CUED - Engineering Tripos Part IIA 2018-2019 **Module Laboratory Report**

Student name:	Harvey Hughes	CRSID:	hh458	Module number:	3f3

Feedback to the student

☐ **See also comments in the text**

		Very good	**Good**	Needs improvmt
CONTENT	**Completeness, quantity of content:** Has the report covered all aspects of the lab? Has the analysis been carried out thoroughly?			
	Correctness, quality of content Is the data correct? Is the analysis of the data correct? Are the conclusions correct?			
	Depth of understanding, quality of discussion Does the report show a good technical understanding? Have all the relevant conclusions been drawn?			
	Comments:			
PRESENTATION	**Attention to detail, typesetting and typographical errors** Is the report free of typographical errors? Are the figures/tables/references presented professionally?			
	Comments:			

Raw report mark	**/ 5**
Penalty for lateness	

The weighting of comments is not intended to be equal, and the relative importance of criteria may vary between modules. A good report should attract 4 marks.

1 mark / week or part week.
Please use allowance forms to inform the teaching office about mitigating circumstances.

Marker: Date:

Random variables and random number generation

Harvey Hughes
Emmanuel College
Lab Date : 17/10/18
hh458@cam.ac.uk

October 23, 2018

Abstract

The purpose of this lab was to introduce random variables and how transformations can be used to generate a different probability density. This would be achieved using Matlab to back up theoretical results of random variables. Jacobian, inverse CDF and non linear transformations were investigated. The theory agreed with Matlabs random number generation throughout.

I. Introduction

During the course of the lab various methods and techniques were used to generate distributions of random variables using Matlab.
With the objectives being:

- Introduce the idea of random variables and functions of them
- To use the Jacobian to transform random variables
- To experiment with non uniform random number generation

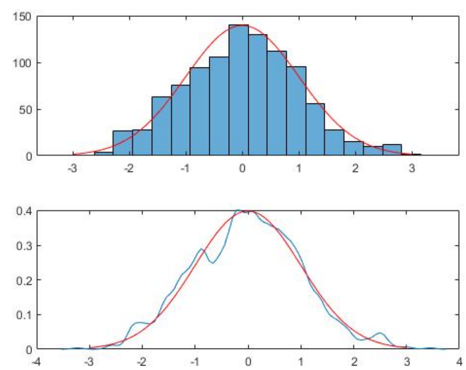

Figure 1: *Histogram and kernel density plot using a Gaussian kernel $\mathcal{N}(0,1)$ with sample size 1000*

II. Results and Discussion

i. 1. Uniform and normal random variables

When a Gaussian kernel is used, as in figure 1, it is clear that a kernel density method to approximate the distribution leads to a more accurate curve. This is due to the smoothing inherent in the kernel density method which removes the jaggedness that is shown in the histogram plot.

The opposite is true for distributions with a large discontinuity in density function such as a uniform distribution pictured in figure 2. This figure shows the smoothing present with the kernel method either side of the bounds [0,1]. This leads to an incorrect shape of den-

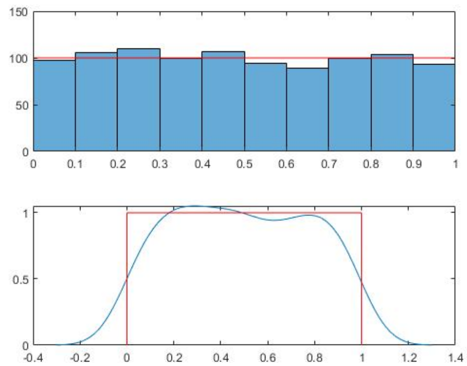

Figure 2: *Histogram and kernel density plot using a uniform distribution $\mathcal{U}(0,1)$ with sample size 1000*

sity function. The histogram plot more accurately depicts the distribution due to bin height tending towards the mean as N increases and therefore all bin heights being close to the pdf.

