MATH3871/MATH5960 Bayesian Inference and Computation, Lab 2 Exercises

Example 0.1 (Generating Standard Normal Random Variables) If X and Y independent standard normally distributed random variables, then their joint pdf is

$$f(x,y) = \frac{1}{2\pi} \mathrm{e}^{-\frac{1}{2}x^2+y^2}, (x,y) \in \mathbb{R}^2,$$

which is a radially symmetric function. We can show that in polar coordinates, the random angle Θ of the point (X,Y) is $\mathsf{Unif}(0, 2\pi)$ distributed (as expected by the radial symmetry), and the radius R has pdf $f_R(r) = r\mathrm{e}^{-r^2/2}, r > 0$. Moreover, R and Θ are independent. In the tutorial we saw that R has the same distribution as $\sqrt{-2\ln U}$ with $U \sim \mathsf{Unif}(0,1)$. So the idea is to first simulate R and Θ independently, and then return $X = R\cos(\Theta)$ and $Y = R\sin(\Theta)$ as a pair of independent standard normal random variables. This leads to the following method for generating standard normal random variables.

Algorithm 0.0.1: Normal Random Variable Generation: Box–Muller Approach

---

output: Independent standard normal random variables X and Y.

Generate two independent random variables, U_1 and U_2, from $\mathsf{Unif}(0,1)$.

$X \leftarrow (-2\ln U_1)^{1/2} \cos(2\pi U_2)$

$Y \leftarrow (-2\ln U_1)^{1/2} \sin(2\pi U_2)$

return X, Y

In R and MATLAB we can generate standard normal random variables via the `rnorm()` and `randn()` functions, respectively. Once a standard normal number generator is available, simulation from any multivariate normal distribution $\mathsf{N}(\boldsymbol{\mu}, \Sigma)$ is relatively straightforward. Given a covariance matrix $\Sigma = (\sigma_{ij})$, there exists a unique lower triangular matrix $\mathbf{B}$

$$\mathbf{B} = \begin{pmatrix} b_{11} & 0 & \cdots & 0 \\ b_{21} & b_{22} & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{pmatrix} \tag{1}$$

such that $\Sigma = \mathbf{B}\mathbf{B}^\top$. In fact there exist many of such decompositions. One of the more important ones is the *Cholesky decomposition*, which is a special case of a LU decomposition; In MATLAB, the function `chol` can be used to produce such a matrix $\mathbf{B}$. CHOLESKY DECOMPOSITION

> Note that it is important to use the option `'lower'` when calling this function, as MATLAB produces an upper triangular matrix by default.

Once the Cholesky factorization is determined, it is easy to simulate $\boldsymbol{X} \sim \mathsf{N}(\boldsymbol{u}, \Sigma)$ as, by definition, it is the affine transformation $\boldsymbol{\mu} + \mathbf{B}\boldsymbol{Z}$ of an n-dimensional standard normal random vector.

Algorithm 0.0.2: Normal Random Vector Generation

---

input : $\boldsymbol{\mu}, \Sigma$

output: $\boldsymbol{X} \sim \mathsf{N}(\boldsymbol{\mu}, \Sigma)$

Determine the lower Cholesky factorization $\Sigma = \mathbf{B}\mathbf{B}^\top$

Generate $\boldsymbol{Z} = (Z_1, \ldots, Z_n)^\top$ by drawing $Z_1, \ldots, Z_n \sim_{\text{iid}} \mathsf{N}(0,1)$

Set $\boldsymbol{X} \leftarrow \boldsymbol{\mu} + \mathbf{B}\boldsymbol{Z}$.

■ **Example 0.2 (Generating from a Bivariate Normal Distribution)** The MATLAB code below draws 1000 samples from two normal pdfs: one with covariance matrix $\Sigma = \mathbf{I}$ and the other with covoariance matrix $\Sigma = [1, \varrho; \varrho, 1]$. The resulting point clouds are given in Figure 1.

```
% bivnorm.m
N = 1000; rho = 0.8;  %change to rho = 0 for the other plot
Sigma = [1 rho; rho 1];
B=chol(Sigma,'lower');
x=B*randn(2,N);

plot(x(1,:),x(2,:),'.')
```

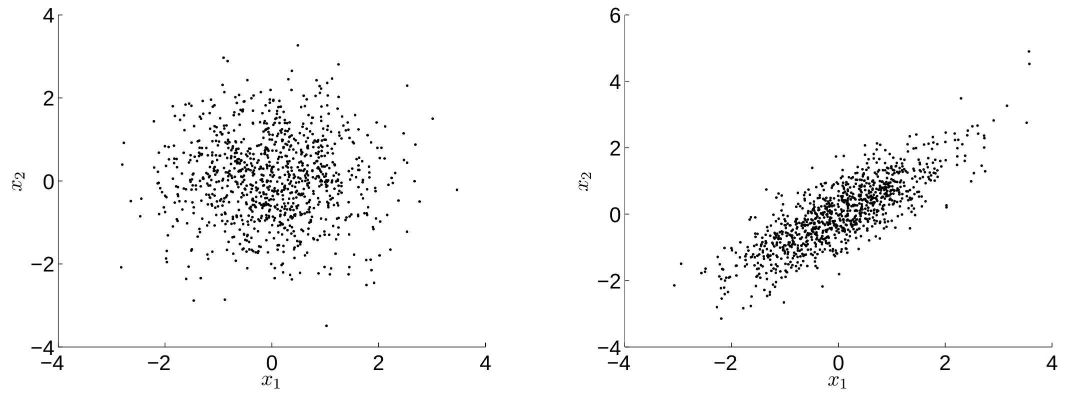

Figure 1: 1000 realizations of bivariate normal distributions with means zero, variances 1, and correlation coefficients 0 (left) and 0.8 (right).

Now, repeat this in `R`. You can use the resources procided in Moodle.