

Using R with ArcGIS

Shaun Walbridge

<https://github.com/scw/r-devsummit-2016-talk>

Handout PDF

High Quality PDF (4MB)

Resources Section

Background Qs

ArcGIS

R

automation / ModelBuilder

programming

Data Science

Data Science

- A much-hyped phrase, but effectively is about the application of statistics and machine learning to real-world data, and developing formalized tools instead of one-off analyses. Combines diverse fields to solve problems.

Data Science

What's a data scientist?

“A data scientist is someone who is better at statistics than any software engineer and better at software engineering than any statistician.” — Josh Wills

Data Science

Us geographic folks also rely on knowledge from multiple domains. We know that spatial is more than just an x and y column in a table, and how to get value out of this data.

Geography has a similar relationship, domain knowledge on top of the spatial

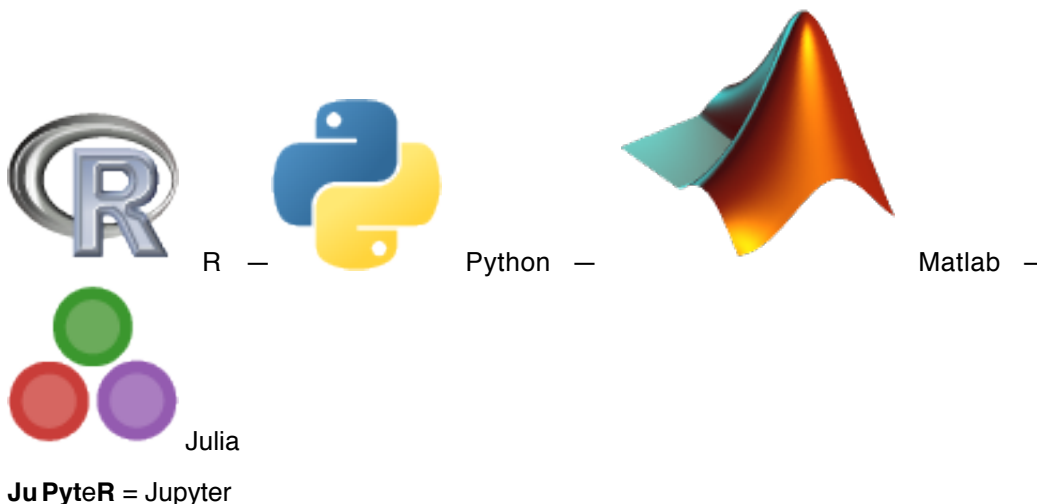
“A data scientist is a statistician who lives in San Francisco”.

Like the Geographic: a similar relationship between our domain and the knowledge we need which spans into other domains. Stats is similar – can’t do it without someone else’s data!

Goodchild bit: difference is for stats, methods came hundred years ago (e.g. Bayes Method), but only recently have we had the ability to actually compute it for hard problems. GIS is the other way around: data came first, we built up methods around it.

Scientific Languages

Languages commonly used in scientific and statistical problem solving:



Scientific Languages

We’re a big Python shop, so why R?

...

“Why can’t everyone just use *Python*?”

. . .

≈“Why can’t everyone just speak *English*?”

. . .

- More like dialects. We speak with our Canadian friends, right?
- Complementary in many workflows. People use both to get real work done.

Scientific Languages

R vs Python for Data Science

[really the case? just perhaps highlight these four, many others... SQL, classic languages, ...]

R



Why

- Powerful core data structures and operations
 - Data frames, functional programming
- Unparalleled breadth of statistical routines
 - The *de facto* language of Statisticians, state of the art statistical methods available
- A fast growing programming language in the past ~5 years
- CRAN: 8000 packages for solving problems
- Powerful language for creating high quality plots and graphics

...

- We assume basic proficiency programming
- See resources for a deeper dive into R

Share the essence of the language.

Open source – GPL

Written in C – some parts are very fast, others less so. R code is relatively pokey.

CRAN is epic. Get immediate access to best of breed methods, written by domain experts.



