UiO : **Matematisk institutt**

Det matematisk-naturvitenskapelige fakultet

STK-4051/9051 Computational Statistics Spring 2024 Ex6

Instructor: Odd Kolbjørnsen, oddkol@math.uio.no

Exercise 13 (Inverse transform sampling)
Assume $U \sim \text{Uniform}[0,1]$.

(a). Let $f(x)$ be a density function with corresponding cumulative distribution function $F(x)$ for which F^{-1} exists. Show that

$$X = F^{-1}(U) \sim f(x).$$

This way of simulation random variables is called the inverse transform sampling.

(a). We have

$$\begin{aligned}\Pr(X \leq x) &= \Pr(F^{-1}(U) \leq x)\\ &= \Pr(U \leq F(x)) = F(x)\end{aligned}$$

showing that X has cumulative distribution function $F(x)$.

(b). Assume $f(x)$ is the Cauchy distribution given by

$$f(x) = \frac{1}{\pi\gamma\left[1+\left(\frac{x-x_0}{\gamma}\right)^2\right]}, \quad F(x) = \frac{1}{\pi}\tan^{-1}\left(\frac{x-x_0}{\gamma}\right)+\frac{1}{2}$$

Use the inverse transform sampling method to simulate 1000 variables from the Cauchy distribution. Calculate the estimated density from your samples together with the true density and confirm that the method works.

Hint: The R command ***runif(1000)*** generate 1000 uniform variables. The R command ***density(x)*** can be used to estimate a density based on the (vector of) observations ***x***.

