# Laboratory 1 – Intro to Geostatistics

Objectives:

The objective of this lab is to set up a working environment from which you can begin to use and learn programming in R. We will each be using our own laptops, so the goal will be to create a persistent set of tools that you can customize and maintain throughout the course and beyond. You will also be loading, running and manipulating your first bit of R code, focusing on aspects that will be unfamiliar now, but you will understand in depth after the next few weeks.

### Introduction and Lab Structure:

I would also like to introduce you to the general structure, content and procedures that the labs throughout the semester will generally follow. The first of which is how the assignments will be packaged and formulated. In general, assignments will be packaged in a single Zip archive that can be downloaded from the Canvas website. Inside this archive you should find a Microsoft Word document containing an introduction, instructions and any questions to be answered and submitted.

Also in this archive you will find folders containing all files relevant to the current lab. For example, today’s lab contains:

/data/ - A folder containing all of the raw data needed

/src/ - Contains all source code and tools needed

/output/ - This should be empty to begin with

At the end of the lab you will re-Zip the /src/ and /output/ folders along with this or another Word file into which you’ve typed the answers to all questions and submit them via Canvas by the deadline set via the Canvas system (in general this will be by midnight on Tuesday, one week from the current lab day).

Questions may appear throughout the lab text in all bold. I suggest copying these questions into a separate word document and answering them there as you go through the lab. Technical instructions and steps to follow will be presented in numbered lists. Code referenced in the lab text will appear in code font.

### Setting Up Your Tools:

I have a firm belief in using the right tool for the job. That said, what is the right tool for me may not work for you. So I’m going to present one toolset that works well for programming in R specifically. Furthermore, I’m going to walk you through setting this toolset up for yourself.

R can be easily programmed and used “interactively.” What I mean by this is that you can open R from the desktop and type commands directly into the graphical user interface (GUI). This works well if you need to test something quickly or use R as a glorified calculator (try it!).

But this type of interaction is not well suited for typical scripting or programming complicated analyses. Most of the time you will want to save a series of statements into a file that can be opened, commented and re-used if for nothing other than documenting what process resulted in the results you generate. Therefore we typically will be saving our output into a script file.

In the R GUI, click on the “File > Open script…” menu item and let’s open the “sample.R” code I provided in the /src folder you unzipped this lab from. This will open the script in an Editor window. From this window you can select and “send” lines or sections of commands to the R console. In practice this works extremely well and is a perfectly adequate way of entering and using R code. However, there are some features of modern programming tools that the R Editor are missing, such as syntax highlighting, code completion, expanded search and replace, etc.

So, if you are interested in any of these things I suggest several alternatives. In class we will set up a free alternative called [RStudio](https://www.rstudio.com/) that I highly recommend for starters. RStudio is a nice GUI that allows you to edit code and visualize output in one neat place. You may alternatively use a text editor like [Notepad++](https://notepad-plus-plus.org/) (a text editor ideally suited for programming in multiple languages) and [NppToR](http://sourceforge.net/projects/npptor/files/PortableApp/) (a program that “connects” Notepad++ and R) or similar Sublime Text. I personally use [Sublime text](https://www.sublimetext.com/) to edit my code and then run it through the R console using the [SendCode](https://github.com/randy3k/SendCode) package.

## Questions

1. **To show that you successfully configured your R programming environment take a screen capture of your R studio window with the sample.R code open in the editor window (pane on top left of RStudio) [50 points]:**
2. **To show that you can successfully execute R code in your R environment, take a screen shot of the final bubble plot of the non-correlated residuals from the sample script [50]:**

**[Total Points: 100]**

**To submit your assignment, please re-zip the /output/ and /src/ directories along with the file you’ve typed all of the answered lab questions in to a single Zip file. This is the file you’ll upload and submit via Canvas to complete your assignment.**

**Thanks!**