Why

- Open source. Dynamic language, both functional + object oriented
- CRAN is impressive. Best of breed methods, written by domain experts.
- Includes domain specific languages for statistics. E.g.:

```
fit.results <- lm(pollution ~ elevation + rain + ppm.nox + elevation:rain)
```
- Similar properties in other parts of the language

R Data Types

Data types you're used to seeing...

Numeric - Integer - Character - Logical - timestamp

...

... but others you probably aren't:

vector - matrix - data.frame - factor

R Data Types

Example source

$$\mathbf{A} = \begin{bmatrix} 4 & 3 & 8 \\ 7 & 1 & 5 \end{bmatrix}$$

Figure 1:

Vector:

```
a.vector <- c(4, 3, 8, 7, 1, 5)
```

Matrix:

```
A = matrix(  
  c(4, 3, 8, 7, 1, 5), # same data as above  
  nrow=2, ncol=3, # what's the shape of the data?  
  byrow=TRUE) # what order are the values in?
```

R Data Types

Data Frames:

- Treats tabular (and multi-dimensional) data as a labeled, indexed series of observations. Sounds simple, but is a game changer over typical software which is just doing 2D layout (e.g. Excel)

R Data Types

```
# Create a data frame out of an existing tabular source
df.from.csv <- read.csv("data/growth.csv", header=TRUE)
```

```
# Create a data frame from scratch
quarter <- c(2, 3, 1)
person <- c("Goodchild", "Tobler", "Krige")
met.quota <- c(TRUE, FALSE, TRUE)
df <- data.frame(person, met.quota, quarter)
```

```
R> df
  person met.quota quarter
1 Goodchild     TRUE      2
2   Tobler    FALSE      3
3    Krige     TRUE      1
```

Many packages define their own objects, conversion is an important step in any analysis dealing with higher order objects beyond simple data frames.

sp Types

- 0D: SpatialPoints
- 1D: SpatialLines
- 2D: SpatialPolygons
- 3D: Solid
- 4D: Space-time

Entity + Attribute model

Spatial types class for R. Solids and space time are both 'in development', nothing directly in sp but folks are working on this.

Also a raster package, but not covering this today.

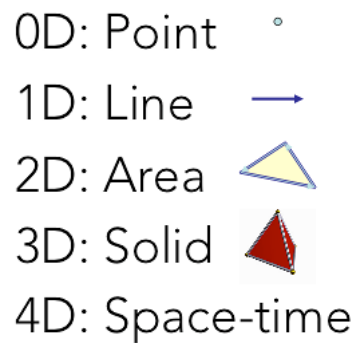


Figure 2:

Data Science with R

Hadley Stack

- Hadley Wickham
- Developer at R Studio, Professor at Rice University
- `ggplot2`, `scales`, `dplyr`, `devtools`, many others

Statistical Formulas

```
fit.results <- lm(pollution ~ elevation + rain + ppm.nox + elevation:rain)
```

- Domain specific language for statistics
- Similar properties in other parts of the language
- `caret` for model specification consistency

Literate Programming

I believe that the time is ripe for significantly better documentation of programs, and that we can best achieve this by considering programs to be works of literature. — Donald Knuth, “Literate Programming”

