**R review of data object types/structures, classes, methods, iteration, and functions (video lectures)**

* [R code (updated so, not perfectly synched with lectures but, generally follows topically)](https://spatialr.s3.us-west-2.amazonaws.com/introduction/Introduction.html)
* [Slide deck (updated so, not perfectly synched with lectures but, generally follows topically](https://spatialr.s3.us-west-2.amazonaws.com/introduction/RIntroductionLecture.pptx)

Part 1 – [Vector, data.frame and sampling](https://spatialr.s3.us-west-2.amazonaws.com/introduction/Intro_01.mp4)

Part 2 - [list, matrices and table](https://spatialr.s3.us-west-2.amazonaws.com/introduction/intro_02.mp4)

Part 3 - [No Data, object classes, methods](https://spatialr.s3.us-west-2.amazonaws.com/introduction/intro_03.mp4)

Part 4 - [spatial object structures and import/export of spatial data](https://spatialr.s3.us-west-2.amazonaws.com/introduction/intro_04.mp4)

Part 5 - [loops, apply and functions](https://spatialr.s3.us-west-2.amazonaws.com/introduction/intro_05.mp4)

CRAN released a “major version” at 4.2.0 (4.3 just released) with changes that required numerous package updates. Given the rapid development, bug fixes and tight version controls on R dependencies, I would HIGHLY recommend updating your R version to the latest along with all of your packages. Otherwise, you could end up with libraries that do not include improvements/bug fixes because they will not install under previous R versions. If using RStudio you will need to point to the updated version of R and not expect that this happens automatically.

Here are links for a pdf [document stepping through installing R](https://spatialr.s3.us-west-2.amazonaws.com/Install+R.pdf) and an [ASCII Rprofile.site](https://spatialr.s3.us-west-2.amazonaws.com/Rprofile.site) file that sets quite a bit of R configuration. Just replace the empty file in your R install (eg., “C:\Program Files\R\R-4.2.3\etc”). Please also install [RTools](https://cran.r-project.org/bin/windows/Rtools/rtools43/files/rtools43-5550-5548.exe) as it will facilitate installs from GitHub, RUniverse and from source code.

All of the required libraries are stable and up on CRAN. However, it looks like spatstat just got rid of the spatstat.core library (at the R 4.3 release) and are now just using the main spatstat library to manage all of the sub-packages. As such, the install code was throwing an error when encountering this depreciated package. There is also a new version of RTools43 corresponding to the release of R 4.3. Because there are some hard-and-fast version dependencies, please update to [R 4.3](https://cran.r-project.org/bin/windows/Rtools/rtools43/rtools.html) and [RTools43](https://cran.r-project.org/bin/windows/Rtools/rtools43/rtools.html). If you already have a eailer version of R installed, just install 4.3, cut-and-paste all of the directories in the old installs library directory to the new library and when asked do not replace existing directories/files. Then you can open an R terminal and in the “Packages” menu, select “Update packages…” and the new install will then be up to date with all of your packages. This is given that you are not using a personal library because you do not have permissions open to the library directory (see below).

Make sure that you have open permissions for the R install “library” directory. To do this open file explorer and navigate to the “library” directory (eg., C:\Program Files\R\R-4.3.0\library). Then right click and select “properties”, in the resulting window select the “security” tab and click edit in the top window. Scroll down and select “users” in the top window, check “full control” under the “allow” column in the bottom window then click OK. Windows will then ask for your credentials to run as administrator. Once it is done dismiss the window. You can do the same for the etc directory so that you can add custom startup options to the Rprofile.site file.

Here is a scroipt that will install required libraries

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

.Library.site **<-** file.path**(**chartr**(**"\\", "/", R.home**())**, "library"**)**

.libPaths**(**file.path**(**chartr**(**"\\", "/", R.home**())**, "library"**))**

options**(**repos **=** c**(**

  jeffreyevans **=** 'https://jeffreyevans.r-universe.dev',

  CRAN **=** 'https://cloud.r-project.org'**))**

p **<-** c**(**"devtools", "dplyr", "forecast", "GeNetIt", "ggplot2", "igraph",

"landscapemetrics", "pdp", "randomForest", "ranger", "RANN",

"raster", "rfUtilities", "rgdal", "rgeoda", "rgeos", "rts", "sf",

"sfnetworks", "sp", "spatialEco", "spatstat", "spdep", "terra", "tmap"**)**

lapply**(**p, install.packages**)**

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

I personally use Notepad++ () as my IDE/Code editor but, RStudio would work as well. Although, I really prefer the R Console for many reasons.

