Northeastern University
College of Science

Module 4 Homework

Problem 1 (25 points)

Suppose that for certain microRNA of size 20 the probability of a purine is binomially distributed with probability 0.7. Say there are 100 such microRNAs, each independent of the other.

Let Y denote the average number of purine in these microRNAs. Find the probability that Y is greater than 15. Please give a theoretical calculation, do NOT use Monte Carlo simulation to approximate. Show all the steps and formulas in your calculation.

ANSWER:
CODE

```
microRNA_size <- 20
purine_probability <- 0.7
sample_size <- 100

mu <- microRNA_size * purine_probability
sigma <- sqrt(microRNA_size * purine_probability * (1 - purine_probability))

mu_Y <- mu
sigma_Y <- sigma / sqrt(sample_size)

z <- (15 - mu_Y) / sigma_Y

probability_Y_gt_15 <- 1 - pnorm(z)

cat("Probability that Y (average purines) is greater than 15:", probability_Y_gt_15, "\n")
```

OUTPUT:
Probability that Y (average purines) is greater than 15: 5.317746e-07

Problem 2 (25 points)
Two genes' expression values follow a bivariate normal distribution. Let X and Y denote their expression values respectively. Also, assume that X has mean=7 and variance=3.Y has mean=12 and variance=7. The covariance between X and Y is 3.

In a trial, 100 independent measurements of the expression values of the two genes are collected and denoted as (X_1, Y_1), …,(X_{100}, Y_{100}). We wish to find the probability $P(\bar{X} + 0.5 < \bar{Y})$, i.e., the probability that the sample mean for the second gene exceeds the sample mean of the first gene more than 0.5.

Conduct a Monte Carlo simulation to approximate this probability, providing a 95% confidence interval for your estimation. Submit your R script for the Monte Carlo simulation.
Probability that Y (average purines) is greater than 15: 5.317746e-07
ANSWER:
CODE:

```
# Set parameters
X_mean <- 7
X_var <- 3
Y_mean <- 12
Y_variance <- 7
covar <- 3
sample <- 100
simul <- 10000

# Set up variables
exceed_count <- 0
results <- numeric(simul)

# Make simulations
for (i in 1:simul) {
```

```
  # Generate samples
  X_samples <- rnorm(sample, mean = X_mean, sd = sqrt(X_var))
  Y_samples <- rnorm(sample, mean = Y_mean, sd = sqrt(Y_variance))

  # Determine sample means
  mean_X_sample <- mean(X_samples)
  mean_Y_sample <- mean(Y_samples)

  # See if the mean difference is more than 0.5.
  if (mean_Y_sample - mean_X_sample > 0.5) {
    exceed_count <- exceed_count + 1
  }
  results[i] <- mean_Y_sample - mean_X_sample
}

probability <- exceed_count / simul
margin_of_error <- 1.96 * sqrt((probability * (1 - probability)) / simul)
lower_bound <- probability - margin_of_error
upper_bound <- probability + margin_of_error

cat("Probability:", probability, "\n")
cat("95% Confidence Interval: [", lower_bound, ",", upper_bound, "]\n")
```

OUTPUT:

```
Probability: 1
95% Confidence Interval: [ 1 , 1 ]
```

Problem 3 (25 points)

Assume there are three independent random variables X_1~chisq(df~8), X_2~ $Gamma(\alpha = 1,\ \beta = 2)$, X_3~ t-distribution with degrees of freedom m=5. Define a new random variable Y as $Y = \sqrt{X_1}X_2 + 4(X_3)^2$ (note that the square root is only for X_1.)

Use Monte Carlo simulation to find the mean of Y. Submit your R script for the Monte Carlo simulation.

ANSWER:
CODE:

```
num_simulations <- 10000
df_chi_sq <- 8
df_t_dist <- 5

mean_Y_result <- 0

for (iteration in 1:num_simulations) {
  X1 <- sqrt(rchisq(1, df_chi_sq))
  X2 <- rt(1, df_t_dist)
  X3 <- rchisq(1, df_chi_sq)

  Y <- X1 * X2 + X3

  mean_Y_result <- mean_Y_result + Y
}

mean_Y <- mean_Y_result / num_simulations

cat("Mean of Y:", mean_Y, "\n")
```

OUTPUT:
Mean of Y: 8.045585

Problem 4. (25 points)
Complete exercise 10 in Chapter 3 of *Applied Statistics for Bioinformatics using R* (page 45-46). Submit the plot, and a brief explanation of your observation.

