Statistical Spatial Data Analysis with R and ArcGIS

Orhun Aydin, Mark Janikas, Marjean Pobuda, Shaun Walbridge

ESRI USER CONFERENCE

Introductions

R-ArcGIS Bridge Team

Seminar Overview

Today’s Schedule

Sections, Topics, and Timing

Section	Topic	Time Frame
Section 1	Introduction to R and the R-ArcGIS Bridge	8:30am - 10:00am
Break	Coffee & Tea	10:00am - 10:15am
Section 2	The In's and Out's of Bridge Functionality and Microsoft R	10:15am - 11:30am
Section 3 (part 1)	Introducing R Script Tools	11:30am - 12:00pm
Lunch	Tasty Eats	12:00pm - 1:00pm
Section 3 (part 2)	A Deeper Dive into R Script Tools	1:00pm - 2:00pm
Section 4	Sharing R Functionality	2:00pm - 3:00pm
Break	Coffee & Tea	3:00pm - 3:15pm
Section 5	Hands-on Practice Session	3:15pm – 4:30pm
Conclusion	Resources and Final Questions	4:30pm – 5:00pm

The R Language

A programming language for data analysis

Why Use R?

The R Language

- Where did it come from?
  - S, from Bell Labs
  - 1976
  - R released 1993

Why Use R?

The R Language

- R is a free and open-source programming language.
  - Designed with data analysis in mind
    - Powerful core data structures and operations
- Unparalleled breadth of statistical routines
  - Both cutting-edge research and field specific methods
- CRAN: Over 12,700 available packages (doubled since 2016!)
- Versatile and powerful plotting

Why Use R?

The R Language

- R is a free and open-source programming language.
  - Designed with data analysis in mind
    - Powerful core data structures and operations
- Unparalleled breadth of statistical routines
  - Both cutting-edge research and field specific methods
- CRAN: Over 12,700 available packages (doubled since 2016!)
- Versatile and powerful plotting

Language features: core types

The R Language

- Data types you're used to seeing:
  - `Numeric - Integer - Character - Logical - timestamp`
- But others that you may not be:
  - `vector - matrix - data.frame - factor`

Language features: data frames

The R Language

- 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)
- This concept of the data frame is core, and matches nicely to how data is manipulated and managed in geographic contexts

Why Use R?

The R Language

- Reproducible research and results
  - Easy to share
    - R Markdown
    - Jupyter Notebooks
    - Shareable R Script Tools