    Notepad++ <https://notepad-plus-plus.org/>

    RStudio <https://www.rstudio.com/products/rstudio/download/>

I have taught this workshop many times, trying different pedagogies. Teaching coding is just hard! I have decided to try a bit of a free form approach for this offering. I have the materials organized in the order that I have previously taught them but, rather than having a rigid schedule to adhere to, we will just work through the materials sequentially. This is how it happens organically anyway so, I figured that we would just go with what works naturally. This will also allow me some flexibility to do “deeper dives” into topics that may interest this specific group. Two things that I will not be covering are: animal movement analysis and geostatistics (ie., kriging). I also teach in base R so, no tidy! Well, very little tidy. This article “[A Thousand Gadgets: My Thoughts on the R Tidyverse](https://towardsdatascience.com/a-thousand-gadgets-my-thoughts-on-the-r-tidyverse-2441d8504433)” sums up nicely my feelings on tidy and why I do not teach it.

To modernize the course, I have moved all teaching materials to the new spatial classes and code-base from ASCII flat-files to HTML markup document so, you will simply need a web browser. The general way I want to structure things is for you to pair-off, yes work together, and rather than copy-and-pasting code I would like you to, at a minimum, re-write the provided code into an IDE. However, ideally you would attempt to work through the problem sets, using the provided code as guidance when stuck (or wanting to get some idea of what I am even asking). I provide hints as to what functions would address the problem. Mindlessly, copy-and-pasting code teachings you nothing but, you do need to see how it may be written (ah, the teaching conundrum). Except for the initial environment set up, the code block are collapsed so, you can open then up, take a quick peak and then hide them again. In the past I had one ASCII file with the problem sets and one with the answers. This single HTML format, with collapsible code blocks, seems more efficient so, please proved feedback.

**Syllabus**

**Foundations of R for spatial analysis - Pre-workshop (self paced)**

Background - Introduction to coding structure, object classes, data manipulation, writing functions, for loops and reading/writing spatial data.We will introduce the basic foundations of R logic and coding syntax, object-oriented structure, object classes and looping. Building on these basics, we will then cover data manipulation of the various spatial class objects introducing indexing, query, merging and subsetting data. We will also introduce creating, reading and writing vector spatial classes. For those wishing to improve their R skills in preparation for the workshop, some suggested foundation tutorial materials will be made available.

Objectives:

*Understanding vector, data frame, matrix, list object types and other object classes*

*Understanding base functions for manipulating and query of data.*

*Understanding bracket “indexing” and data manipulation.*

*Understanding the use of apply functions*

*Implementing “for/while” loops for repetitive task and simulation.*

*How to write functions*

*Creating, reading and writing spatial classes.*

**Section 1**

**Objectives -** Introduction to coding structure, object classes, data manipulation, writing functions, for loops and reading/writing spatial data. We will introduce the basic foundations of R logic and coding syntax, object-oriented structure, object classes and looping. Building on these basics, we will then cover data manipulation of the various spatial class objects introducing indexing, query, merging and subsetting data. We will also introduce creating, reading and writing vector spatial classes.

1.1 - Reading and writing of spatial classes

Reading shapefile vectors

Subset vector data

Write vector data

Read raster data

Read multi-band raster

Coercion of raster data

Write raster data

1.2 - Indexing and query of spatial vector classes

Create point vector

Subset rows of point vector

Create random sample

Subset using bracket query

Query to get percent

Aggregated statistics

Random sample polygons

Distance-based random sample

1.3 Functions

Step through and dissect function

Write observed mean distance function

1.4 Basic plotting of spatial objects

Plot attribute query

Plot nominal colors

Plot continuous colors

**Section 2**

Objectives - One seemingly limiting factor to full migration of spatial analysis in R is ability to implement basic GIS functions. We will cover various vector overlay procedures, distance and proximity analysis. We will focus on spatial data analysis, starting with data manipulation, query and overlay. We will also cover distance and proximity analysis thus, providing a foundation to topics covered later in the workshop (e.g., assessing spatial autocorrelation, network analysis). We will also cover basic raster analysis including single raster operators, overlay and focal functions. Finally, we will introduce concepts of quantifying landscape structure using landscape metrics.

