Introduction to Statistical Analysis and Programming

Lecture 1

Introductions

Brief Biography

Instructor: Siraj Ul Islam **Email:** sirajul.islam@unbc.ca **Office:** 4-256

- QAU: MSc, Physics, PAKISTAN
- QAU: MPhil, Computational Physics, PAKISTAN
- GCISC: Scientific Officer (Climate Modeling), PAKISTAN
- ITCP: Junior Associate (Earth System Science), ITALY
- UNBC: PhD, Climate Modeling/Dynamics, CANADA
- UNBC: PDF, Hydrological Modeling, Analysis, CANADA
- UNBC: Adjunct Professor, Environmental Science, CANADA
- UNBC: Research Associate, Hydrological-Water Temperature Modeling , CANADA

Course Outlines

- Course: Introduction to Environmental Data Analysis
- Lectures: Teaching Lab 8-162 , Tuesday, 09:30am-10:20am
- Labs: Teaching Lab 8-129, Tuesday, 02:30pm-05:20pm
- Website: http://web.unbc.ca/~islam/

Course Description

This course focuses on the basic training and practical skill in analyzing environmental data using the R programming language. The focus is on the principles and practicality of programming with R, including reading data into R, accessing R packages, writing R functions, debugging and organizing/commenting R code. Topics in environmental data analysis are used as working examples.

Objectives

To develop practical skills of data analysis and computer programming. The lectures will emphasize conceptual understanding, while the labs will facilitate practical understanding i.e. 'learning by doing'. By the end of this course students will know how to apply R programming to perform environmental data analysis.

Format

- This course will consist of lectures and labs with emphasis on the lab, in which the students will solve practical problems using R for basic statistical, graphical and spatial analysis of geophysical and environmental data.
- At the start of each lab, instructor will provide a quick review of the lecture and will introduce a sample dataset that will be used by the students.
- Students will work individually with assistance and guidance from the instructor.

Evaluation

Test 1:	20%
Test 2:	20%
10 Labs:	50% (10 labs x 5%)
Class Presentation:	10%

Labs Submission

- The labs require a formal, typed write-up R studio notebook, the format of which will be discussed in class. Test 1 will cover first 5 labs and test 2 will cover remaining 5 labs

Class Presentation

- Student will need to select an appropriate topic in consultation with the instructor.

- Presentation will be 6-10 slides with introduction of analysis method and data, results and conclusion.

Programming Tool

- R language
  - https://www.r-project.org/about.html
  - Open source
- R Studio
  - https://www.rstudio.com/
  - Open source

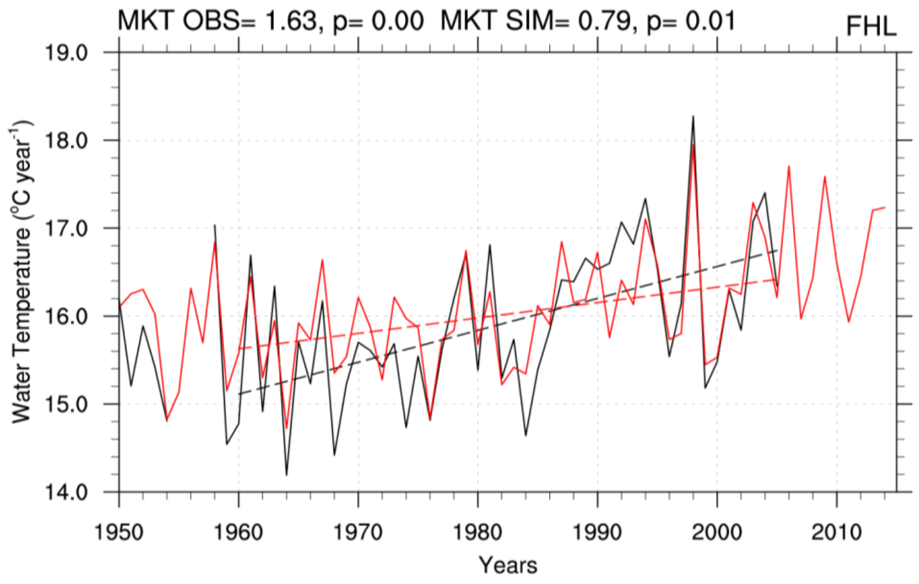
Any Questions ?