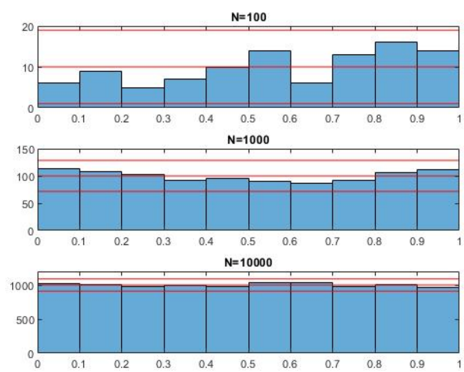

Figure 3: *Histogram plots using a uniform distribution $\mathcal{U}(0,1)$, showing how the theoretical mean and $\pm 3\sigma$ of bin height varies with sample size*

$$\begin{aligned} \mu &= Np_j \\ \sigma^2 &= Np_j(1-p_j) \end{aligned} \tag{1}$$

The theoretical mean and standard deviation of bin height in histogram plots can be calculated using equations 1. These values have been calculated and plotted in figure 3 for three different sample sizes from a uniform distribution. P_j is the probability of the distribution being located in one bin, for the plot mentioned $p_j = 0.1$ as 10 bins are present. Table 1 shows the values calculated. The results observed in figure 3 show that Matlab accurately generates uniform random variable that agree with the multinomial distribution theory which equation 1 is derived from. As sample size increases the 3σ lines are located far closer to the mean height. This agrees with the bin heights becoming far less varied and residing within the 3σ bounds. These bounds should contain about 99% of all possible bin heights.

Table 1: *Theoretical mean and standard deviation of bin height for a histogram plot of a uniformly distributed random variable*

Sample Size N	Mean	Standard Deviation
100	10	3
1000	100	$3\sqrt{10}$
10000	1000	30

The lines are relatively closer as the standard deviation increases with square root of N (but mean increases linearly with N)

i.1 Matlab code

For the Matlab code see section 1.0 of the appendix, all three graphs from this section were generated in the same Matlab script.

ii. 2. Function of random variables

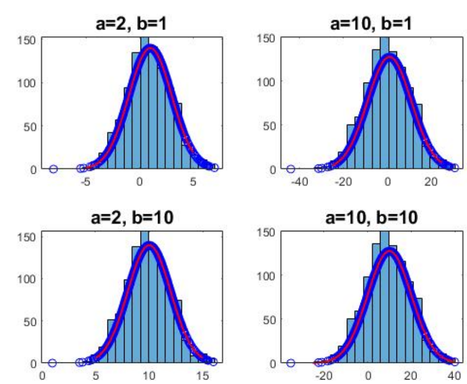

Figure 4: *Histogram plot of transforming a Gaussian kernel $\mathcal{N}(0,1)$ by $f(x) = ax + b$ with sample size 1000*

Starting with the distribution $X \sim \mathcal{N}(0,1)$ and performing the transformation Y=aX+b leads to the distribution plotted in figure 4. This is the distribution $Y \sim \mathcal{N}(a\mu + b, \sigma^2 a^2) = \mathcal{N}(b, a^2)$. The derivation for this can be seen in equations 2. Figure 4 shows the histogram data for a sample of 1000 variables in addition to each variable being transformed individually using the Jacobian to form the dotted distribution shown.

$$\begin{aligned}
&f(X) = aX + b \\
&J = |f'(x)| = |a| \\
&p(y) = \sum_1^1 \frac{p(x)}{J} = \frac{p(\frac{y-b}{a})}{|a|} \\
&p(y) = \frac{1}{a\sigma\sqrt{2\pi}} exp(-\frac{1}{2}[\frac{y-b-a\mu}{a\sigma}]^2) \\
&Y \sim \mathcal{N}(a\mu + b, a^2\sigma^2)
\end{aligned} \tag{2}$$

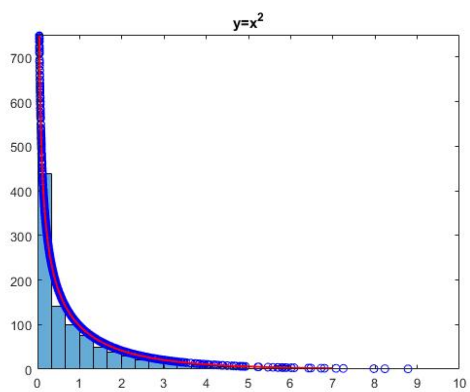

Figure 5: *Histogram plot of transforming a Gaussian kernel $\mathcal{N}(0,1)$ by $f(x) = x^2$ with sample size 1000*