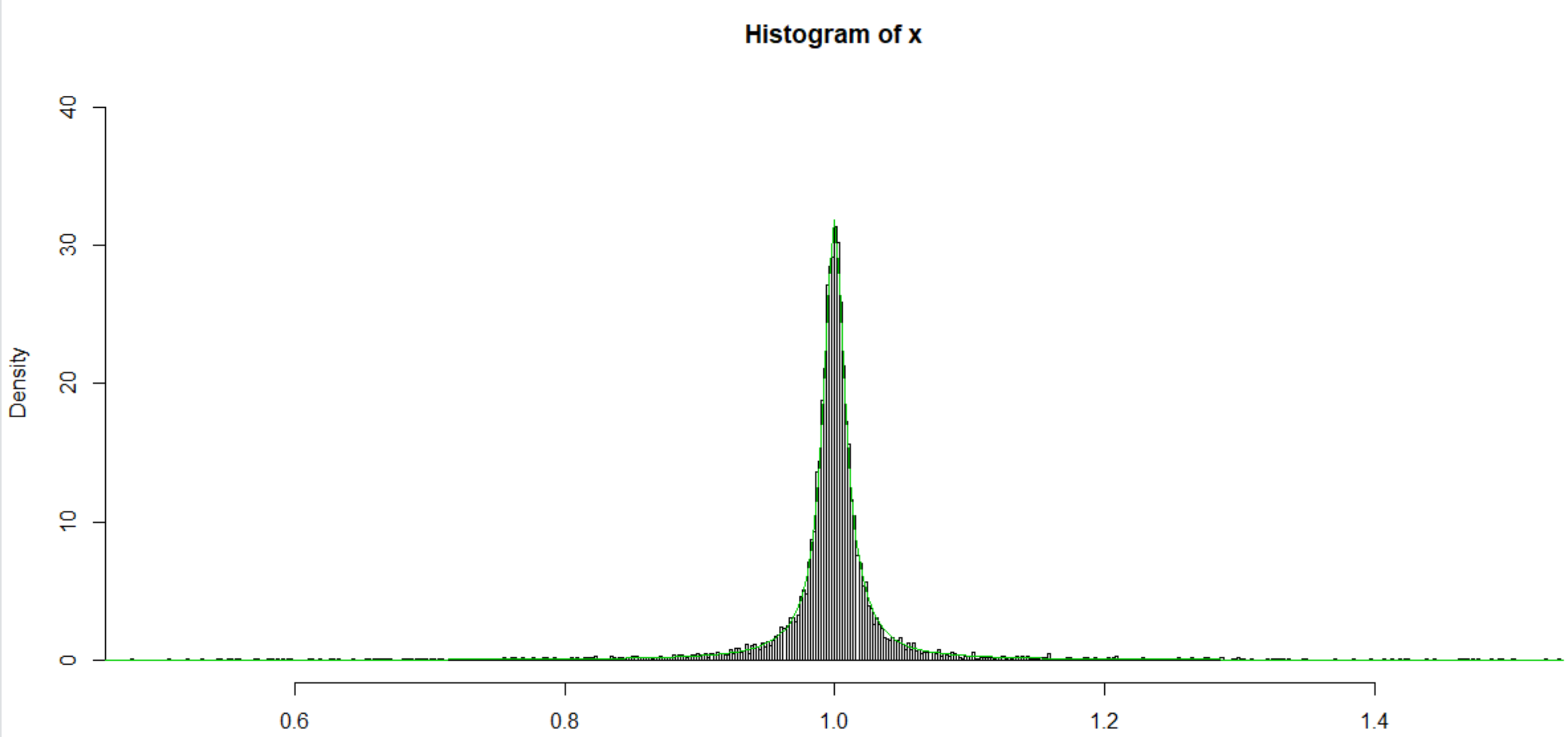
(b). We have

$$\begin{aligned} X =& F^{-1}(U) \\ &\Updownarrow \\ F(X) =& U \\ &\Updownarrow \\ \frac{1}{\pi}\tan^{-1}\left(\frac{X-x_0}{\gamma}\right)+\frac{1}{2} =& U \\ &\Updownarrow \\ X =& x_0 + \gamma\tan(\pi[U-\tfrac{1}{2}]) \end{aligned}$$

```
#Extra 13 b

u = runif(10000)
x0 = 1
gam = 0.01
x = x0 + gam*tan(pi*(u-0.5))
```

```
#Extra 13 b

u = runif(10000)
x0 = 1
gam = 0.01
x = x0 + gam*tan(pi*(u-0.5))
y = seq(-20,20,length=10001)

hist(x,100000,freq=F,xlim=c(0.5 ,1.5),ylim=c(0 ,40))
#lines(y,1/(pi*gam*(1+(y-x0)^2/gam^2)),col=2)
lines(y,dcauchy(y,x0,gam),col=3)
```

$$X = x_0 + \gamma \tan(\pi[U - \tfrac{1}{2}])$$



Exercise 14 (Generating variables from the normal distribution)
Recall that if $\boldsymbol{Y}$ is a random vector and $X = \boldsymbol{g}(\boldsymbol{Y})$ with $\boldsymbol{Y} = g^{-1}(\boldsymbol{X})$, then

$$f_X(\boldsymbol{x}) = f_Y(\boldsymbol{g}^{-1}(\boldsymbol{x}))\left|\frac{\partial}{\partial \boldsymbol{x}} g^{-1}(\boldsymbol{x})\right|$$

(a). Assume $\boldsymbol{U} = (U_1, U_2)$ where U_1 and U_2 are independent and Unif[0, 1]. Define

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2), \quad X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

Show that

$$U_1 = e^{-0.5(X_1^2+X_2^2)}$$
$$U_2 = \frac{1}{2\pi}\tan^{-1}(X_2/X_1)$$

(b). Show that X_1, X_2 are independent and $N(0, 1)$.

(c). Show how you then can generate from a general $N(\mu, \sigma^2)$ distribution.

(d). Assume now you want to simulate from a multivariate distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. How can you do that?

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2), \quad X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

Show that

$$U_1 = e^{-0.5(X_1^2+X_2^2)}$$
$$U_2 = \frac{1}{2\pi}\tan^{-1}(X_2/X_1)$$

We have

$$\begin{aligned}
\exp(-0.5(X_1^2+X_2^2)) &= \exp(-0.5(-2\log(U_1)\cos^2(2\pi U_2) - 2\log(U_1)\sin^2(2\pi U_2))) \\
&= \exp(-0.5(-2\log(U_1))) = U_1 \\
\frac{1}{2\pi}\tan^{-1}(X_2/X_1) &= \frac{1}{2\pi}\tan^{-1}(\sin(2\pi U_2)/\cos(2\pi U_2)) \\
&= \frac{1}{2\pi}\tan^{-1}(\tan(2\pi U_2)) = U_2
\end{aligned}$$