- packages: `RMarkdown`, `Roxygen2`
- Jupyter notebooks

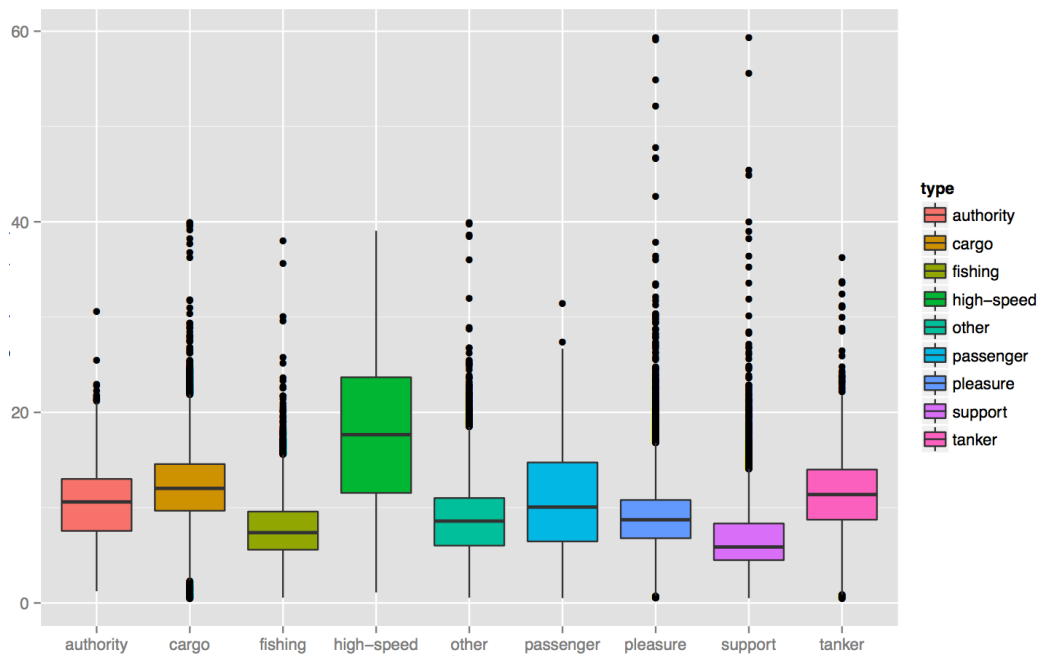


Figure 3:

What does this mean? You can interweave text with documentation fluidly, makes 'living documents' possible. Can have code embedded...

Development Environments





• *née IPython*

- R Tools for Visual Studio *brand new*

...

- Best of class tools for interacting with data.

dplyr Package

```
Batting %.%  
  group_by(playerID) %.%  
  summarise(total = sum(G)) %.%  
  arrange(desc(total)) %.%  
  head(5)
```

Introducing dplyr

In depth from Cam's workshop: `filter()` – Subset rows from a data frame. Similar in function to base R subsetting. `filter(crime_df, Arsons > 3, Thefts > 10)` `arrange()` – Sort rows in a data frame based on a set of column names. Can sort by multiple different columns. `arrange(crime_df, Arsons, Assaults)` `select()` – Select specified columns (or variables) from a data frame. `select(crime_df, AREA_S_CD, Equity_Score)` `summarize()` – Summarize values from a data frame given a function, and collapse results to a single row (unless data are grouped). `summarize(crime_df, mean_fire = mean(Fire.Vehicle.Incidents, na.rm = TRUE))` `summarize_each()` – Summarize values from a data frame given multiple functions. `summarize_each(crime_df, c('mean', 'sd'), Equity_Score) %>%` (Forward-pipe operator) – Allows you to pipe a value forward into an expression or function call, e.g., `f(x, y)` become `x %>% f(y)`. `crime_df %>% filter(Assaults == 0) %>% select(Equity_Score, Thefts) %>% arrange(Thefts)` `group_by()` – Group a data frame given a variable (or list of variables). Groups will be used when you apply functions to this data frame. `arson_groups = group_by(crime_df, Arsons)` `summarize(arson_groups, mean_fire = mean(Fire.Vehicle.Incidents, na.rm = TRUE))` Adding an underscore to the end of any of these functions (e.g., `arrange_()`) to be able to pass parameters as lists (or more so,

vectors). `sort_fields = c('Arsons', 'Thefts')` `arrange_(crime_df, .dots = sort_fields)`

R Challenges

- Performance issues
- Not a general purpose language
- Lacks purely UI mode of interaction (e.g. plots must be manually specified)
- Programmer only. There is `shiny`, but R is first and foremost a language that expects fluency from its users

R without underlying C code can be slow. More challenging, R is by design an in-memory language, and each operation creates a new in-memory copy of the data structure. Working with large files can be problematic, typically heavy R users invest in lots of RAM.