The transformation $Y = X^2$ was applied to another Gaussian distribution following $X \sim \mathcal{N}(0,1)$. Figure 5 shows the result of this. The distribution $p(y) = \frac{1}{\sqrt{y2\pi}} exp(-\frac{1}{2}y)$ is overlain. This distribution was calculated using the same Jacobian method and is shown in equations 3. As $Y = X^2$ is not a 1:1 function the derivation required summing over all possible y values for a given x.

$$\begin{aligned}
&f(X) = X^2 \\
&J = |f'(x)| = 2|X| = 2\sqrt{y} \\
&p(y) = \sum_1^2 \frac{p(x)}{J} = \frac{p(+\sqrt{y}) + p(-\sqrt{y})}{2\sqrt{y}} \\
&p(y) = \frac{p(\sqrt{y}))}{\sqrt{y}} \; by\, symmetry \\
&p(y) = \frac{1}{\sigma\sqrt{y2\pi}} exp(-\frac{1}{2}[\frac{\sqrt{y}-\mu}{\sigma}]^2) \\
&\sigma = 1,\ \mu = 0
\end{aligned} \tag{3}$$

ii.1 Matlab code

The Matlab code for the transformation Y=aX+b can be seen in section 2.0 of the appendix, the transformation $Y = X^2$ is shown in section 2.1.

iii. 3. Inverse CDF method

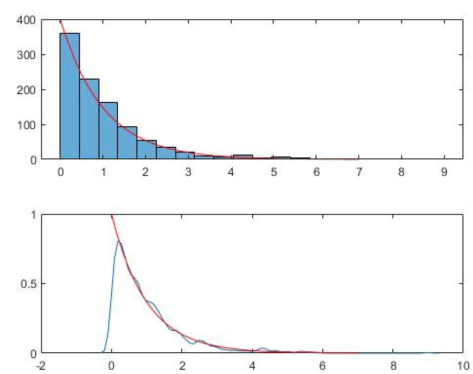

Figure 6: *Histogram and kernel density plot of an exponential distribution generated using the inverse CDF method on a uniform kernel $\mathcal{U}(0,1)$ with sample size 1000*

$$\begin{aligned}
&CDF = \int_0^y p(y)dy = \int_0^y e^{-y} \\
&CDF = 1 - e^{-y} = F(y) \\
&F^{-1}(y) = -ln(1-y)
\end{aligned} \tag{4}$$

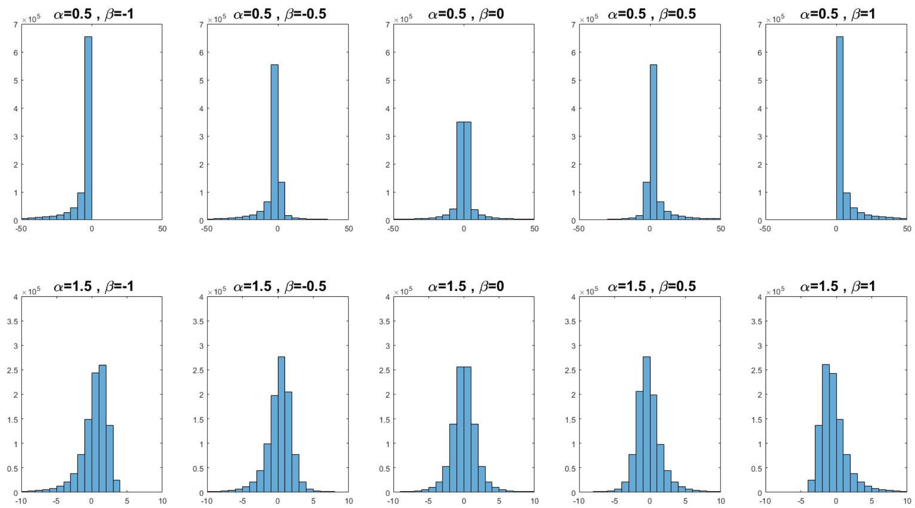

Figure 7: *Histogram plots of distribution X in equation 6 using various values for constants a and b, all from the same kernels $\mathcal{U}(-\frac{\pi}{2},\frac{\pi}{2})$ and $\mathcal{E}(1)$ with sample size 1000.*

