BIOINFORMATICS & COMPUTATIONAL GENETICS MSc
PROBABILITY AND STATISTICS

S-Plus EXERCISE SHEET 4

1. Starting the Splus session

Double click the S-Plus icon, or start the program from the **Start** menu. When the program has opened up the *Object Browser* window, start a *Commands Window* by using the **Window -> Commands Window** pull down menus.

2. ORDER STATISTICS AND APPROXIMATIONS

We have seen in lectures that the maximum and minimum order statistics derived from independent and identically distributed samples have distributions that can be identified exactly. If

$$Y_1 = X_{(1)} = \min\{X_1, ..., X_n\} \qquad Y_n = X_{(n)} = \max\{X_1, ..., X_n\}$$

then

$$F_{Y_n}(y_n) = \{F_X(y_n)\}^n \qquad f_{Y_n}(y_n) = n\{F_X(y_n)\}^{n-1}$$

$$F_{Y_1}(y_1) = 1 - \{1 - F_X(y_1)\}^n \qquad f_{Y_1}(y_1) = n\{1 - F_X(y_1)\}^{n-1} f_X(y_1)$$

We can study such random variables using S-Plus .

First suppose that $X_1, ..., X_n \sim Exponential(\lambda)$ say with $\lambda = 3$, where

$$f_X(x) = \lambda e^{-\lambda x} \qquad F_X(x) = 1 - e^{-\lambda x} \qquad x > 0$$

We should have that

$$F_{Y_1}(y_1) = 1 - \{1 - F_X(y_1)\}^n = 1 - e^{-n\lambda x}$$

and

$$F_{Y_n}(y_n) = \{F_X(y_n)\}^n = \left\{1 - e^{-\lambda y_n}\right\}^n$$

Think about what each of the following steps is computing:

```
>n <- 500
>lambda <- 3
>nits <- 5000
>xmin_rep(0,nits)
>xmax_rep(0,nits)
>for(i in 1:nits){
+ x <- rexp(n,lambda)
+ xmin[i] <- min(x)
+ xmax[i] <- max(x)
+ }
>hist(xmin)
>hist(xmax)
```

```
>length(xmin[xmin <= 0.0001])/nits
>1-exp(-n*lambda*0.0001)
>length(xmin[xmin > 0.0005])/nits
>exp(-n*lambda*0.0005)
>length(xmax[xmax <= 2])/nits
>(1-exp(-lambda*2))^n
>length(xmax[xmax > 3])/nits
>1-(1-exp(-lambda*3))^n
```

EXERCISE Find a sequence of commands (or write a function) that allows the cdf and/or pdf for the two extreme order statistics to be plotted. Use something like the following construction:

```
>x <- c(0:5000)/500
>cdf.x <- function(x,lambda){1-exp(-lambda*x)}
>cdf.xmin <- function(x,lambda,n){1-(1-cdf.x(x,lambda))^n}
>cdf.xmax <- function(x,lambda,n){(cdf.x(x,lambda))^n}
```

Maximum Order statistics from *Geometric* (θ) random variables play a particular role in pattern detection and sequence alignment for biological sequences. Using the S-PLUS package we can simulate some suitable values for $\theta = 0.25$. The Geometric distribution in S-PLUS has mass function and cdf

$$f_X(x) = (1-\theta)^x \theta = (1-\phi)\phi^x \qquad F_X(x) = 1-(1-\theta)^{x+1} = 1-\phi^{x+1} \qquad x = 0,1,2...$$

where $\phi = 1-\theta$. For the simulation, proceed as follows:

```
>n <- 500
>theta <- 0.25
>nits <- 5000
>xmin_rep(0,nits)
>xmax_rep(0,nits)
>for(i in 1:nits){
+ x <- rgeom(n,theta)
+ xmin[i] <- min(x)
+ xmax[i] <- max(x)
+ }
>hist(xmin)
>hist(xmax)
```