Recall that if $\boldsymbol{Y}$ is a random vector and $X = \boldsymbol{g}(\boldsymbol{Y})$ with $\boldsymbol{Y} = g^{-1}(\boldsymbol{X})$,

$$f_X(\boldsymbol{x}) = f_Y(\boldsymbol{g}^{-1}(\boldsymbol{x}))\left|\frac{\partial}{\partial \boldsymbol{x}} g^{-1}(\boldsymbol{x})\right|$$

$$X_1 = \sqrt{-2\log(U_1)}\cos(2\pi U_2), \quad X_2 = \sqrt{-2\log(U_1)}\sin(2\pi U_2)$$

Show that

$$\left.\begin{aligned} U_1 &= e^{-0.5(X_1^2+X_2^2)} \\ U_2 &= \frac{1}{2\pi}\tan^{-1}(X_2/X_1) \end{aligned}\right\} g^{-1}(\boldsymbol{x})$$

(b). We have that

$$\begin{aligned} f_X(\boldsymbol{x}) =& f_U(\boldsymbol{g}^{-1}(\boldsymbol{x}))\left|\frac{\partial}{\partial \boldsymbol{x}} g^{-1}(\boldsymbol{x})\right| \\ =& \begin{vmatrix} -X_1\exp(-0.5(X_1^2+X_2^2)) & -X_2\exp(-0.5(X_1^2+X_2^2)) \\ \frac{1}{2\pi}\frac{-X_2/X_1^2}{1+X_2^2/X_1^2} & \frac{1}{2\pi}\frac{1/X_1}{1+X_2^2/X_1^2} \end{vmatrix} \\ =& \frac{1}{2\pi}\exp(-0.5(X_1^2+X_2^2))\begin{vmatrix} -X_1 & -X_2 \\ \frac{-X_2}{X_1^2+X_2^2} & \frac{X_1}{X_1^2+X_2^2} \end{vmatrix} \end{aligned}$$

Common factor for a row can be set infront

$$\left|-X_1 \cdot \frac{X_1}{X_1^2+X_2^2} - (-X_2)\cdot\frac{-X_2}{X_1^2+X_2^2}\right| = 1$$

$$= \frac{1}{\sqrt{2\pi}}\exp(-0.5X_1^2))\frac{1}{\sqrt{2\pi}}\exp(-0.5X_2^2))$$

(c). Show how you then can generate from a general $N(\mu, \sigma^2)$ distribution.

(d). Assume now you want to simulate from a multivariate distribution $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. How can you do that?

(c). Use that $X = \mu + \sigma Z$ where Z is standard Gaussian.

(d). Assume $\boldsymbol{Z}$ is a vector of independent standard Gaussian variables. Let $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}^{1/2}(\boldsymbol{\Sigma}^{1/2})^T$ (e.g. the Cholesky decomposition). Then define $X = \boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}$. The vector is Gaussian since it is a linear combination of Gaussians, and

$$
\begin{aligned}
E[X] =& E[\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}] = \mu \\
\text{Var}[X] =& \text{Var}[\boldsymbol{\mu} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{Z}] = \boldsymbol{\Sigma}^{1/2}\boldsymbol{I}(\boldsymbol{\Sigma}^{1/2})^T = \boldsymbol{\Sigma}
\end{aligned}
$$

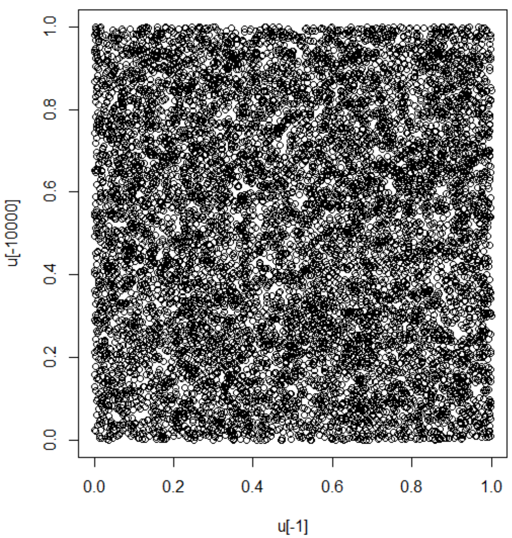
Exercise 15 (Checking random generators)
Random generators on computers are never completely random. Typically, one generated number is a function of the previous (or several previous) numbers.