R — ArcGIS Bridge

Delicate Arch at Night: https://commons.wikimedia.org/wiki/File:Delicate_Arch_at_Night_%288708111489%29.jpg

R — ArcGIS Bridge

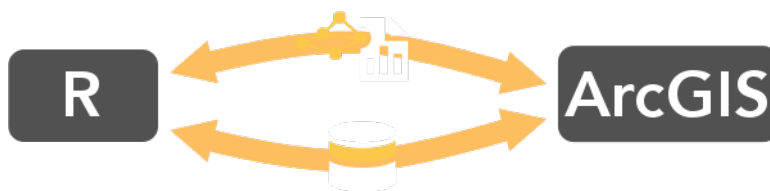


Figure 4:

- ArcGIS developers can create custom tools and toolboxes that integrate ArcGIS and R
- ArcGIS users can access R code through geoprocessing scripts
- R users can access organizations GIS' data, managed in traditional GIS ways

<https://r-arcgis.github.io>

The project serves three roles:

- Allows developers with experience with R and ArcGIS to create custom tools and toolboxes that integrate ArcGIS and R, both for their own use, and for building toolboxes to share with others both within their organization and with other ArcGIS users.
- R developers can quickly access ArcGIS datasets from within R, save R results back to ArcGIS datasets and tables, and easily convert between ArcGIS datasets and their equivalent representations in R via the sp package.
- Allows our users to integrate R into their workflows, without necessarily learning the R programming language directly.
- Building tools with ArcGIS and R
 - architecture + performance, round tripping data. sp objects.
 - example of R + GP workflow
 - building your own packages

R — ArcGIS Bridge

Store your data in ArcGIS, access it quickly in R, return R objects back to ArcGIS native data types (e.g. geodatabase feature classes).

Knows how to convert spatial data to sp objects.

Package Documentation

ArcGIS vs R Data Types

ArcGIS	R	Example Value
Address Locator	Character	Address Locators\\MGRS
Any	Character	
Boolean	Logical	
Coordinate System	Character	"PROJCS[\"WGS_1984_UTM_Zone_19N\"]. . .
Dataset	Character	"C:\\workspace\\projects\\results.shp"
Date	Character	"5/6/2015 2:21:12 AM"
Double	Numeric	22.87918

ArcGIS vs R Data Types

ArcGIS	R	Example Value
Extent	Vector (xmin, ymin, xmax, ymax)	c(0, -591.561, 1000, 992)
Field	Character	
Folder	Character	full path, use with e.g. <code>file.info()</code>
Long	Long	19827398L
String	Character	
Text File	Character	full path
Workspace	Character	full path

Access ArcGIS from R

Start by loading the library, and initializing connection to ArcGIS:

```
# load the ArcGIS-R bridge library  
library(arcgisbinding)  
# initialize the connection to ArcGIS. Only needed when running directly from R.  
arc.check_product()
```

Access ArcGIS from R

Opening data has two stages, like data cursors:

- Open data source with `arc.open`
- Select with filtering with `arc.select`

Similar to using `arcpy.da cursors`

Access ArcGIS from R

First, select a data source (can be a feature class, a layer, or a table):

```
input.fc <- arc.open('data.gdb/features')
```

Then, filter the data to the set you want to work with (creates in-memory data frame):

```
filtered.df <- arc.select(input.fc,  
                          fields=c('fid', 'mean'),  
                          where_clause="mean < 100")
```

This creates an *ArcGIS data frame* – looks like a data frame, but retains references back to the geometry data.

Access ArcGIS from R

Now, if we want to do analysis in R with this spatial data, we need it to be represented as sp objects. `arc.data2sp` does the conversion for us:

```
df.as.sp <- arc.data2sp(filtered.df)
```

`arc.sp2data` inverts this process, taking sp objects and generating ArcGIS compatible data frames.