The cumulative distribution function (CDF) for an exponential distribution p(y) = e^{-y} is calculated in equations 4. Though the use of the inverse CDF and equation 5 an exponentially distributed random variable can be generated using a uniform kernel $\mathcal{U}(0,1)$. This generated distribution is shown in figure 6 with p(y) overlain. The curves match up with great accuracy especially the kernel density plot. This shows the validity of this method in generating different random variable densities.

$$y^i = F^{-1}(x^i) \tag{5}$$

iii.1 Matlab code

The Matlab code for the inverse CDF method can be seen in section 3.0 of the appendix.

iv. 4. Simulation from a 'difficult' density

\alpha controls the tail thickness

By using the non linear transformation listed in equations 6 and various values for α and β the graphs in figure 7 can be produced. These graphs indicate that as α approaches 2 the distribution approaches normal, with a more rounded central peak. A low α such as 0.5 results in a very sharp density peak at 0. β appears to affect the skew of the distribution. A negative β results in negative skew and vice versa.

$$\begin{aligned}
&\alpha \in (0,2)\ \beta \in [-1,1] \\
&b = \frac{1}{\alpha} tan^{-1}(\beta tan(\frac{\pi\alpha}{2})) \\
&s = (1+\beta^2 tan^2(\frac{\pi\alpha}{2}))^{\frac{1}{2\alpha}} \\
&U \sim \mathcal{U}(-\frac{\pi}{2},\frac{\pi}{2})\ V \sim \mathcal{E}(1) \\
&X = s\frac{sin(\alpha(U+b))}{(cosU)^{1/\alpha}}
\end{aligned} \tag{6}$$

iv.1 Matlab code

The Matlab code for this transformation can be seen in section 4.0 of the appendix.

III. Appendix

1.0 Uniform and normal random variables

%Plot Normal Distributions
figure(1)

```
x=randn(1000,1);
subplot(211),
histogram(x,20)
hold on
n = [-3:.1:3];
norm = normpdf(n,0,1)*350;
plot(n,norm,'red')
hold off
subplot(212),
ksdensity(x,'width',0.1)
hold on
n = [-3:.1:3];
norm = normpdf(n,0,1);
plot(n,norm,'red')
hold off

%plot Uniform Distribution
figure(2)
y=rand(1000,1);
subplot(211),
histogram(y,10)
axis([0 1 0 150])
hold on
height = 100.* ones(length(n),1)
plot(n,height,'red')
hold off
subplot(212),
ksdensity(y,'width',0.1)
hold on
x1=[0,0,1,1];
y1=[0,1,1,0];
plot(x1,y1,'red')
hold off

%plot Uniform no.2 Distribution
N=[100,1000,10000]
height=[20,150,1200]
figure(3)
for i =1:3
y=rand(N(i),1);
subplot(3,1,i),
histogram(y,10)
title(['N=',num2str(N(i))])
hold on
mean=N(i)*0.1;
sd= sqrt(N(i)*0.1*0.9);
line([0,1],[mean,mean],'color','red')
line([0,1],[mean+3*sd,mean+3*sd],'color','red')
line([0,1],[mean-3*sd,mean-3*sd],'color','red')
axis([0 1 0 height(i)])
hold off
end
```