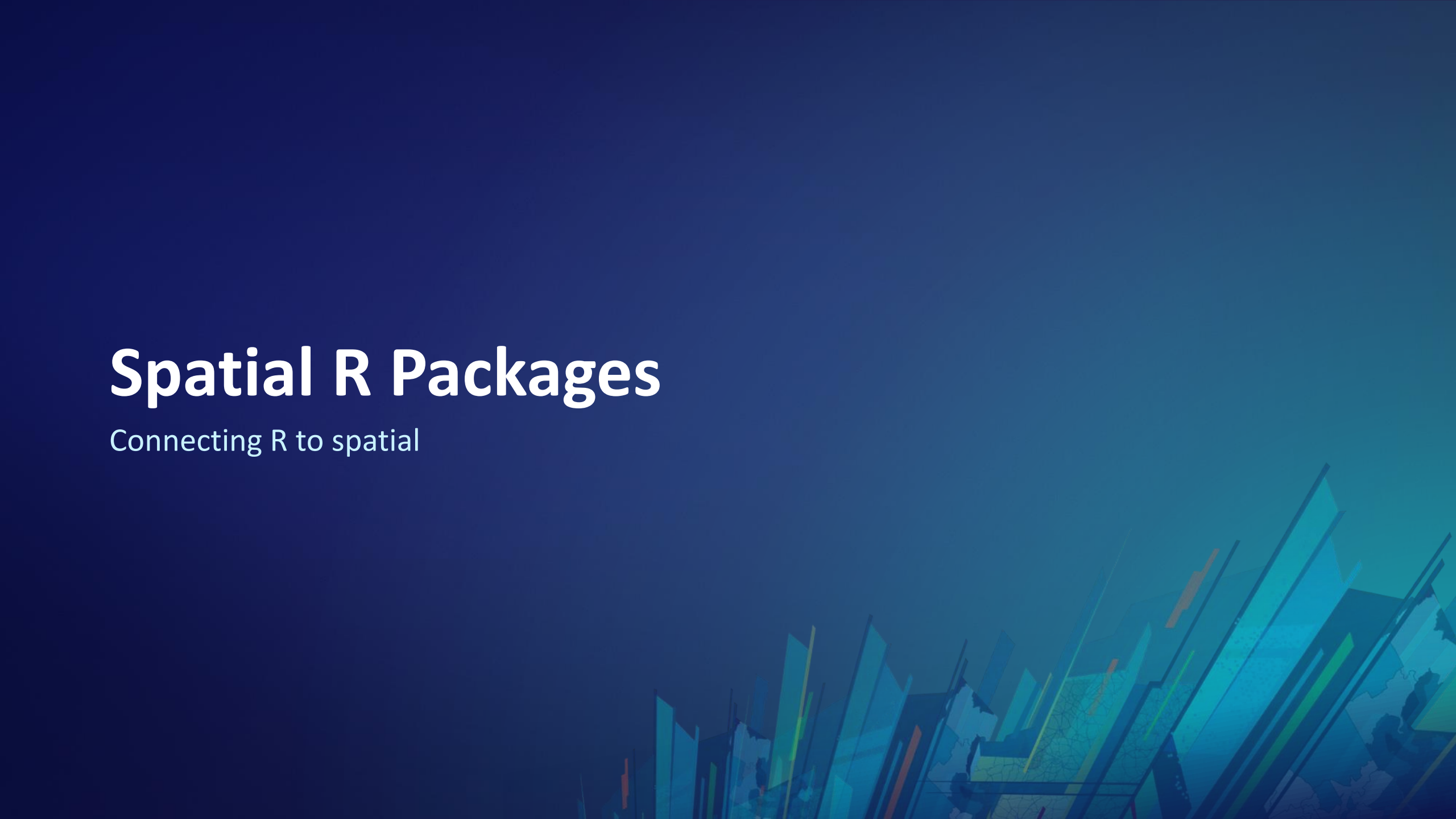
R Resources

The R Language

- A strong and engaged user community
  - Easy to find tutorials and resources for help
    - Cross-Validated - https://stats.stackexchange.com/
    - R-bloggers - https://www.r-bloggers.com/
    - Kaggle - https://www.kaggle.com/competitions
    - Hadley Wickham's Advanced R - http://adv-r.had.co.nz/
    - Introduction to Statistical Learning with Applications in R - http://www-bcf.usc.edu/~gareth/ISL/
    - Applied Spatial Data Analysis with R - http://www.asdar-book.org/
  - A variety of active user groups
    - R Consortium - https://www.r-consortium.org/
    - Meetup Communities – https://www.meetup.com/find/tech/
    - R-Ladies - https://rladies.org/

Spatial R Packages

Connecting R to spatial



R Spatial Packages

Connecting R to spatial

- Spatial data has a strong support framework
  - Well-defined classes and methods for vector and raster data
  - Consistent handling across the language
  - Functions for simple tasks, like plotting, subsetting, and summarizing spatial objects
  - Functions for advanced, niche spatial analyses
  - Interfaces well with ArcGIS

R Spatial Packages

Connecting R to spatial

- Working with feature data (points, lines, areas):
  - Historically `sp` has been the main package for working with feature data
  - New effort going into the `sf` package, which contains more extensive integration, and a richer underlying data model
  - 100s of packages have interop
- Working with raster data:
  - GDAL and `rgdal`
  - `raster` package
  - Many others with specialized roles have interop
- https://www.r-spatial.org/projects/