There are many kinds of tests (formal and visual) for checking whether a random generator is good. One such is to test for independence between successive generated numbers.

Generate 10 000 variables $U_1, ..., U_{10000}$ from the uniform distribution. Plot U_i against U_{i+1}. Is there any indication of dependence?

Hint: If `u` contains the 10 000 variables, then `u[-1]` will contain the same data except the first one, while `u[-10000]` will contain the same data except the 10 000th one.

```
#Checking random generators
## EX 1
u = runif(10000)
mean(u)-0.5
var(u)*12-1
hist(u[-1],50)
plot(u[-1],u[-10000])
cor(u[-1],u[-10000])
```

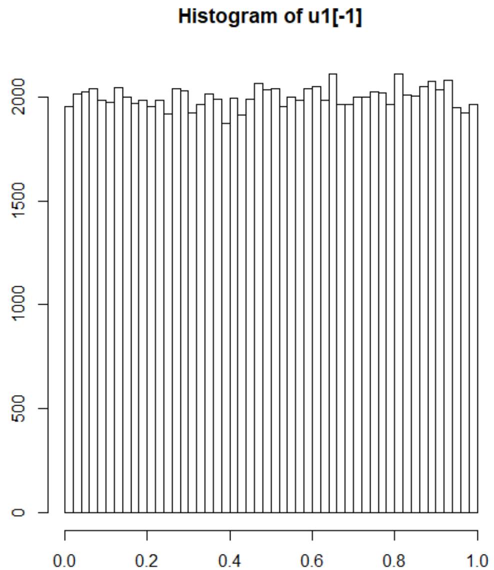


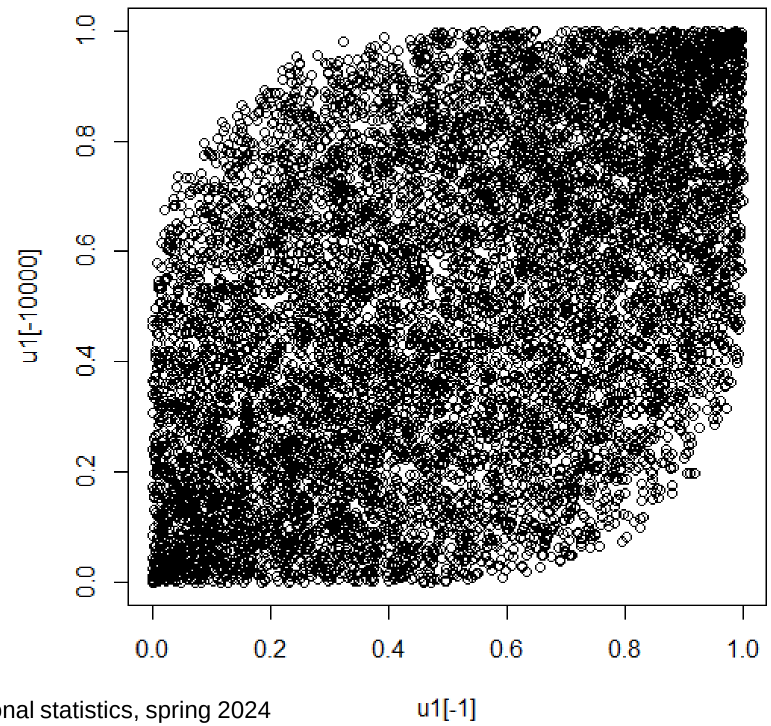
Example of Correlation

```
# creating correlation
u0 = runif(100001)
u1 = u0[-1]+u0[-100001]
u1[u1<1] = u1[u1<1]^2/2
u1[u1>=1] = 1-(2-u1[u1>=1])^2/2
## EX 2
mean(u1)-0.5
var(u1)*12-1
hist(u1[-1],50)
plot(u1[-1],u1[-10000])
cor(u1[-1],u1[-10000])
```

