ASSIGNMENT 1: INTRODUCTION TO R IN HYDROMETEOROLOGY

Due: Wednesday, 23 January 2019

1 Objectives

The discipline of hydrology focuses on the occurrence and movement of water at and near the earth's surface. Of particular interest are the quantities of water stored (for example, as soil moisture), and the rates at which water moves (e.g., from a lake to the atmosphere by evaporation). This exercise introduces typical calculations related to quantifying storage and movement of water, working with units of measurement, and basic descriptive statistics.

This assignment will first introduce you to 'R', a statistical/data analysis package that we will be using for the course.

2 Introduction to R

R should be installed on your desktop computer in your office or on a personal laptop computer. Because R is open source, you can download it from this web site: (http://www.r-project.org/index.html) and install it on your own machine. Note also that R is available in multiple flavors (windows, mac, linux, unix). R may also be available through remote desktop connections through Student Terminal Services (see details below).

2.1 Starting R

To start R under Windows, find it under Programs → R.

To start R through the remote desktop connection to Student Terminal Services, follow the instructions here: `http://www.unbc.ca/service-desk/virtual-desktop-students`, and then find R through the Start menu.

To start R in a new working directory, first find the command to start R ('Rgui.exe'), and create a shortcut on your desktop. Right-click, and, select 'Properties' and then edit the start location to reflect your working directory.

2.2 Using R as a Calculator

The simplest thing R can do is evaluate arithmetic expressions:

```
> 1
[1] 1
> 1+4.23
[1] 5.23
```

```
> 1+1/2*9-3.14
[1] 2.36
# Note the order in which operations are performed in the final calculation
# Now try the following:
> (1+1)/2*9-3.14
[1] 5.86
```

Note: the # sign indicates a comment. R ignores anything after a # sign, and you can too when entering these commands. Also note that the text appears as it does on your screen. This is on purpose.

2.3 Assignment, Vectors, and Indexing

Better still, you can assign values or text strings to variables:

```
> a=5 #assigns a value of 5 to variable 'a'
> a
[1] 5
> a==4 #tests whether 'a' is equal to 4
[1] FALSE
```

You can create vectors of data using the concatenation function `c(...)`:

```
> a = c(1,2,3)
> a
[1] 1 2 3
> a =  c("Ali","Bet","Cat")
> a
[1] "Ali" "Bet" "Cat"
```

Note that strings are denoted by single (or double) quotation marks.

Now create a variable 'ppt', with a series of values, and use some basic statistical functions in R to extract basic statistical information about 'ppt':

```
> ppt = c(12,23,56,67,78,56)
> mean(ppt)
[1] 48.66667
> summary(ppt)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  12.00   31.25   56.00   48.67   64.25   78.00
> mean(ppt[2:5]) # INDEXING: only use values in positions 2 through 5
[1] 56
> sd(ppt) # return standard deviation
[1] 25.71899
```

2.4 Data Frames

Data frames are a collection of variables (known as vectors in R). Variables are arranged in columns while cases of a variable are arranged in rows.

```
> a = data.frame(x=c(1,2,3),y=c(0,5,10))
> a
  x  y
1 1  0
2 2  5
3 3 10
```

Another example:

```
> runoff=data.frame(a=rep(5, 10), b=rnorm(10), c=sort(rnorm(10)))
```

You just made a data frame called ′runoff′ consisting of three variables,′a′, ′b′, and ′c′. Variable ′a′ was made by repeating the number ′5′ ten times (using the rep() command). Variable ′b′ is 10 random normal variates (random numbers with mean of zero and standard deviation of 1), and ′c′ is a list of sorted random normal variates.

As you might guess, R has thousands of built-in functions, which are specified with arguments inside brackets. To access help for any function you can simply type **help(function)** where ′function′ is the one of interest. Try `help(plot)` or `?plot`. (Tip: To escape from the help text, just hit ′q′)

2.5 Missing Values

Its important to note that missing values are specified as NA. If you use a zero instead, what would happen to your calculated statistics? Values of NA also get special treatment, depending on the command:

```
> b=c(NA,4,5,7)
> b
[1] NA  4  5  7
> mean(b)
[1] NA
> mean(b,na.rm=T) #calcuate mean, ignore NA's
[1] 5.333333
```

Finally, the command `> ls()` lists all objects in R's namespace (created during a given session) and to remove an object from the workspace you use `> rm(a)`. Once an object is removed from R's namespace (its memory during our session), it's gone and can't be recovered. Similarly, renaming an object is similar to removing the first object (i.e. a=b, old object a is now replaced with values from b).

Loading Data in R

First, get organized and make a directory called `A1` under your student account (the H: drive). Copy the file 'hydro.txt' from the course website (`weather.unbc.ca/310/799.html`) to your new directory.

Having a new directory (i.e. A1, A2, etc) for each assignment is convenient so that you can keep track of all of your data (your data files, graphs, and write ups) for a given assignment.

After starting a new R session, load the text file:

```
> hydro = read.table('hydro.txt', header=T)
```

If the file is in a comma-delimited format (e.g., '.csv'), then you can use:

```
> hydro = read.table('hydro.csv', header=T, sep=",")
or
> hydro = read.csv("hydro.csv")
```

We have just assigned everything in the text file 'hydro.txt' to the R object 'hydro', which is a dataframe. The `header=T` command tells R that our input file contains variable names (you can verify this by opening the file in Excel).