Conda

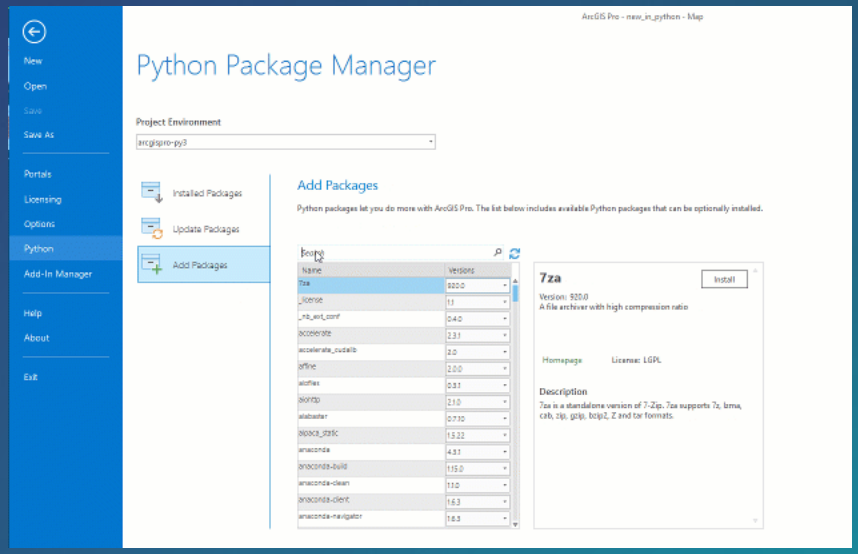
An Introduction

Managing Software Complexity

Conda

- Software is composed of many smaller components, often called *packages* or *libraries*.
- But, sharing code is a hard problem. Do you have the same packages of the same versions as the developer did?

- Conda solves this by providing easy access to install *packages* of software, and managing their interactions with a dependency solver
- It also creates *environments* — isolated collections of packages independent from the rest of the system



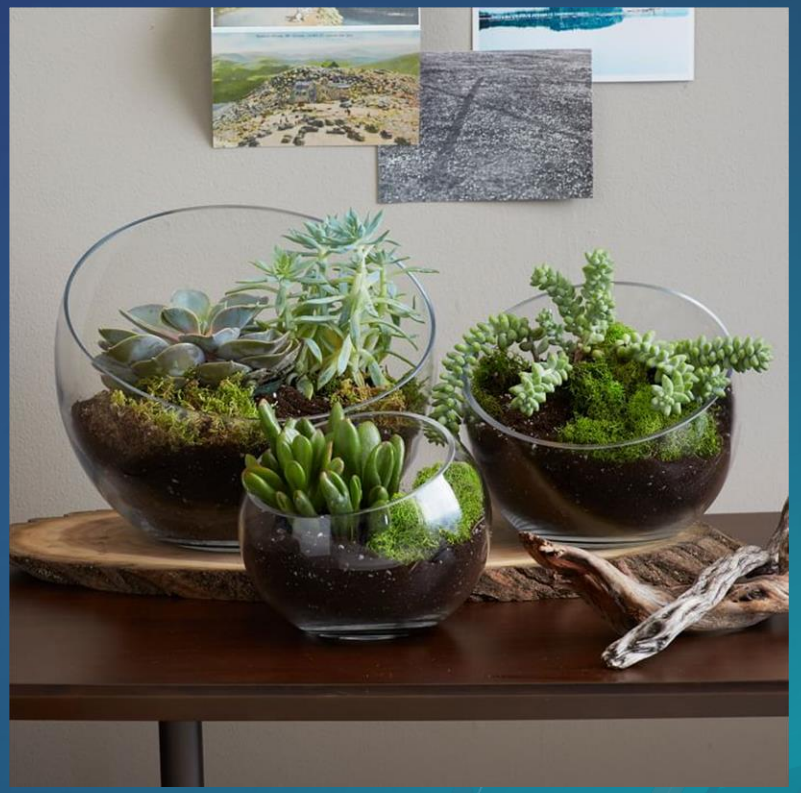
ArcGIS Pro includes Conda and a UI for interacting with it

CONDA

Expand what is possible with packages and environments

Conda

- Handle Python, R and many other languages and use cutting edge software from diverse fields (stats, AI, ML + many others)
- Works with R, Microsoft R, R-Studio and complex hybrid stacks (deeper look in Section 4)
- Command line tool, GUI available in Pro
- `arcgispro-r` environment on machines today for you to try out, and an intro to conda notebook



Terraria, isolated environments

CONDA

Jupyter Notebooks

An Introduction

Notebooks and Interactive Computing

Jupyter Notebooks

- Intermingle code, text, plots and more into a single notebook, run in a browser – *interactive computing*
- New modality: great for sharing, exploration, and writing with code
- We use it today to help guide you through examples, but if you're interested, can show you more in hands-on sections

