Project 1 for Bayesian Inference
Spring 2009
Name Surname

Question 1 Generate a sample of length 1000 from a $N(0,1)$ distribution, making use of the Box-Muller technique. Use graphical methods (e.g. histogram) to convince yourselves that you get what you want. Estimate the sample mean and the sample variance. Do a q-q plot.

Answer The Box-Muller technique is is based on the observation that if (X, Y) is a normal random variable in $\mathbb{R}^2$ then its representation (R, Θ) in polar coordinates is such that R^2 is exponential with mean 1/2 and Θ uniform on $(0, 2\pi)$.

Now, if U, V are i.i.d. uniform on $(0, 1)$ then $R^2 = -2 \log U$ is exponential with mean 1/2, and $\Theta = 2\pi V$ is uniform on $(0, 2\pi)$.

We first n independent samples from U and n from V:

```
n=10000; u=runif(n); v=runif(n)
```

We then compute R^2 and Θ:

```
rsquared=-2*log(u); theta=2*pi*v
```

Finally, we generate samples for (X, Y) by using the formula $X = R \cos \Theta$, $Y = R \sin \Theta$:

```
r=sqrt(rsquared); x=r*cos(theta); y=r*sin(theta)
```

We can concatenate x and y to obtain a size $2n$ sample:

```
z=c(x,y)
```

We estimate the mean and the variance

```
mean(z); var(z)
```

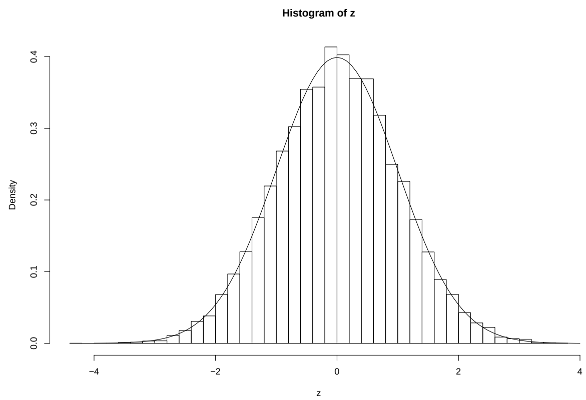
The answer is: 0.002601275 and 1.000343, respectively.

We generate a histogram for z and compare it against the theoretical density

$$f(z) = (2\pi)^{-1/2} \exp(-z^2/2)$$

as follows:

```
hist(z,breaks=50,probability=1)
f=function(x){(2*pi)^(-1/2)*exp(-x^2/2)}
plot(f,-4,4,add=TRUE)
dev.copy(postscript,'plot1.ps')
dev.off()
```



We obtain a good much and we're happy.

Define now

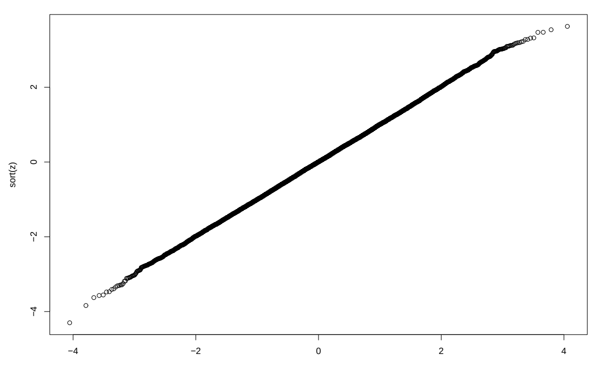
$$F(z) = \int_{-\infty}^{z} f(t)dt,$$

and its inverse function

$$F^{-1}(p) = \inf\{z \in \mathbb{R} : \ F(z) > p\} = \inf\{z \in \mathbb{R} : \ F(z) = p\}.$$

The function F^{-1} is called quantile function. The formula for $F^{-1}(p)$ in R is `rnorm(p)`. We have a sample z of size $2n$ and want to check if it comes from F. We generate $2n$ equally spaced points in the interval $(0, 1)$, store them in a vector p and plot $F^{-1}(p)$ against z:

```
p = ((1:(2*n))-0.5)/(2*n)
plot(qnorm(p),sort(z))
dev.copy(postscript,'plot2.ps')
dev.off()
```



We obtain a rather straight line and we're happy.