EXERCISES (i) Study the distribution for different values of θ and n.
(ii) The theory of extreme order statistics says that if $X_1, ..., X_n \sim \mathit{Geometric}(\theta) \equiv \mathit{Geometric}(1-\phi)$ then the maximum order statistic $Y_n = X_{(n)} = \max\{X_1, ..., X_n\}$ has cdf given by

$$F_{Y_n}(y_n) = \{F_X(y_n)\}^n = \left\{1-\phi^{y_n+1}\right\}^n$$

and hence that

$$\mathrm{P}[Y_n < y_n] = F_{Y_n}(y_n - 1) = \{1-\phi^{y_n}\}^n$$

which can be approximated (for large n) by

$$\mathrm{P}[Y_n \geq y_n] = 1 - F_{Y_n}(y_n - 1) = 1 - \{1-\phi^{y_n}\}^n \approx 1 - \exp\{-n\phi^{y_n}\}$$

that gives an approximate method of computing a p-value in a runs test. Study the validity of this approximation for different values of ϕ and n

Extreme Order statistics from $Uniform\,(0,1)$ random variables play a role in pattern detection; in particular we again study the distributions of the minimum and maximum order statistics because we wish to use them as test statistics in a hypothesis test of the uniformity of the sample. We can do this as follows:

```
>n <- 500
>nits <- 5000
>umin_rep(0,nits)
>umax_rep(0,nits)
>for(i in 1:nits){
+ u <- runif(n)
+ y <- c(0,sort(u),1)
+ udiff <- diff(y)
+ umin[i] <- min(udiff)
+ umax[i] <- max(udiff)
+ }
>hist(umin)
>hist(umax)
```

EXERCISE Study the behaviour for different values of n

3. STATISTICAL HYPOTHESIS TESTING CALCULATIONS

S-PLUS is a package with many of the standard tests (Z-test, T-Test, Chi-squared test etc) For example, the function **t.test** allows a one sample or two sample test to be computed, and the pull-down menus **Statistics -> Compare samples** can be used to automate testing procedures. However, the tests can be carried out by construction of the test statistic, critical values and p-values etc from first principles. For example to test the hypothesis H_0 that $\mu = 10$ against a two-sided alternative hypothesis assuming that $\sigma^2 = 2^2$ for a generated sample.of size $n = 20$, type

```
>mu <- 10
>sigma <- 2
>c <- 10
>n <- 20
>x <- rnorm(n,mu,sigma)
>z <- (mean(x)-c)/sqrt(sigma^2/n)
>z
```

which computes the test statistic z, and then

```
>qnorm(0.025)
>qnorm(0.975)
>pvalue <- pnorm(-abs(z))+1-pnorm(abs(z))
>pvalue
```

that computes the critical values and p-value. Recall that, in S-PLUS the **q-** in functions look up the quantiles (i.e. ordinates) that give a specified probability value in the cdf (and hence are used to find critical values), and the **p-** functions give the probability values for a given quantile (and hence can be used to find p-values).

EXERCISES : For appropriately simulated data sets; think about how to carry out

- a one-tailed one sample **Z-test**
- a one sample **T-test**
- a one sample test on σ^2
- a two sample **Z-test**
- a two sample **T-test** where $\sigma_1^2 = \sigma_2^2 = \sigma^2$ is unknown
- a two sample **F-test** of $\sigma_1^2 = \sigma_2^2$

In each case, think about how the calculation would work for different values of significance level α in both one-and two-sided hypothesis tests. For the two sample t-test, the "pooled" estimate of variance is used in the formula for the test statistic, that is, we need for two samples x and y

$$s_P^2 = \frac{(n_X - 1)s_X^2 + (n_Y - 1)s_Y^2}{n_X + n_Y - 2}$$

and then use the test statistic t defined by

$$t = \frac{\bar{x} - \bar{y}}{s_P\sqrt{\dfrac{1}{n_X} + \dfrac{1}{n_Y}}}$$

so to compute t

```
>nx <- 10
>ny <- 15
>x <- rnorm(nx,10,1)
>y <- rnorm(ny,10,1)
>sx <- var(x)
>sy <- var(y)
>sp <- ((nx-1)*sx+(ny-1)*sy)/(nx+ny-2)
>t <- (mean(x)-mean(y))/sqrt(sp*(1/nx+1/ny))
```