Hands-On Setup

Jupyter Notebooks

1. Visit https://github.com/R-ArcGIS/UC-2018
2. Click green “Clone or Download” button, then “Download Zip”
3. Using File Explorer, Copy the zip file to C:\Workspace, replace existing file
4. Still in Explorer, right click, select “Extract All”, accept defaults. Should now see a folder, C:\workspace\UC-2018-master that contains “Extras-An-Introduction-To-Conda.ipynb”
5. Click Start Menu, Search for “Python Command Prompt”, click to open
6. `activate arcgispro-r`
7. `conda upgrade jupyter_console jupyter_core`, press enter to accept
8. `jupyter notebook C:\workspace\UC-2018-master`

Hands-On Setup

Jupyter Notebooks

https://github.com/R-ArcGIS/UC-2018

- Marjean Pobuda – mpobuda@esri.com
- Orhun Aydin - oaydin@esri.com
- Mark Janikas – mjanikas@esri.com
- Shaun Walbridge – swalbridge@esri.com

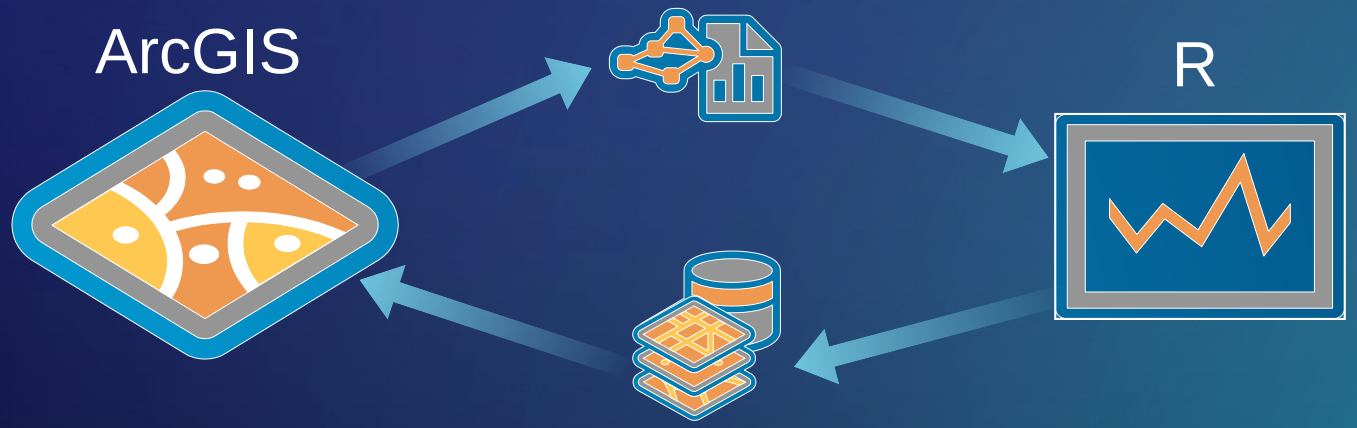
The R-ArcGIS Bridge

Some Background

Introducing the R-ArcGIS Bridge

The R-ArcGIS Bridge

- The R-ArcGIS bridge allows you to connect ArcGIS to R and enables the seamless transfer of data back and forth, along with the ability to integrate R and ArcGIS functionality.



Who Can Use the R-ArcGIS Bridge?

The R-ArcGIS Bridge

ArcGIS users

R users

ArcGIS developers

Vector Support

The R-ArcGIS Bridge

- Ability to read and write vector data
- Support for key R objects and spatial packages
  - R data frame object
  - Compatibility with **sp**
  - Compatibility with **sf**
- Customize data manipulations
  - Craft SQL queries to make selections
  - Subset by specific columns
  - Reproject data as needed
- Maintain spatial geometries when working with **dplyr**

Points

Lines

Polygons

Reading your ArcGIS vector data: part 1

arcgisbinding Package Functionality

- 1) Open ArcGIS data, tables, layers


```
gis_data <- arc.open(path = 'C:/Data/Data.gdb/AfricanBuffalo')
```