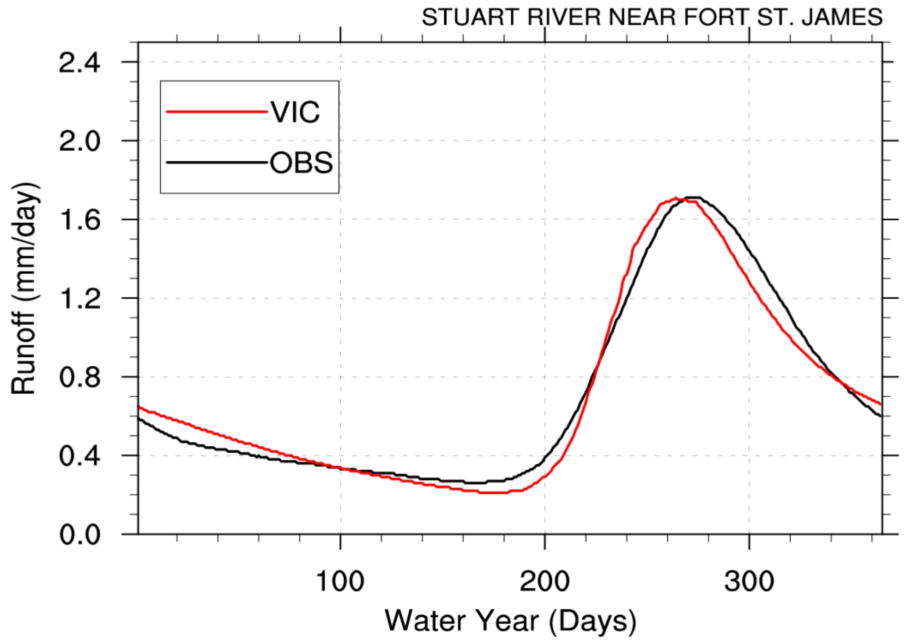
Scientific Datasets and Analysis

- The goal of data analysis is to discover relationships and features in a dataset.
- Geophysical datasets are those that record a measurement at particular location. Data are measurements of something of interest. They are also called observations because the measurements help us to observe (and to quantify) an attribute of whatever is being studied.
- Geophysical data analysis can be understood as using the data to detect, examine, understand or predict events and phenomena that are of relevance to a geophysical study.

Datasets types

- Time Series





Importance and application of statistics in sciences

- Example: Climate Change
  - Rate of Change air temperature with time
  - Dependence of rainfall on SST
  - Mean change within a time period
- Example: Rivers flows
  - Changes in river flows quantity and timing
  - Increasing or decreasing variability of flows

Programming

- Computer programming involves controlling computers, telling them what calculations to do, what to display, etc.
- Statistical programming involves doing computations to aid in statistical analysis. For example, data must be summarized and displayed. Models must be fit to data, and the results displayed.

R Package

- R is based on the computer language S, developed by John Chambers and others at Bell Laboratories in 1976.
- In 1993 Robert Gentleman and Ross Ihaka at the University of Auckland wanted to experiment with the language, so they developed an implementation, and named it R. They made it open source in 1995, and thousands of people around the world have contributed to its development.