2.0 Transforming a Gaussian by Y=aX+b

```
x=randn(1000,1);
a=2;
b=1;
y=a*x+b;
J=abs(a);

py=0;
for i = 1:1 %only one possible value for x given y
py = normpdf((y-b)/a,0,1)/J ;
end

%Plot Transformed Distributions
figure(1)
subplot(221),
histogram(y,20)
hold on
scatter(y,py*700,'blue')
f = @(x) 1/(sqrt(2*pi)*a) *exp(-0.5*(x-b)^2/(a^2))
*700
fplot(f,[-5,5],'red','LineWidth',1.5)
hold off
title('\fontsize{14} a=2, b=1')

subplot(222),
a=10;
b=1;
y=a*x+b;
J=abs(a);
py=0;
for i = 1:1 %only one possible value for x given y
py = normpdf((y-b)/a,0,1)/J ;
end
histogram(y,20)
hold on
scatter(y,py*3200,'blue')
f = @(x) 1/(sqrt(2*pi)*a) *exp(-0.5*(x-b)^2/(a^2))
*3200
fplot(f,[-25,25],'red','LineWidth',1.5)
hold off
title('\fontsize{14} a=10, b=1')
```

```
subplot(223),
a=2;
b=10;
y=a*x+b;
J=abs(a);
py=0;
for i = 1:1 %only one possible value for x given y
py = normpdf((y-b)/a,0,1)/J ;
end
histogram(y,20)
hold on
scatter(y,py*700,'blue')
f = @(x) 1/(sqrt(2*pi)*a) *exp(-0.5*(x-b)2̂/(a2̂)) *700
fplot(f,[5,15],'red','LineWidth',1.5)
hold off
title('\fontsize{14} a=2, b=10')

subplot(224),
a=10;
b=10;
y=a*x+b;
J=abs(a);
py=0;
for i = 1:1 %only one possible value for x given y
py = normpdf((y-b)/a,0,1)/J ;
end
histogram(y,20)
hold on
scatter(y,py*3200,'blue')
f = @(x) 1/(sqrt(2*pi)*a) *exp(-0.5*(x-b)2̂/(a2̂)) *3200
fplot(f,[-25,40],'red','LineWidth',1.5)
hold off
title('\fontsize14 a=10, b=10')
```

2.1 Transforming a Gaussian by $Y = X^2$

```
x=randn(1000,1);
y=x.^2;
py=zeros(length(y),1);
for i = 1:2 %y=x^2 is a 2:1 function
J=abs(x)*2;
py = py + normpdf((-1)^i*sqrt(y),0,1)./J ;
end

%Plot Transformed distribution
figure(1)
histogram(y,40)
hold on
scatter(y,py*400,'blue')
f = @(x) 1/(sqrt(2*pi*x)) *exp(-0.5*(sqrt(x))^2) *400
fplot(f,[0,7],'red','LineWidth',1.5)
hold off
title('y=x^2')
axis([0 10 0 750])
```

3.0 Generating an exponential distribution using CDF method

```
x=rand(1000,1);
y= -log(-x+1);

figure(1)
subplot(211),
histogram(y,20);
hold on
f = @(x) 400*exp(-x)
fplot(f,[0,7],'red')
hold off
subplot(212),
ksdensity(y,'width',0.1)
hold on
f = @(x) exp(-x)
fplot(f,[0,7],'red')
hold off
```

4.0 Simulation from a difficult density

```
v=exprnd(1,1000,1);
u=rand(1000,1);
u=(u-0.5)*pi ;
alpha=0.5;
betavals=[-1,-0.5,0,0.5,1,-1,-0.5,0,0.5,1];
x=zeros(length(u),10);

figure(1)
histrange=50;
range=700;

for i = 1:10
beta = betavals(i);
if i>5
alpha = 1.5;
histrange=10;
range=400;
end
```

```
b=1/alpha * atan(beta*tan(pi*alpha/2)) ;
s=(1+beta^2*(tan(pi*alpha/2))^2)^(1/(2*alpha));
x(:,i)=s*   sin(alpha*(u+b))./(cos(u).^(1/alpha)).*
(cos(u-alpha*(u+b))./v).^((1-alpha)/alpha);
subplot(2,5,i),
histogram(x(:,i),[-histrange:histrange/10:histrange])
axis([-histrange histrange 0 range])
tit = strcat('\fontsize{18} \alpha=',num2str(alpha),
'\fontsize{18} , \beta=',num2str(beta));
title(tit)
end
```