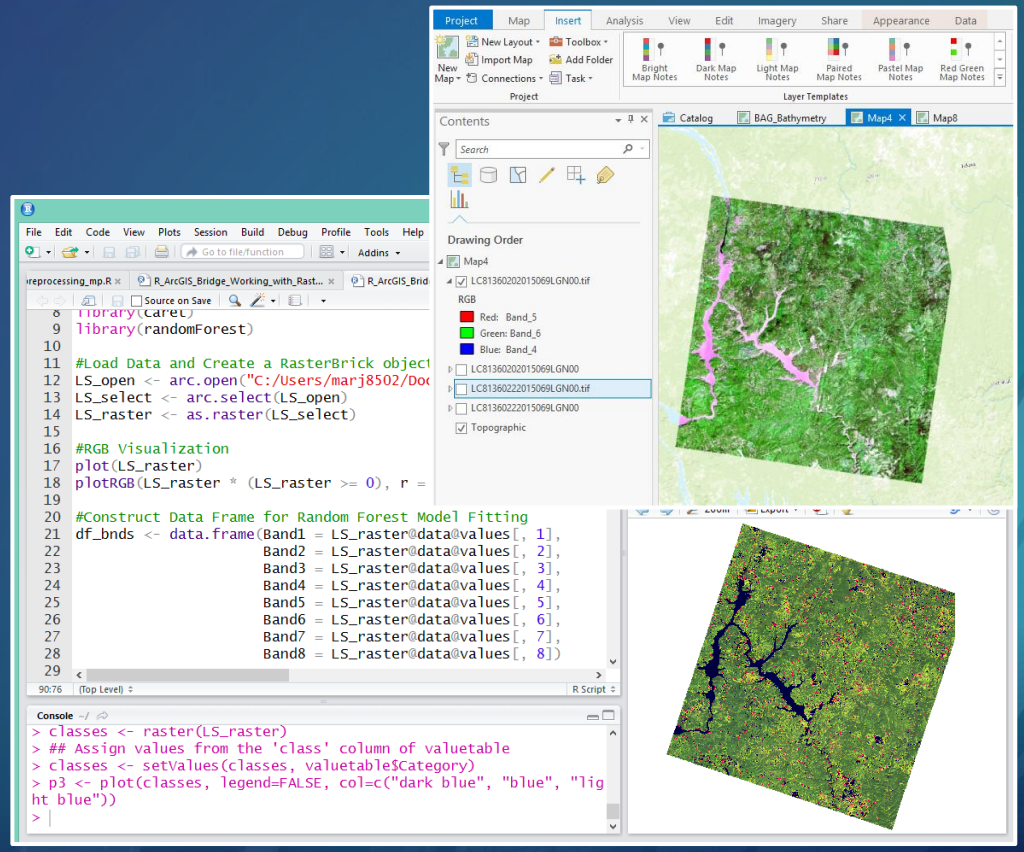
- 2) Load dataset to R data frame

```
r_data <- arc.select(gis_data, fields, SQL, spatial ref)
```

Raster Support

The R-ArcGIS Bridge

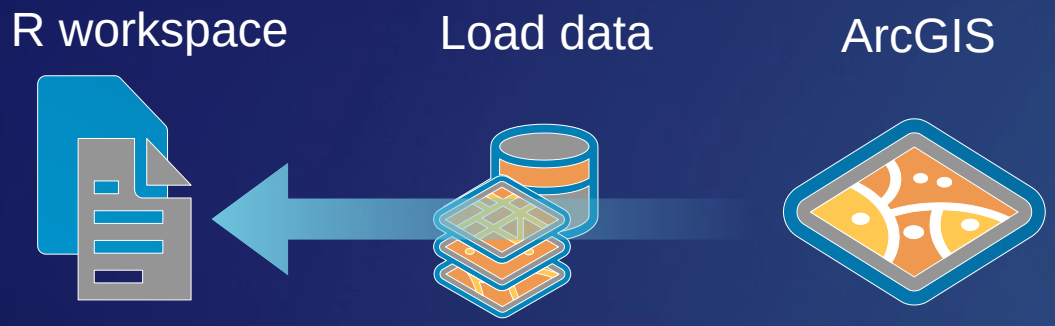
- Ability to read and write raster data
  - Handle big data raster data with the ability to read in chunks by bands
  - Compatibility with CRF format and Mosaic Datasets
- Customize selections and subsets
  - Create subsets by bands or pixel rows and columns
  - Resample options available
  - Select desired pixel format for specific analyses