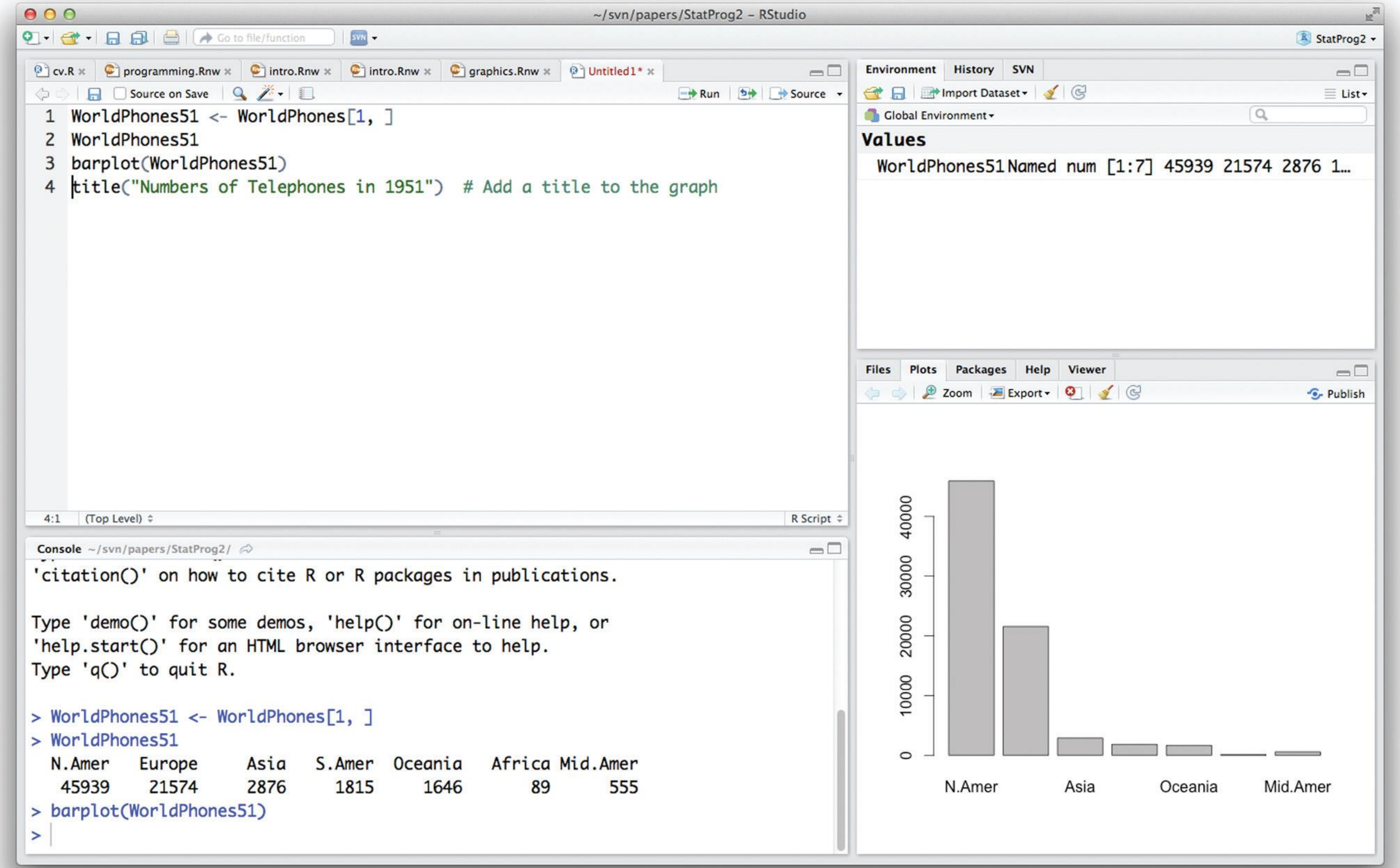
Source: A first course in statistical programming with R - W. John Braun and Duncan J. Murdoch, Cambridge, 2016.

Programming in R

- R is an open source computing package which has seen a huge growth in popularity in the last few years. Being open source, it is easily obtainable by students and economical to install in our computing lab.
- Students starting this course are not assumed to have any programming experience or advanced statistical knowledge. They should be familiar with university-level calculus.

Source: A first course in statistical programming with R - W. John Braun and Duncan J. Murdoch, Cambridge, 2016.

RStudio



```r
WorldPhones51 <- WorldPhones[1, ]
WorldPhones51
barplot(WorldPhones51)
title("Numbers of Telephones in 1951")  # Add a title to the graph
```

Values

```
WorldPhones51 Named num [1:7] 45939 21574 2876 1…
```

```
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> WorldPhones51 <- WorldPhones[1, ]
> WorldPhones51
 N.Amer  Europe    Asia  S.Amer Oceania  Africa Mid.Amer
  45939   21574    2876    1815    1646      89      555
> barplot(WorldPhones51)
> 
```

