_sample[i]*10)-1]++;
        }
        for(i=0;i<(int)ceil(maximum*10);i++){
                printf("%d %d\n",i+1,frequency[i]);
        }
        printf("Mean is : %lf\n",mean);
        printf("Maximum is :%lf\n",maximum);
        printf("Minimum is :%lf\n",minimum);
}
```

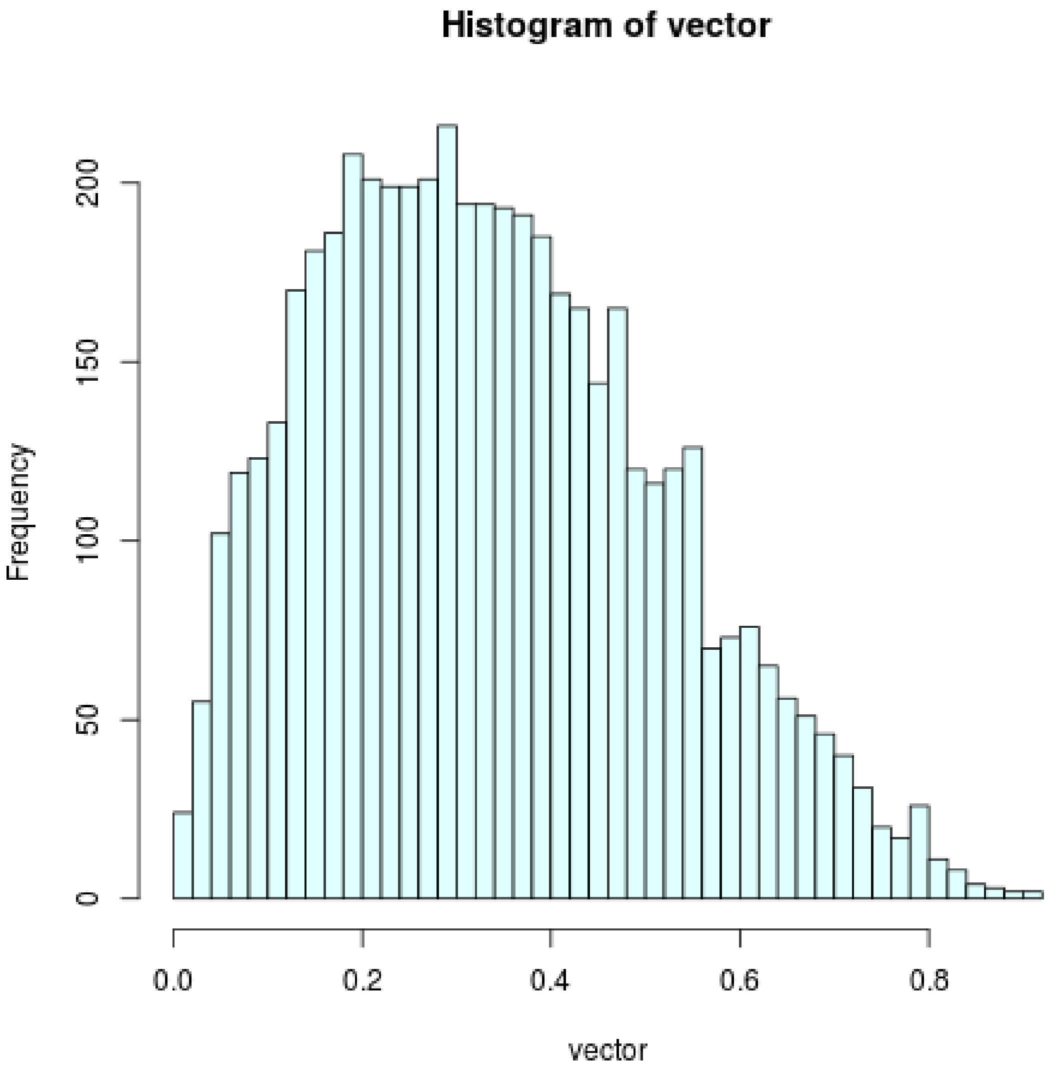


4 Question-3

4.1 Observation:

- By **"The rejection method"**, we can see that our approximation is almost correct as **we try to figure out a function covering** $f(x)$ **and is easily invertible in a sense**.The graph of the sample variables generated vs the frequency graphs is as follows.
- **R Code and Histogram :** Mean: 0.33657 Maximum: 0.9070434 Minimum: 0.003623385

```
count<-0
mean<-0
max<- -2147483646
min<- 2147483646
vector<-c()
while(1){
        u1<-runif(1)
        u2<-runif(1)
        if(u2<=(256*u1*(1-u1)**3)/27){
                cat("Sample Values are: ",u1)
                cat("\n")
                count<-count+1
                vector[count]<-u1
                mean<-(mean+u1)
                if(max<=u1)
                        max<-u1
                if(min>=u1)
                        min<-u1
        }
        if(count==5000)
                break
}
png("question3.png")
hist(vector, breaks=50 , col="light cyan",plot=TRUE)
cat("Mean: ",mean/5000)
cat("\nMaximum: ",max)
cat("\nMinimum: ",min)
cat("\n")
```


Histogram of vector
Frequency
0
50
100
150
200
0.0
0.2
0.4
0.6
0.8
vector