1 Generating a bivariate normal

Let Z_1 and Z_2 be independent standard normal random variables.

Define,

$$X_1 = \mu_1 + \sigma_1 Z_1 \qquad \text{and } X_2 = \mu_2 + \sigma_2(\rho Z_1 + \sqrt{1-\rho^2} Z_2)$$

Show (X_1, X_2) is bivariate normal, $\mathcal{N}_2(\mu, \Sigma)$, with

$$\mu = (\mu_1, \mu_2) \quad \text{and } \Sigma = \begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho \\ \sigma_1\sigma_2\rho & \sigma_2^2 \end{pmatrix}.$$

You already know how to generate standard normals in R using `rnorm` so the above will enable you to simulate from a bivariate normal.

2 Rejection sampling

Read section "Rejection" on page 79. Basic idea is to use an "easy to simulate" distribution to generate samples from a "hard to simulate" distribution.

Example - Uniform on circle

We want to generate a sample drawn uniformly on a circle of radius 1.
Use a uniform on a square with corners $(1,1), (1,-1), (-1,-1), (-1,1)$.
Procedure:

1. Draw u_1, u_2 independently from a Uniform(-1,1).
2. If $u_1^2 + u_2^2 <= 1$ keep (u_1, u_2) otherwise repeat step 1.

```
# keep running following until see a sample
u <- runif(2, min = -1, max = 1)
# introduce if statement
if (u[1]^2 + u[2]^2 <= 1){print(u)}

# I really want, say, 1000 samples

# introduce while statement
sample.size <- 1    # keep track of samples so far
sample <- matrix(numeric(2 * 1000), ncol = 2)   # to hold samples
while (sample.size <= 1000){
    u <- runif(2, min = -1, max = 1)
    if (u[1]^2 + u[2]^2 <= 1){
        sample[sample.size, ] <- u
        sample.size <- sample.size + 1
        }
}

head(sample)
dim(sample)
plot(sample[,1], sample[,2])
```

For Lab question 2 you need to pick two points from a uniform on a sphere. Each point is 3 dimensional you need a total of $2 \times n$ points (you choose n, probably want it big).

For each pair find the distance between them. Then use this to produce as estimate of E(distance between two points).

3 Maximum Likelihood

Example Have Y_i, $i = 1, \ldots, n$ iid Poisson(θ). Then,

$$L(\theta) = \prod_{i=1}^{n} \theta^{y_i} \frac{\exp(-\theta)}{y_i!}$$

$$l(\theta) = \log L(\theta) = \sum_{i=1}^{n} y_i \log \theta - \theta - \log y_i!$$

```
# likelihood for poisson
pois.lhood <- function(theta, y){
  prod(theta ^ y * exp(-theta) / factorial(y))
}

# some fictional data
y <- c(5, 0, 6, 3, 7, 5, 5, 7, 3, 5)

# want to plot likelihood for theta
theta <- seq(0, 10, 0.1)
l.at.theta <- sapply(theta, pois.lhood, y = y)

par(mfrow = c(2,1))
plot(theta, l.at.theta, type = "l", ylab = "Likelihood")

# log-likelihood for poisson
pois.llhood <- function(theta, y){
  sum(log(theta) * y - theta - log(factorial(y)))
}

ll.at.theta <- sapply(theta, pois.llhood, y = y)
plot(theta, ll.at.theta, type = "l", ylab = "Log likelihood")

# maximise by numerical methods
# use function nlm - but this is a minimiser so we minimise negative log-likelihood
neg.pois.llhood <- function(theta, y){
  -pois.llhood(theta, y)
}

nlm(neg.pois.llhood, p = 5, y = y)
# MLE estiamte of theta is 4.6
abline(v = nlm(neg.pois.llhood, p = 5, y = y)$estimate)
```