These data represent 3 years of hourly streamflow measurements for two rivers, Lillooet River (08MG005) and Van Horlick Creek. The contributing watersheds are 2160 km^2 and 118 km^2, respectively. The records begin around 15 April and end late in November.

To see the headers of the data frame, type

```
> head(hydro)
```

Once you know the variable names (i.e. the column titles), you can subset or acess the individual vectors in the data frame as follows:

```
a = hydro$lill98
```

2.6 Visualizing Your Data

One of R's most powerful aspects is its ability to produce high-quality graphs. To get a flavor of its capabilities try running the graphics demo:

```
> demo(graphics())
```

Start by making a simple plot of Lilloet 1998 data:

```
> plot(a)
```

This brings up a graphics window with the values of a (y-axis) plotted against their Index (or their position within the vector: try a[1] to see what the first value is, or a[2000] for the 2000th variable, etc.)

To plot the data with a valid x-axis (time), let's create a new variable for days since 15 April, and call the plot command:

```
> days=(1:length(hydro[,1]))/24  #create vector of decimal days since 15 April
> plot(days,hydro[,1])
> plot(days,hydro[,1],type='l',col='red', xlab='Days since 15 April',
  ylab=expression(Discharge~(m^3~s^{-1})))  #make it fancy
```

The final command here gives a number of arguments to the plot command: type='l' specifies a line graph, col='red' does what you think it does, and the xlab and ylab commands set the x and y-axis labels. Type help(plot) for more details. The expression command takes a while to get used to, but can be used to set subscripts and superscripts in the axis labels.

2.7 Saving Your Plots

To save an R plot in a .pdf or .png file, simply type

```
> pdf('test.pdf')
...
## Type all your plotting commands here
...
> dev.off()
```

Assignment Part 1: Basic Stats and Plotting (20 marks)

Using the data file 'hydro.txt', determine the following:

1. In a table, provide the mean, maximum, minimum, and standard deviation in streamflow (Q) for the Lillooet River and Van Horlick Creek for each year (1998, 1999 and 2000). (4 marks)

2. Using the hist(x) function, visually compare the distributions of Q for each river for the 2000 runoff season. Are the data normally distributed? You can also statistically test this. Let's test for normality using the Shapiro Wilks test,

   ```
   > shapiro.test(x)
   ```

where vector x is the variable you want to test. Use the help command to examine the details and alternatives for normality testing, and include the output for each site in your answer.

Save your histogram plots and include them in your answer. (4 marks)

3. Make a plot with all the discharge records on a single page. To do this, we first need to tell R that we are planning on putting 6 plots on a page:

```
> par(mfrow=c(3,2))
> plot(days, hydro$lill98, col=’red’, type=’l’,
 main=’Lillooet River 1998’,
 xlab=’Days from 15 April’,
 ylab=expression(Discharge~(m^3~s^-1)))
>plot(days,hydro$van98, col=’blue’,type=’l’,
 main=’Van Horlick Creek 1998’,
 xlab=’Days from 15  April’,
 ylab=expression(Discharge~(m^3~s^-1)))
```

   Continue adding plots and changing the y-data and the main label as appropriate. Use the `dev.off()` command at the prompt to reset plot dimensions if you make a mistake. Include this plot in your report. (4 marks)

4. Standardize the annual data for both the Lillooet River and Van Horlick Creek (i.e. transform each series so they have a mean of zero and a standard deviation of 1). To do this you will need to calculate the mean (μ_x) and standard deviation (σ_x) of each time series (Hint: you can use the results from question 1), and create a new standardized time series:

$$x_{std} = (x - \mu_x)/\sigma_x$$

   Plot these new standardized time series on three plots (one for each year), with different colors each site, and give the plot a legend (type `help(legend)` for help). Label the y-axis with the appropriate units, and include this plot in your writeup. (6 marks)

5. What might be causing the difference in discharge between the two rivers (both are snowmelt driven, located close to one another, and their hypsometry (i.e. distribution of elevation as a function of area) are similar)? (2 marks)

Assignment Part 2: Linear Regression (10 marks)

Channel width and depth data were collected at several locations within the Bivouac Creek watershed in the Fort St. James district. Measurement points were geo-referenced and upstream drainage area from each was determined in the office. Using the data in the file 'Bivouac-Drainage-Area.csv', provide the following:

1. Cross-sectional area (in m^2) for each location. [Hint: Multiply the width times the depth of the channel to obtain the cross-sectional area (CSA).] (2 marks)

2. Conduct a linear regression of the drainage area (x) and cross-sectional area (y) variables using the linear model function:

   ```
   lmod = lm(y~x)
   ```

   The command `summary(lmod)` will output a summary of the model. What is the regression equation? Is it statistically significant? What is the R^2 value of the regression, and what does this really tell you? (4 marks)

3. A scatter plot of drainage area (x) versus cross-sectional area (y) of the main channel. Use proper labels, and add the linear model to your plot using the `abline(lmod)` command. (4 marks)