Reading your ArcGIS raster data: part 1

arcgisbinding Package Functionality

- 1) Open ArcGIS single band or multiband rasters:




```
gis_data <- arc.open(path = 'C:/Data/MyRaster.tif')
```

- 2) Customize data details

```
arc_data <- arc.raster(gis_data, nrow, ncol, bands,
                       extent, pixel_type, resample_type)
```

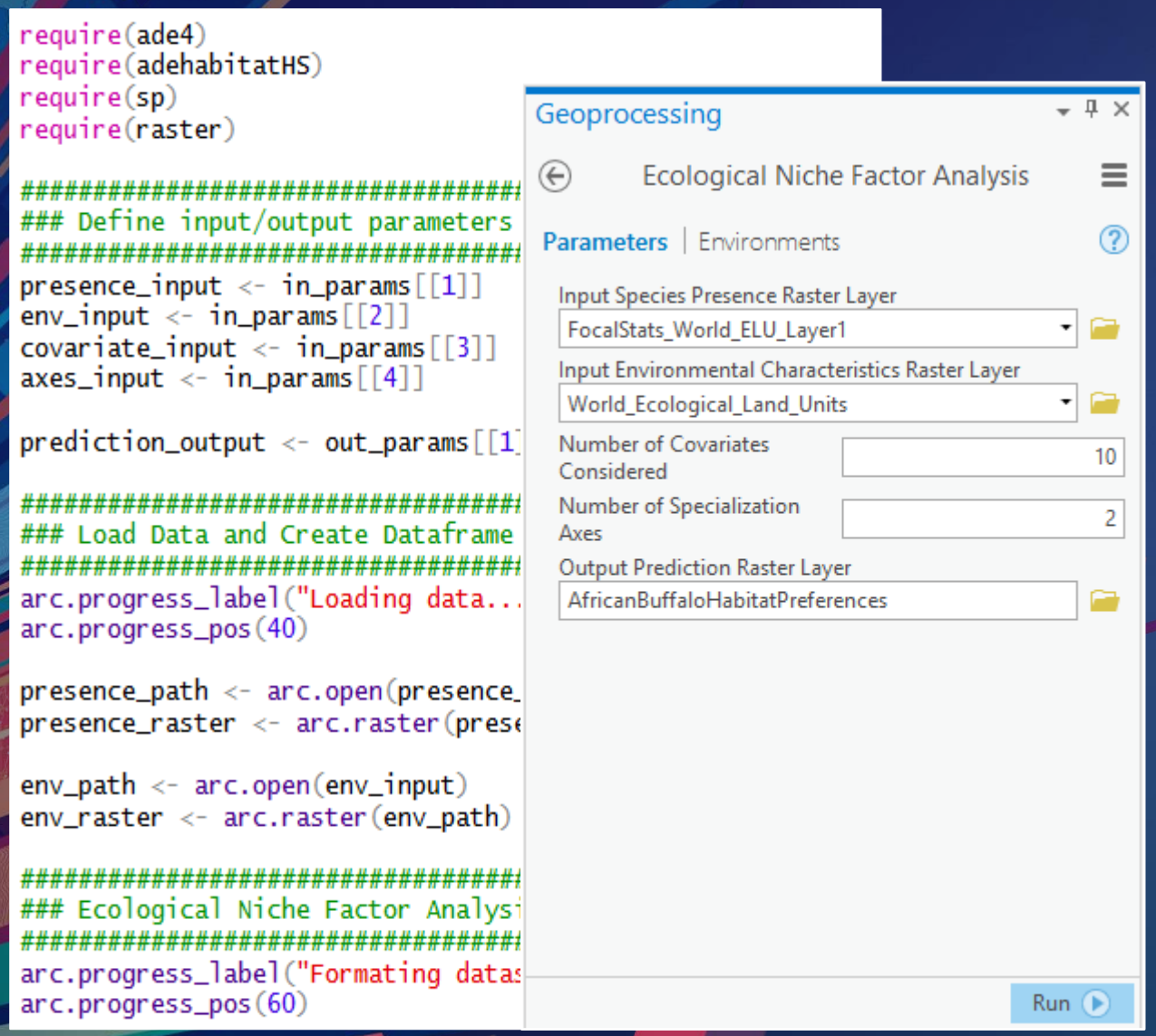
Big Data Support

The R-ArcGIS Bridge

- Supports all versions of Microsoft R
  - Microsoft Open
  - Microsoft Client
  - Microsoft Server

