Lab #1 Basics in Spatial Statistics  
(Due Date: 01/30/2017)

PURPOSES
1. Get familiar with statistics and GIS
2. Learn to use open-source software R for statistical analysis

Before starting your lab, create a new directory named lab1 in your network drive (where you could organize files for this course), and create a Word document named lab1_yourusername.doc for your lab write up.

1. Statistical Software R

There are a suite of software packages for statistical analysis, for example, SAS, SPSS, and R. We will focus on using R, an open-source software package (http://www.r-project.org/), for the study of spatial statistics. There are online R tutorials or manual that you could use for self-study. A nice R manual (Introduction to R) is available from here: http://cran.r-project.org/doc/manuals/r-release/R-intro.html

In this section, we will learn basics in R software.

1.1. Launch R in Windows

R can be installed in alternative operating systems, such as Windows, Linux, Mac. We will use Windows version here.

To launch R, go to Start->All Programs->R. Click on the R program, you will have the GUI (Graphic User Interface) of R (see the following figure).
Now get familiarized with this GUI of R (demo here).

1.2. Basic operations in R

R supports scripting language, which means you could just type commands to conduct statistical analysis. In the Console of R, after “>” symbol, you could specify commands. Here are some examples that you could try:

1. Type: 10+20
2. Type: 10*20
3. Type: 10/20

Then you will results as shown in the following figure.

To declare a variable (say, x) and assign it with a value, type:

\[
> x <- 10
\]

Then type x to see what you have.

1.3. Data structures in R

R supports a set of data structures for representing a vector of data instead of a single value. These data structures include array, matrices, lists, and data frames. Detailed explanation can be found from the R manual here: [http://cran.r-project.org/doc/manuals/r-release/R-intro.html](http://cran.r-project.org/doc/manuals/r-release/R-intro.html)

For example, if you want to declare a vector (say, y) that contains four values (say, 10, 15, 30, 20), use the following command:
Now type y to see what you have.

Operations on data structures in R are very flexible, which allows us to manipulate our data easily. Try the following command if you define the vector y:

```r
> z <- c(y, 10, 10, y)
```

You could directly conduct operations on vectors in R. For example, try the following command:

```r
> k <- 10*y + 100
```

Type k to see your results of vector operation.

Now you could try to use other data structures (e.g., matrix or data frame) to store your data possibly in multiple dimension. For example, if you want to create a two dimension array, use the following command (adapted from Introduction to R):

```r
> x <- array(1:20, dim=c(4,5))
```

If you want to replace array x with value 0, use the following command:

```r
> x <- 0
```
Now try to learn other data structures (list or data frame) by following the online R manual on your own. These data structures give you more flexibility to manipulate and analyze your data.

Anytime you need help on a specific function (say, rbind), type: help(function name)
Now type help(rbind) to see what you have.

1.4. Summary statistics in R
To conduct descriptive statistics in R is relatively straightforward. The simplest approach is to use function summary(data).

Declare a vector x with a set of values using the following command:
> x <- c(10, 20, 30, 20, 10, 30)
Then type:
> summary(x)

Of course, you could use specific commands to obtain the mean, variance, standard deviation of a variable. So try the following command:
> mean(x)
> var(x)
> sd(x)

The following figure shows what you will have:

Next, generate another variable y, which is a row-wise vector with 6 values.

> y <- c(22, 12, 3, 50, 100, 6)
To obtain the covariance between variable x and y, use the following command:

```r
> cov(x,y)
```
Likewise, using `cor(x,y)` gives you the correlation coefficient of variable x and y.

R provides graphical functionality that allows to visualize the data directly. The function that supports the graphical functionality is `plot()`. So try the following command:

```r
> y <- x
> plot(x,y)
```

If you want to see more graphic functionality available in R, check out this link: [http://www.r-project.org/screenshots/screenshots.html](http://www.r-project.org/screenshots/screenshots.html)

1.5 Random Number Generation in R

R provides a suite of functions that allows us to draw random numbers from standard probability distribution (e.g., uniform distribution, normal distribution). For example, if you want to generate a vector of random values following uniform distribution, use the command `runif(number of values)`.

Now, generate a vector with 100 uniformly distributed random values using the following command:

```r
> x1 <- runif(100)
> x1
```
If you want to generate a vector with random numbers following normal distribution, use the command `rnorm(number of values)`. Try the following commands to generate two vectors each having 1000 random numbers.

```r
> random<-cbind(rnorm(1000),rnorm(1000))
> plot(random)
```
2. GIS Software for Spatial Statistics

A set of GIS software (commercial or open source) is available. We will focus on using ESRI ArcGIS. I assume most of you are familiar with ArcGIS. So here are some illustrations on ArcGIS environment.

First, launch ArcGIS in your Windows environment:

ArcGIS has a spatial statistics toolbox (in Arc Toolbox) that collects a set of spatial statistics functions (e.g., spatial autocorrelation, spatial cluster analysis, and spatial regression). If you invoke ArcToolbox, you will see these functions. At this time, you just need to keep in mind this option for spatial statistics.
Questions:

**Question 1:** Create an $n \times m$ matrix ($n=100$, $m=5$; hint: you could use `cbind` function to combine $m$ vectors). In this matrix, each column is a vector with $n$ random values following uniform distribution. Generate the scatterplot of these five variables (hint: use function `pairs()`). Write down your commands in your report as well as inserting the screenshot for your scatterplot and the data that you generated.

**Question 2:** What are the estimator formulas for the mean, variance, standard deviation of a variable.

**Question 3:** What are the estimator formulas for the covariance and correlation coefficient of two variables.

**Question 4:** Conduct summary statistics of the matrix that you generated in Question 1. Write down the mean, variance, standard deviation of each variable. Write down the covariance matrix and correlation coefficients of the five variables.

**Note that** you need to provide the R commands (code) that you use together with the results. Otherwise, you will receive ZERO credit.