Corelation

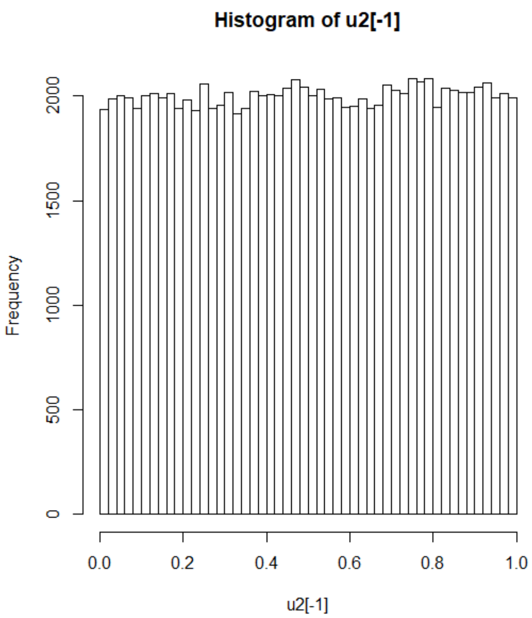
Transform to uniform

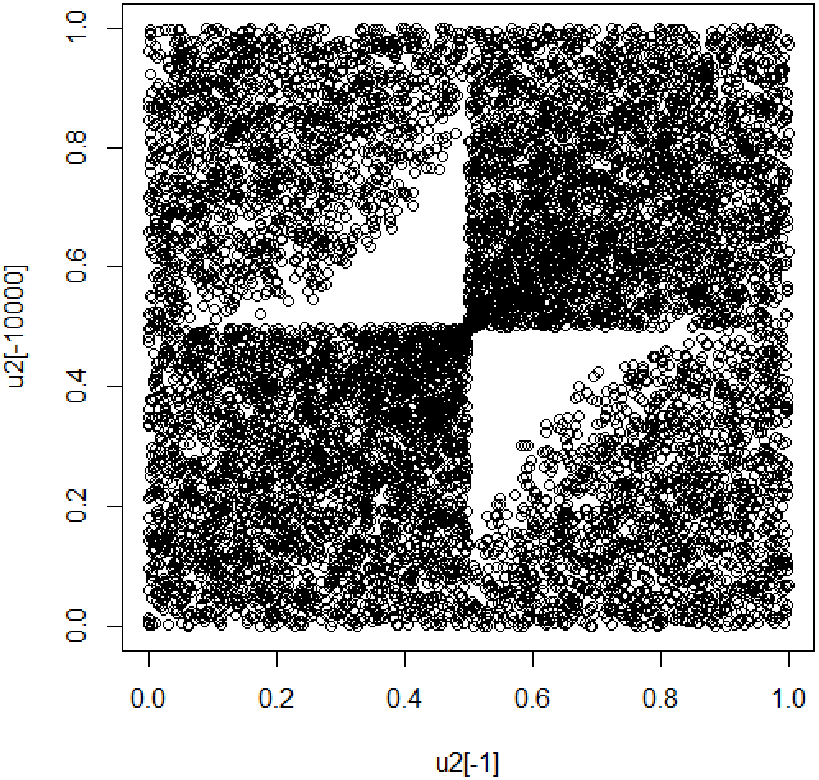




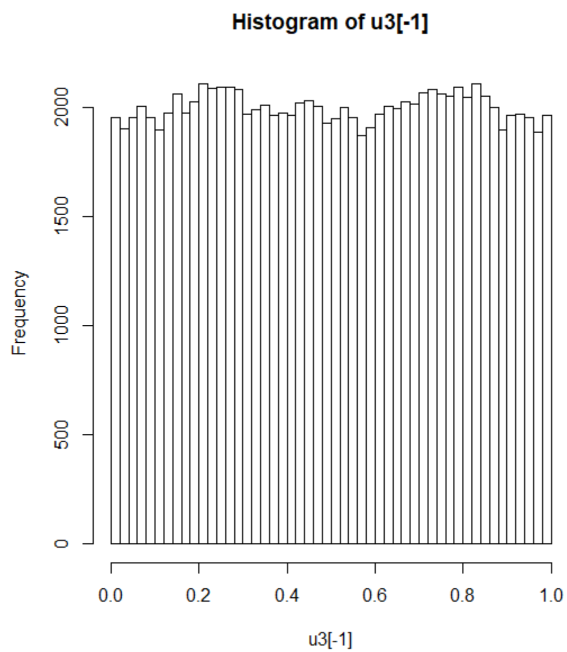
Example of Correlation

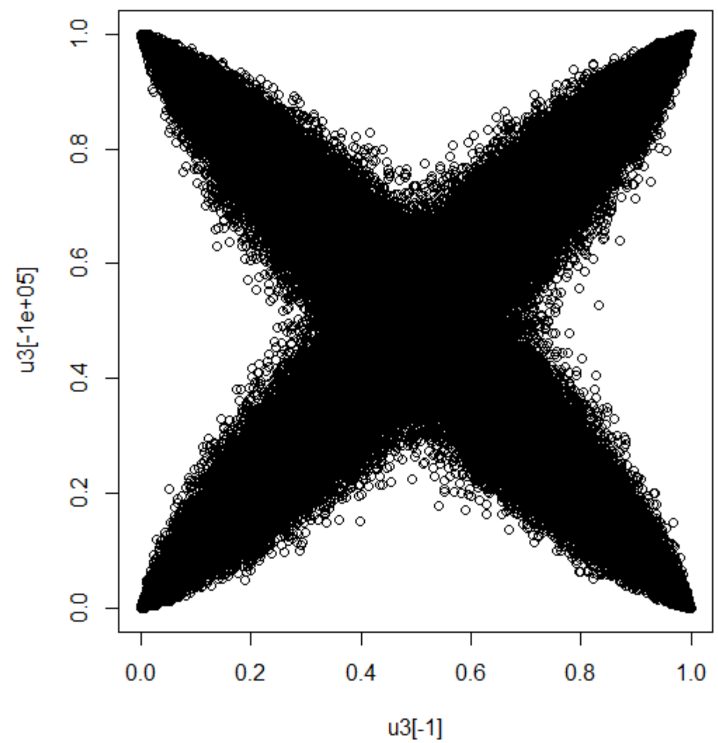
```
## EX 3
u2=u1
u2[u1<0.5]=0.5-u1[u1<0.5]
u2[u1>=0.5]=1.5-u1[u1>=0.5]

mean(u2)-0.5
var(u2)*12-1
hist(u2[-1],50)

plot(u2[-1],u2[-10000])
cor(u2[-1],u2[-10000])
```





```
alternating = rep(c(1,-1,-1,1),25000)
rho=0.99
x=rep(0,100000)
x[1]=rnorm(1)
for( i in 2:100000)
{
  x[i]=rho*x[i-1]+sqrt(1-rho^2)*rnorm(1)
}