Access ArcGIS from R

Finished with our work in R, want to get the data back to ArcGIS. Write our results back to a new feature class, with `arc.write`:

```
arc.write('data.gdb/new_features', results.df)
```

Access ArcGIS from R

WKT to proj.4 conversion:

```
arc.fromP4ToWkt, arc.fromWktToP4
```

Interacting directly with geometries:

```
arc.shapeinfo, arc.shape2sp
```

Geoprocessing session specific:

```
arc.progress_pos, arc.progress_label, arc.env (read only)
```



Figure 5:

The screenshot shows a software interface for configuring a Semiparametric Regression model. At the top, there is a back arrow, the title "Semiparametric Regression", and a hamburger menu icon. Below the title, there are two tabs: "Parameters" (selected) and "Environments", with a help icon to the right. The main area contains several configuration sections, each with a red asterisk icon: "Input Features" with a dropdown menu and a yellow plus icon; "Locations To Predict" with a dropdown menu and a yellow plus icon; "Dependent Variable" with a text input field; "Output Prediction Feature Class" with a text input field and a yellow plus icon. Below these are two sections for "Linear Explanatory Variables" and "Nonlinear Explanatory Variables", each with a "Select All" button and a refresh icon. The "Nonlinear Explanatory Variables" section also has an information icon. At the bottom, there are "Input Knot Features" with a dropdown menu and a yellow plus icon, and "Output Graphs" with a text input field and a yellow plus icon. A "Run" button with a play icon is located at the bottom right of the interface.

Figure 6:

Building R Script Tools

Building R Script tools



Figure 7:

```
tool_exec <- function(in_params, out_params) {  
  # the first input parameter, as a character vector  
  input.features <- in_params[[1]]  
  
  # alternatively, can access by the parameter name:  
  input.input <- in_params$input_features  
  print(input.dataset)  
  # ... next, do analysis steps  
  
  # this will be returned as the "Output Graphs" parameter.  
  out_params[[1]] <- plot(results.dataset)  
  return(out_params)  
}
```