2.1 - Vector analysis

Buffer points

Clip polygons

Calculate spatial area fractions

Point in polygon

Spatial aggregation (dissolve)

Spatial aggregation (zonal)

2.2 Raster data analysis

Reading raster data (single band and multi band)

Global raster statistics

Raster transformations

Multi-band statistics

Multi-band functions

Focal statistics

Focal functions

Reproject raster

2.3 - Raster/Vector integration

Extract raster values for points

Extract raster values for polygons

Aggregate polygon raster values

2.4 Quantifying landscape structure

Reclassifying to forest/non-forest

Smooth by calculating percent forest

Forest cores

Landscape-level metrics

Sample-level metrics

**Section 3**

Objectives - We will build foundation of more advanced spatial modeling by introducing proximity, distance and nearest neighbor analysis, then progressing to Point Pattern Analysis for quantifying spatial structure. Understanding spatial autocorrelation is fundamental for understanding spatial process, testing model assumptions and conducting spatial analysis. In this section, we will cover what is spatial autocorrelations, importance and implications of spatial autocorrelation in modeling ecological systems, how to assess spatial autocorrelation (global and local), and how to test model assumptions using spatial autocorrelation tests. Finally, we will cover basics of time-series analysis, moving towards practical applications on raster time-series data.

3.1 - Distance and proximity

Create distance matrix

Distance matrix - nearest neighbors

Distance matrix - conditional nearest neighbors

Graph structures - nearest neighbors

3.2 Spatial dependency - Point Pattern Analysis

Create spatstat object

Identify duplicates

Geits G(r) statistic

Ripley’s-K statistic

3.3 Spatial dependency - global autocorrelation

Spatial Weights Matrix (Wij)

Calculate Moran’s-I (global autocorrelation)

Calculate Geary’s-C (global autocorrelation)

Calculate Getis-Ord (global autocorrelation)

3.4 Spatial dependency - model assumptions

Create linear model

Moran’s-I on regression residuals

3.4 Spatial dependency - local autocorrelation

Local-G

Cross correlation algebraic approximation

3.5 - Time-series analysis

Subset single pixel time-series observation

Imputing missing values

Smoothing time-series

Smoothing time-series (confidence)

Create time-series class object

Specify trend model

Periodicity

3.6 - Raster time-series analysis

Temporal statistical aggregation

Rate of change

Mann-Kendall

Raster time-series trend model

**Graph-theoretic models in ecology**

Objectives - We will provide an overview to graph theoretical approaches with emphasis on gravity models. There are many ways graphs can be implemented to understand population structure and relate that structure to landscape characteristics (see Dyer and Nason 2004). In this exercise, we will focus on one specialized case. Gravity models are a type of inferential model that exploit graph characteristics. Gravity models include both at site (nodes) and between side (edges) landscape data. In this section, we will use the gravity model framework to build an empirical model of connectivity for a Columbia spotted frog dataset in central Idaho (Murphy et al. 2010).

Overview of Worked Example - Background: Gravity models are a type of inferential model that exploit graph characteristics. Gravity models include both at-site and between-site landscape data. They are a type of graph consisting of nodes and edges. These nodes and edges of landscape characteristics associated with these graph elements. There are many ways graphs can be implemented to understand population structure and relate that structure to landscape characteristics (see Dyer and Nason 2004). In this exercise, we will calculate various graph metrics and apply graphs to fit a gravity model. Data set: In this exercise, you will use the gravity model framework to build an empirical model of gene flow for the Columbia spotted frog dataset in central Idaho that you have used for several other exercises (Murphy et al. 2010).

Lecture introduction (video Dr. Melanie Murphy)

Wetland complex data preparation

Read in wetlands data

Create wetlands graph

Graph metrics

Plot graph metric

Field-data preparation

Read site data

Saturated Graph

1. Create graph from site locations

2. Merge the graph with genetic distance.

Spatial model data preparation

Read raster data using terra

Reclassify wetlands

Calculate the proportion of the landscape around sites

Challenge:

Add values of rasters to sample sites

Add raster covariates to graph edges (lines).

What about categorical variables?

Evaluate node and edge correlations

Add node data

1. Build node data

2. Merge nodes and edges

Gravity model

Develop hypothesis

Compare competing models.

Fit final model(s)

Back predict global\_fit model

Aggregate estimates and plot

**Probabilistic Binominal Random Forests**

Objectives - I will introduce the conceptual and mathematical foundations of recursive partitioning, ensemble and Random Forests methods along with implementation of a binomial (probabilistic) predictive model. In a probabilistic binomial instance, Random Forest does need a null [0,1]. In this example we will be using surveyed data representing presence/absence. However, when this is not the case methods must be employed to create a pseudo absence to act as the null. The simplest sample framework is a random or systematic sample. However, there are numerous issues with a null that is disconnected from the spatial process observed in the data. The pseudo.absence function in the spatialEco library uses an isotropic intensity estimate to create sample weights in generating a random sample. This incorporates a continuous gradient of the spatial process in the observed data.

In this exercise, we will leverage a spatial sample and raster data representing various hypothesized ecological processes associated with our species. We will employ a model selection procedure, evaluate the significance of the selected parameters, fit a final model, estimate the spatial uncertainty and perform model validation to evaluate fit and performance. Finally, we will explore inferential methods using partial dependence plots to evaluate functional relationships and take a brief look at Shapley analysis for individual and aggregated relationships.