u3=as.numeric(alternating<0 )+alternating*pnorm(x)
```





Exercise 16 (Calculation by Monte Carlo)

(a). Simulate 10000 samples from the $N(2,1)$ distribution and store them in a vector called `z`.

Hint: The R command `rnorm(m,2,1)` simulates `m` random variables from the $N(2,1)$-distribution.

```
#Exercise 16
n = 10000

par(mfrow=c(2,2))
#a
print("Normal")
z = rnorm(n,2,1)
```

(*b*). Plot the estimated density and the true density in the same figure.

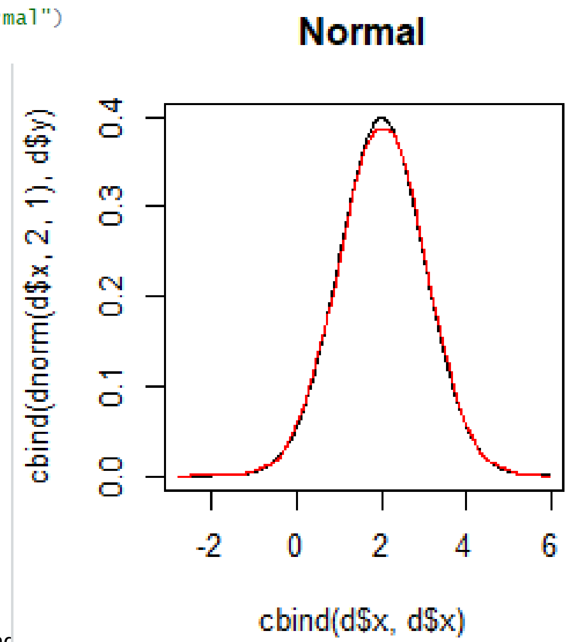
Hint: The R command

```
> d <- density(z)
```

