1. Eigenvalue algorithm

Given a symmetric matrix $A \in \mathbb{R}^{n \times n}$ with n different eigenvalues, write a program to compute its eigenvalues and eigenvectors in Armadillo. Equivalently, approximately find its eigendecomposition:

$$A = \sum_{i=1}^{n} \lambda_i u_i u_i^T, \quad (\lambda_i \neq \lambda_j \, , \forall i \neq j)$$

You can use **power iteration algorithm** as follows:

Starts with a vector b_0. The method is described by the recurrence relation

$$b_{k+1} = \frac{Ab_k}{\|Ab_k\|}$$

$$\mu_k = b_k^T A b_k$$

Under some conditions, μ_k, b_k converges to the dominant eigenvalue and corresponding eigenvector.

Check your algorithm by generating a 3×3 random matrix A with standard normal entries and then set $A = A + A^T$. Compare your result with `eig_sym()` in Armadillo.

2. Generating gaussian distribution

(a)

Write a program to generate standard normal sample using uniform random sample generator(for example, `randu()` in Armadillo or `uniform_dist` in trng). This can be done by following **Box–Muller transform**:

Given U_1 and U_2 independently from $\mathrm{Uniform}(0, 1)$.

Consider polar coordinates (R, Θ) for independent standard normal (Z_0, Z_1). we know $R^2 \sim \chi_2^2$ and $\Theta \sim \mathrm{Uniform}(0, 2\pi)$, thus generate R and Θ by:

$$R = \sqrt{-2 \ln U_1}$$
$$\Theta = 2\pi U_2$$

Then independent standard normal (Z_0, Z_1) can be obtained by

$$Z_0 = R\cos(\Theta) = \sqrt{-2 \ln U_1} \cos(2\pi U_2)$$
$$Z_1 = R\sin(\Theta) = \sqrt{-2 \ln U_1} \sin(2\pi U_2)$$

(b)

Why your generated Z_0 and Z_1 are actually truncated? (hint: for a computer, there is a smallest non-zero number)