Northeastern University
College of Science

Module 4 Homework

Problem 1 (10 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 purines in these microRNAs. Find the probability that Y is great 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)

1-pnorm(15,mean=14, sd = sqrt(20*0.7*0.3)/sqrt(100))

Output:

[1] 5.317746e-07

Problem 2 (35 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.

Answer)

```
require(mvtnorm)
nsim <- 10000
XmeanLess.sim <- numeric(nsim)
for (i in 1:nsim) {
  data.sim <- rmvnorm(100, mean = c(7, 12), sigma = matrix(c(3, 3, 3, 7),
nrow=2))
  mean.sim <- apply(data.sim,2,mean)
  Xmean <- mean.sim[1]
  Ymean <- mean.sim[2]
  # Check if Ymean is more than Xmean + 0.5
  XmeanLess.sim[i] <- (Xmean + 0.5 < Ymean)
}
# Calculate the mean of all the MC simulations
mean(XmeanLess.sim)

# Calculate the 95% confidence interval
mean(XmeanLess.sim) + c(-1, 1) * 1.96 * sqrt(var(XmeanLess.sim) /
nsim)
```

Output)

```
> mean(XmeanLess.sim)
[1] 1
>
> # Calculate the 95% confidence interval
> mean(XmeanLess.sim) + c(-1, 1) * 1.96 * sqrt(var(XmeanLess.sim) / nsim)
[1] 1 1
```

Problem 3 (30 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)

```
X1<-rchisq(n=10000, df=8)
X2<-rgamma(10000, shape = 1, scale = 2)
X3<- rt(10000, df=5)
Y<-sqrt(X1)*X2 + 4*(X3^2)
mean(Y)
```

output)

```
[1] 12.28662
```

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}.

Extreme value investigation. This (difficult!) question aims to teach the essence of an extreme value distribution! An interesting extreme value distribution is given by Pevsner (2003, p.103). Take the maximum of a sample (with size 1000) from the standard normal distribution and repeat this 1000 times. So that you sampled 1000 maxima. Next, subtract from these maxima an and divide by bn, where an <- sqrt(2*log(n)) - 0.5*(log(log(n))+log(4*pi))*(2*log(n))^(-1/2) bn <- (2*log(n))^(-1/2)

Now plot the density from the normalized maxima and add the extreme value function f(x) from Pevsner his book, and add the density (dnorm) from the normal distribution. What do you observe?

Answer)

```
an <- sqrt(2*log(n)) - 0.5*(log(log(n))+log(4*pi))*(2*log(n))^(-1/2)
bn <- (2*log(n))^(-1/2)
e <- double();
n <- 1000
for (i in 1:1000) e[i] <- (max(rnorm(n))-an)/bn
plot(density(e),ylim=c(0,0.4), xlab="x", ylab="Density", main= "Density plot of Normalized max and Extreme Values")
f<-function(x){exp(-x)*exp(-exp(-x))}
curve(f,range(density(e)$x),add=TRUE,col = "blue")
curve(dnorm,add=TRUE,col = "red")
legend("topright", c("f(x)", "Normal", "Normalized"), fill = c("blue","red","black"), lty= 1, lwd=2)
```

Output:

The graphic shows how the distribution of average and extreme values compared against the density of normalized maxima. They are so similar that it is difficult to tell them apart. The extreme value distribution is closer to the normal line at the mean, the normalized maxima are lower, and the normal line is higher and blue and green curves are closer and properly distributed.