will estimate the density by a non-parametric kernel method. `d` will be a list containing two components with names `x` and `y`. `d$x` contain the values at which the density is estimated, while `d$y` are the corresponding estimates. Plot both curves in the same figure by

```
> matplot(cbind(d$x,d$x),cbind(dnorm(d$x,2,1),d$y),type="l")
```

```
#b
d = density(z)
matplot(cbind(d$x,d$x),cbind(dnorm(d$x,2,1),d$y),type="l",lty=1,main="Normal")
```



(*c*). Estimate mean and variance by standard estimates from the sample. Compare with the true values.

(*d*). Assume now $Z \sim N(2, 1)$. For each of the two sub-exercises below, measure the time you spend on the problem:

(*i*) Find $\nu = E[Z|Z > 3.0]$ trough analytical calculation.

(*ii*) Find $\nu = E[Z|Z > 3.0]$ through simulation.

How many observations was the simulated estimate based on?

Hint: The R command `trunc <- z[z>3.0]` picks the set of z's larger than 3.0 from the vector `z`. The length of the vector `trunc` can be found by the command `length(trunc)`.

```
#c
print("mean and sd")
show(c(mean(z),sd(z)))

#d
v = z[z>3]
print("True and estimated conditional mean")
show(c(2+exp(-0.5)/(sqrt(2*pi)*(1-pnorm(1))),mean(v)))
show(length(v))
```

```
> show(c(2+exp(-0.5)/(sqrt(2*pi)*(1-pnorm(1))),mean(v)))
[1] 3.525135 3.519874
> show(length(v))
[1] 1551
```

Theory

(i) Find $\nu = E[Z|Z > 3.0]$ trough analytical calculation.

(d). Define $Z = X + 2$ were $X \sim N(0,1)$. Then

$$E[Z|Z>3] = E[X+2|X+2>3] = 2 + E[X|X>1]$$
$$= 2 + \int_1^\infty x \frac{\frac{1}{\sqrt{2\pi}} e^{-0.5x^2}}{\Pr(X>1)} dx$$
$$= 2 + \frac{[-\frac{1}{\sqrt{2\pi}} e^{-0.5x^2}]_1^\infty}{1-\Pr(X\leq 1)} = 2 + \frac{\frac{1}{\sqrt{2\pi}} e^{-0.5}}{1-\Pr(X\leq 1)}$$

(e). For exponential we have the forgetting property so that given that $Z > 3$ we still have that it is exponential, so that

$$E[Z|Z>3] = 3 + E[Z_0] = 3 + 0.5$$

(e). Repeat (a) – (d) with the following distributions: Exp(2), T-distribution with 4 degrees of freedom and Gamma(2) (calculate ν analytically only when it is not too difficult).

```
#e1
print("Exponential")
z = rexp(n,2)
d = density(z,from=0)
matplot(cbind(d$x,d$x),cbind(dexp(d$x,2),d$y),type="l"
print("mean and sd")
show(c(mean(z),sd(z)))
v = z[z>3]
print("True and estimated conditional mean")
show(c(3.5,mean(v)))
show(length(v))

#e2
print("T-distribution")
z = rt(n,4)
d = density(z)
matplot(cbind(d$x,d$x),cbind(dt(d$x,4),d$y),type="l",l
print("mean and sd")
show(c(mean(z),sd(z)))
v = z[z>3]
print("Estimated conditional mean")
show(mean(v))
show(length(v))

#e4
print("Gamma")
z = rgamma(n,2,1)
d = density(z,from=0)
matplot(cbind(d$x,d$x),cbind(dgamma(d$x,2,1),d$y),type
show(c(mean(z),sd(z)))
print("Estimated conditional mean")
v = z[z>3]
show(mean(v))
show(length(v))
```