R ArcGIS Bridge Demo

- Details of model based clustering analysis in the R Sample Tools

The How and Where

How To Install

- Install with the R bridge install
- Detailed installation instructions

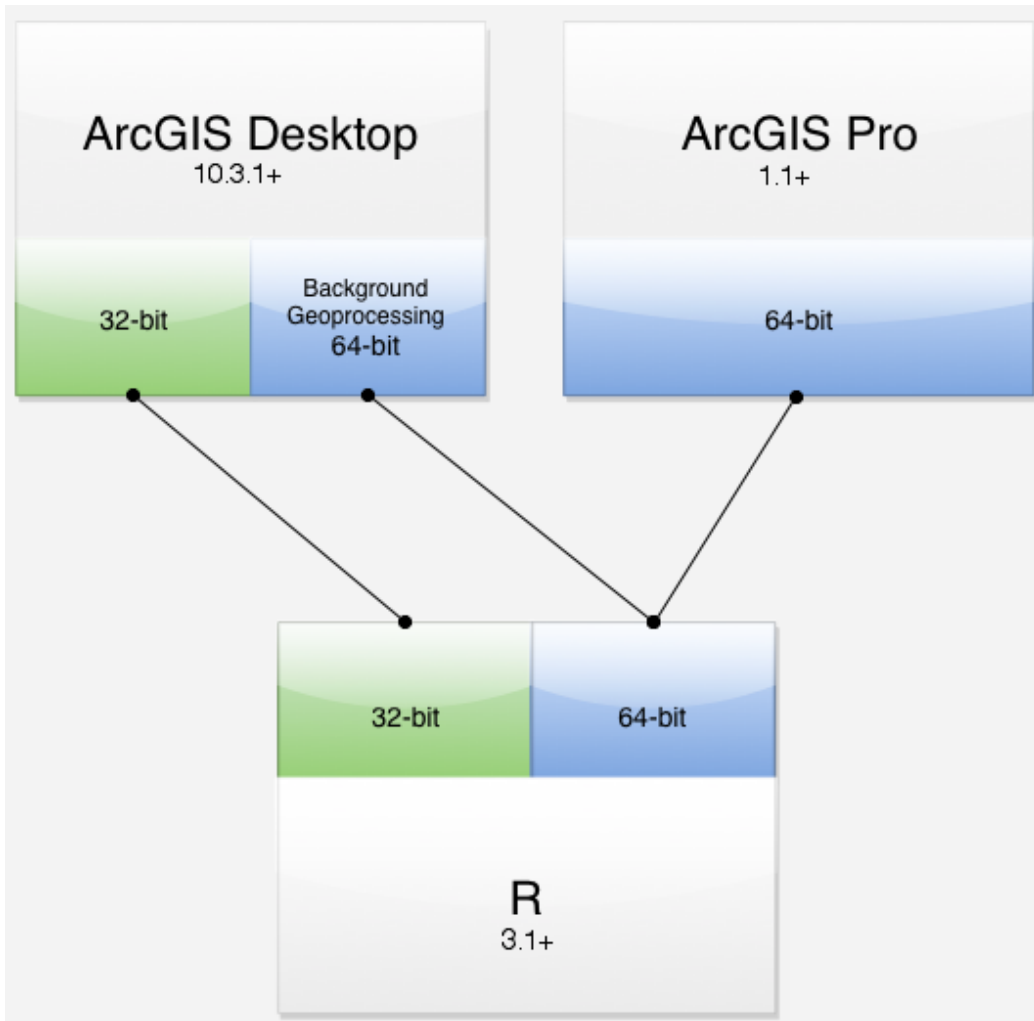


Figure 8:

Where Can I Run This?

Where Can I Run This?

- Now:
 - First, install R 3.1 or later
 - ArcGIS Pro (64-bit) 1.1 or later
 - ArcGIS 10.3.1 or later:
 - * 32-bit R by default in Desktop
 - * 64-bit R available via Server and Background Geoprocessing
- Upcoming:
 - Conda for managing R environments

32-bit version required for ArcMap, 64-bit version required for ArcGIS Pro (Note: the installer installs both by default).

64-bit version can be used with ArcGIS Pro, or with ArcMap by installing Background Geoprocessing and configuring scripts to run in the background. NOTE: Background Geoprocessing only allows using the bridge from ArcGIS, not from within R itself.

possible future improvements: - Conda for managing R environments - raster support

Resources

R

Looking for a package to solve a problem? Use the CRAN Task Views.

Tons of good books and resources on R available, check out the RSeek engine to find resources for the language which can be difficult to locate because of the name.

R Packages by Hadley Wickham

Spatial R / Data Science

- An Introduction to Statistical Learning (PDF) website A free and accessible version of the classic in the field, *Elements of Statistical Learning*.
- Getting Started in Data Science

ArcGIS + R

- Cam Plouffe (Esri CA) gave a two-part workshop that wrapped up yesterday, find out more in this post
 - Integrating R with ArcGIS: Part One
- Getting Data Science with R DevSummit talk this is one based on

Courses

Courses:

- High Performance Scientific Computing
- The Data Scientist's Toolbox
- A number of them on Coursera – useful topics even if you don't plan on using R

Books

- Spatial Statistical Data Analysis for GIS Users Konstantin Krivoruchko (GA creator)
 - Too big to print. Tons of useful stuff, covers both R and ArcGIS extensively.
- Practical data science with R
- Advanced R
- Applied Spatial Data Analysis with R
- Machine Learning with R

R ArcGIS Extensions

- R ArcGIS Bridge
- Marine Geospatial Ecology Tools (MGET)
 - Combines Python, R, and MATLAB to solve a wide variety of problems
- Geospatial Modeling Environment
 - An R flavored language for spatial analysis

Conferences

- useR! Conference
 - useR 2016 is being held at Stanford June 27-30

- Open Data Science Conference (ODSC)
 - Many happening around world, some upcoming ones:
 - ODSC East May 20-22 in Boston
 - ODSC West Nov 4-6 in Santa Clara

Closing

Outreach

- Resources and outreach – connect the dots, want this to be outreach so we can build up more R + ArcGIS people who aren't as common as our core language folks.
- Future of the project, questions

Community

- Open source project, different ethos
- Contributions are the currency
 - That said, major uptake in the commercial space:
 - Microsoft R (bought Revolution Analytics); R Studio
- Our involvement:
 - Recently hosted a Space-time Statistics Summit
 - More soon

Thanks

- R team: Dmitry Pavlushko, Shaun Walbridge, Steve Kopp, Mark Janikas, Konstantin Krivoruchko
 - Contact Us
- Geoprocessing Team

fin