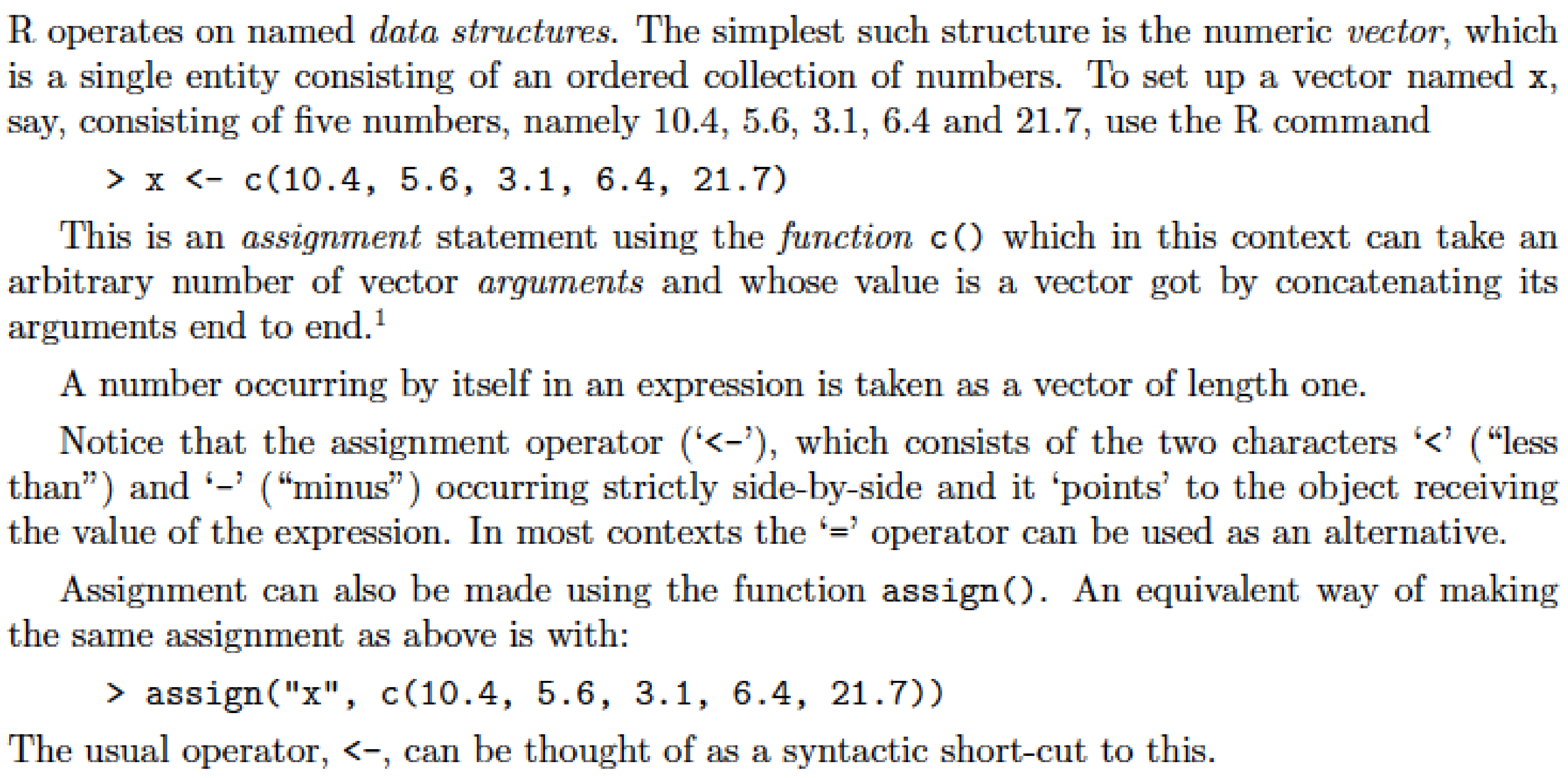
Source: A first course in statistical programming with R - W. John Braun and Duncan J. Murdoch, Cambridge, 2016.

R Studio

To create new script file in R studio, go to the "File" menu, select the "New" option, and then click on "R script". This will open a new window within the "source" panel. Then you can type the commands you want (or ***code*** as it is generally called when you're typing the commands into a script file) and save it when you're done.

Source: A first course in statistical programming with R - W. John Braun and Duncan J. Murdoch, Cambridge, 2016.

Vectors and Assignment



R operates on named *data structures*. The simplest such structure is the numeric *vector*, which is a single entity consisting of an ordered collection of numbers. To set up a vector named `x`, say, consisting of five numbers, namely 10.4, 5.6, 3.1, 6.4 and 21.7, use the R command

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
```

This is an *assignment* statement using the *function* `c()` which in this context can take an arbitrary number of vector *arguments* and whose value is a vector got by concatenating its arguments end to end.[1]

A number occurring by itself in an expression is taken as a vector of length one.

Notice that the assignment operator (‘`<-`’), which consists of the two characters ‘`<`’ (“less than”) and ‘`-`’ (“minus”) occurring strictly side-by-side and it ‘points’ to the object receiving the value of the expression. In most contexts the ‘`=`’ operator can be used as an alternative.

Assignment can also be made using the function `assign()`. An equivalent way of making the same assignment as above is with:

```
> assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
```

The usual operator, `<-`, can be thought of as a syntactic short-cut to this.

Vectors and Assignment

If an expression is used as a complete command, the value is printed *and lost*[2]. So now if we were to use the command

```
> 1/x
```

the reciprocals of the five values would be printed at the terminal (and the value of x, of course, unchanged).

The further assignment

```
> y <- c(x, 0, x)
```

would create a vector y with 11 entries consisting of two copies of x with a zero in the middle place.

```
> v <- 2*x + y + 1
```

generates a new vector v of length 11 constructed by adding together, element by element, 2*x repeated 2.2 times, y repeated just once, and 1 repeated 11 times.

R Scripting

```
## --- hello.R
 x <- "hello world"
print(x)

source( "hello.R" )
```

Script with comments

```
## --- abc.R
# A script to analyse xyz.data
# author: Siraj # date: 5/9/2019
# Loading the data
```