Lecture introduction

Data preparation

Read shapefile vector and img rasters from disk

Extract raster values

Check data balance

Check for collinearity and multi-collinearity

Create data withhold

Specify probabilistic binomial model

Apply a model selection procedure

Evaluate selected parameter significance

Fit final model

Derive importance

Model validation

Confusion matrix based statistics

Log Loss

cross-validation

Sensitivity test

Prediction calibration

Spatial predictions

Subset rasters

Create prediction raster

Spatial uncertainty

Functional relationships and inference

Partial dependence probability plots

Shapley analysis - aggregated

Shapley analysis - individual (observation)

**Density (point process) Random Forests model**

Objectives - An introduction to density-based point process random forests models. Conceptually, a Poisson Point-process Model (PPM) uses point observations [X,Y] to fit an intensity process (surface) that is then used with matching covariates to estimate the Probability Density (PDF) or Intensity function. The intensity function is an estimate of the density following an assumption of a Poisson point process. We can emulate the PPM structure by simply estimating the intensity and then using random forests, rather than a linear model, to solve the regression equation. I would note that MaxEnt is not a presence-only model per se but, generates a large number of regular samples to act as the null (background) with the know observations embedded in the background sample. This is not dissimilar to a PPM, (Renner & Warton 2013 demonstrated this equivalency) but, MaxEnt does not rely on an intensity function, rather solving a probability distribution via maximized entropy.

Data preparation

Read shapefile vector and img rasters from disk

Create intensity process

Assign raster values

Check for collinearity and multi-collinearity

Specify Poisson (density) random forests model

Apply a model selection procedure

Fit final model

Derive importance

Validation

Understanding common metrics for continuous data (eg., root mean square error)

Spatial predictions

Subset rasters

Create prediction raster

Create uncertainty raster(s)

Compare density and probability results

**Instructors Biography**

Jeffrey S. Evans, Ph.D., is a Senior Landscape Ecologist and Biometrician with The Nature Conservancy and a Visiting Professor at University of Wyoming, where he attempts to bring vigor from diverse fields such as landscape ecology, spatial statistics, remote sensing, population genetics and applied mathematics to answer practical conservation questions. His focus is on the development and implementation of novel statistical methods for the assessment of cumulative impacts to ecological systems, understanding species limits, quantifying habitat use through patterns of genetic structure and monitoring biodiversity. With over 25 years of experience, he also has formal training in quantitative ecology and spatial statistics from UCSD and UC Berkeley with a tenure at the Berkeley-Stanford Joint Statistics program. Before moving to TNC, he worked as an Assistant Professor at University of Idaho in the USDA Forest Service Research Station Co-op unit. Prior to this he was a forest ecologist on the Plumas National Forest and a research fellow at the California Academy of Sciences. He currently has over 110 publications in peer-reviewed journals and several published R packages.

For downloading materials, If possible, I would like you to create a directory called “spatialR” directly under your root directory (ie., “C:\”). To download the materials and format the directory structure to be consistent with the code examples, run the attached code. These are not big files so, if anybody has issue we can trouble-shoot later.

# Set your root working directory for the workshop here

# Please create directory first!

setwd("C:/spatialR")

# Once working directory is set, run below code to download

# and uncompressed archives. This will result in a new directory

# for each workshop section (eg., "C/spatialR/session01" containing

# materials

# Session 01

download.file("https://spatialr.s3.us-west-2.amazonaws.com/session01.zip",

destfile="session01.zip", mode="wb")

unzip("session01.zip")

file.remove("session01.zip")

# Session 02

download.file("https://spatialr.s3-us-west-2.amazonaws.com/session02.zip",

destfile="session02.zip", mode="wb")

unzip("session02.zip")

file.remove("session02.zip")

# Session 03

download.file("https://spatialr.s3-us-west-2.amazonaws.com/session03.zip",

destfile="session03.zip", mode="wb")

unzip("session03.zip")

file.remove("session03.zip")

# Random Forests

download.file("https://spatialr.s3-us-west-2.amazonaws.com/randomForests.zip",

destfile="randomForests.zip", mode="wb")

unzip("randomForests.zip")

file.remove("randomForests.zip")

# Graph Theoretic

download.file("https://spatialr.s3-us-west-2.amazonaws.com/graph.zip",

destfile="graph.zip", mode="wb")

unzip("graph.zip")

file.remove("graph.zip")

AWS introduction materials

<https://spatialr.s3.us-west-2.amazonaws.com/introduction.zip>