The problem refers to the density function of extreme value distribution in another book. You do not have to look for the other book, the density function is

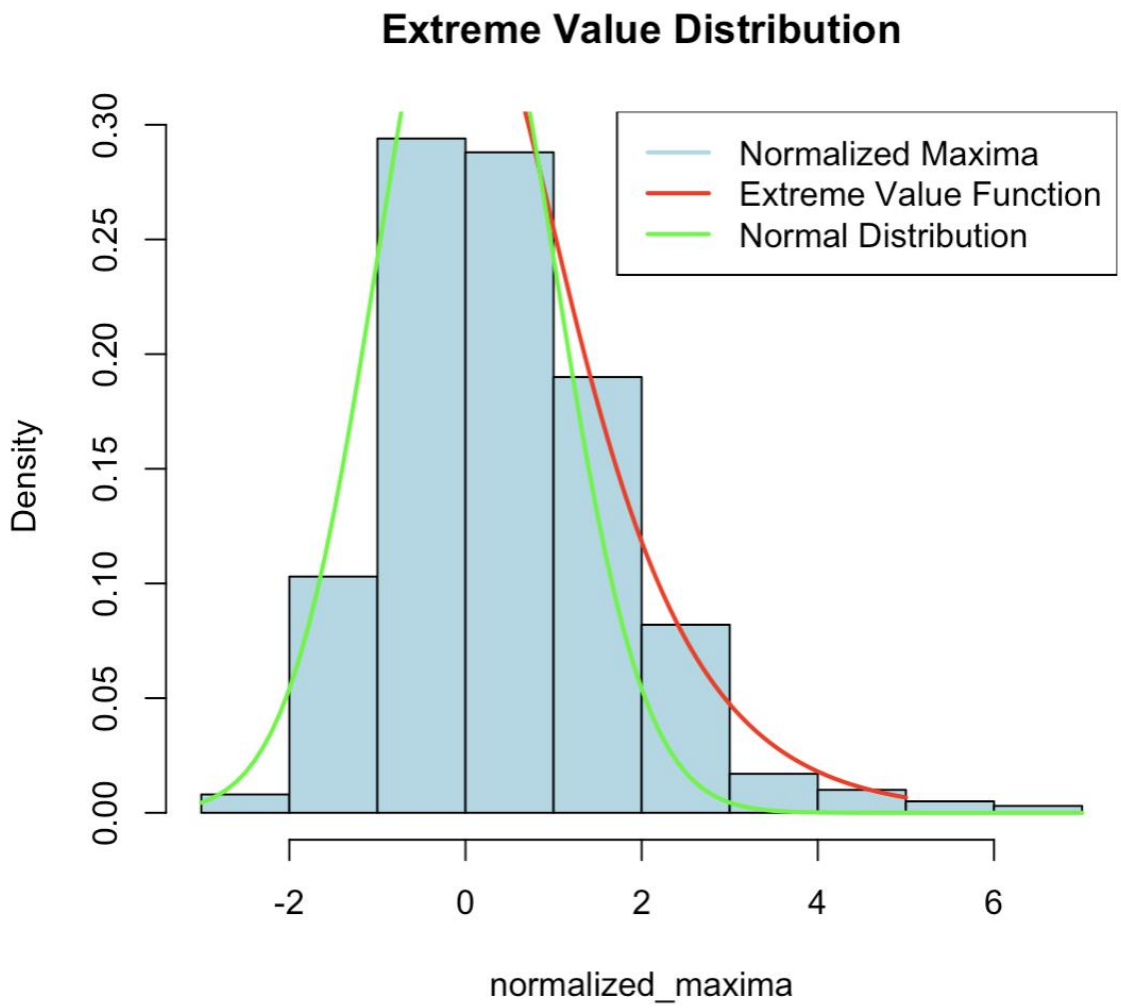
$$f(x)=\exp(-x)\exp(-\exp(-x))$$

Here exp(-x) is the same as e^{-x}.

ANSWER:
CODE

```
set.seed(123)

sample_size <- 1000
a_n <- sqrt(2 * log(sample_size)) - 0.5 * (log(log(sample_size)) + log(4 * pi)) * (2 * log(sample_size))^(-1/2)
b_n <- (2 * log(sample_size))^(-1/2)

maxima_values <- replicate(1000, max(rnorm(sample_size)))
normalized_maxima <- (maxima_values - a_n) / b_n

extreme_value_function <- function(x) exp(-x) * exp(-exp(-x))
x_values <- seq(0, 5, length.out = 100)

hist(normalized_maxima, prob = TRUE, col = "lightblue", main = "Extreme Value Distribution")
lines(x_values, extreme_value_function(x_values), col = "red", lwd = 2)
curve(dnorm(x), add = TRUE, col = "green", lwd = 2)

legend("topright", legend = c("Normalized Maxima", "Extreme Value Function", "Normal Distribution"),
       col = c("lightblue", "red", "green"), lwd = 2)
```

OUTPUT:


Extreme Value Distribution
Normalized Maxima
Extreme Value Function
Normal Distribution
Density
0.00
0.05
0.10
0.15
0.20
0.25
0.30
-2
0
2
4
6
normalized_maxima