Working With The Bridge

A Guide to The Various Ways



Ways to Work with the Bridge

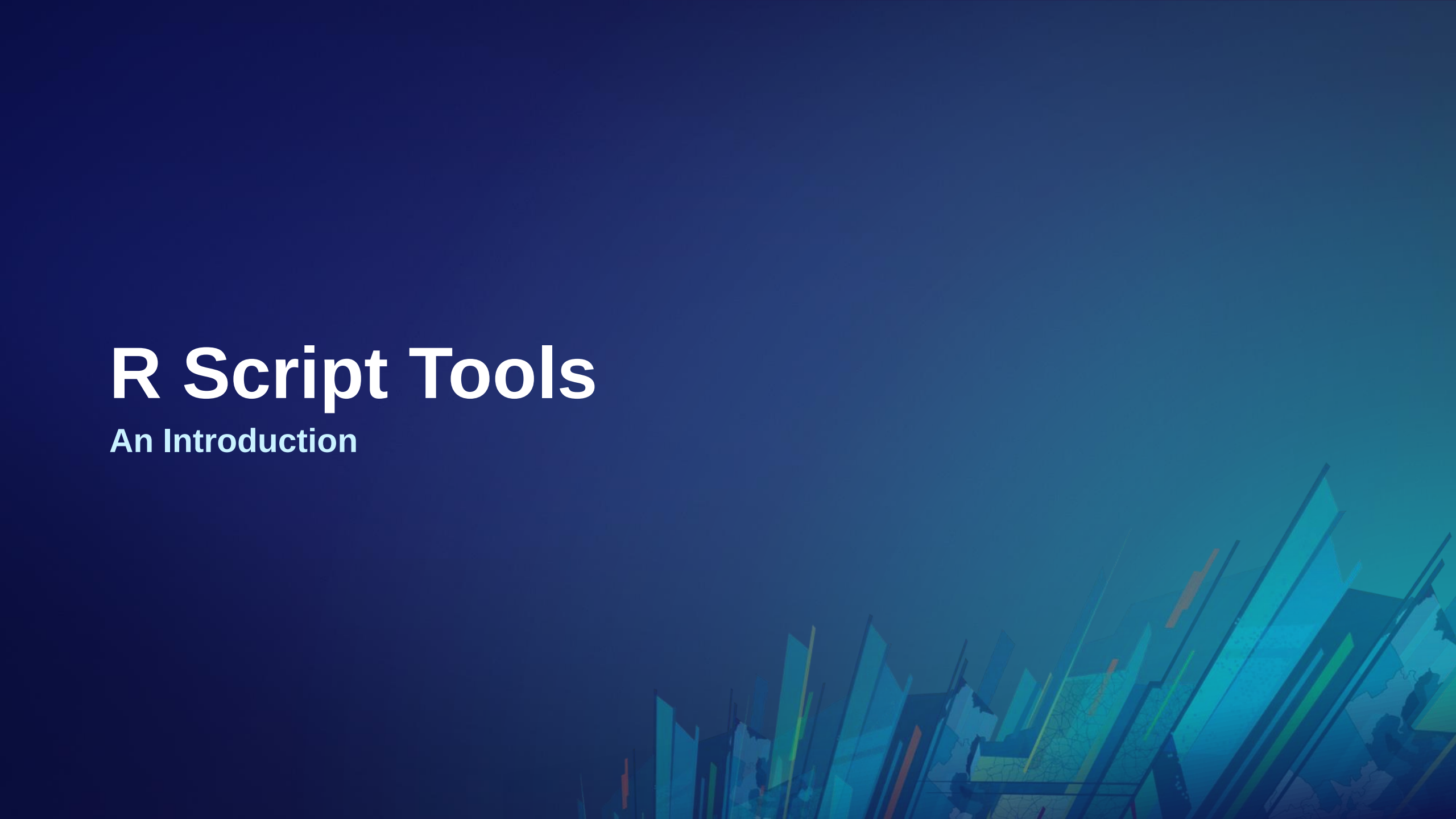
Demo - Shaun Walbridge

Break

Resuming at 10:15 am

R Script Tools

An Introduction

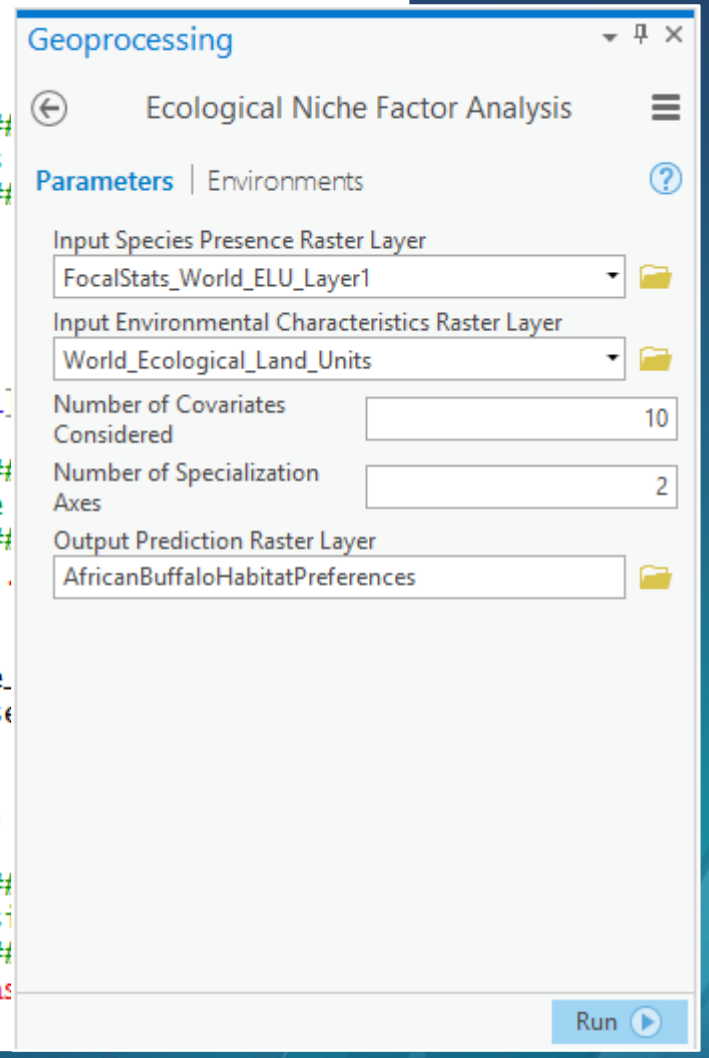


Why Create Script Tools

R Script Tools

- Automation
- Reproducibility
- Sharing
- Ease of use, accessibility
- Empowering for the non-R pro

```
require(ade4)
require(adehabitatHS)
require(sp)
require(raster)

#####################################
### Define input/output parameters
#####################################
presence_input <- in_params[[1]]
env_input <- in_params[[2]]
covariate_input <- in_params[[3]]
axes_input <- in_params[[4]]

prediction_output <- out_params[[1]

#####################################
### Load Data and Create Dataframe
#####################################
arc.progress_label("Loading data...
arc.progress_pos(40)

presence_path <- arc.open(presence_
presence_raster <- arc.raster(prese

env_path <- arc.open(env_input)
env_raster <- arc.raster(env_path)

#####################################
### Ecological Niche Factor Analysi
#####################################
arc.progress_label("Formating datas
arc.progress_pos(60)
```



Create Script Tools in R

R Script Tools

Generic, reusable script template

Wrapping functionality

```
tool_exec <- function(in_params, out_params)
             {
             }
```

Defining input and output parameters

```
animal_presence_input <- in_params [[1]]
environmental_input <- in_params [[2]]
covariate_input <- in_params [[3]]
axes_input <- in_params [[4]]

prediction_output <- out_params [[1]]
```

Create Script Tools in ArcGIS

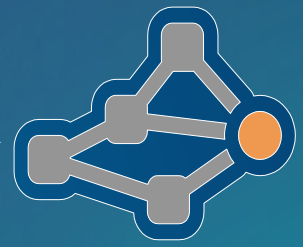
R Script Tools

1) Create toolbox

2) Add R script

3) Set parameters

4) Perform analysis



How to Create a Script Tool

The R script behind the scenes

```
require(ade4)
require(adehabitatHS)
require(sp)
require(raster)

#########################################################
### Define input/output parameters
#########################################################
presence_input <- in_params[[1]]
env_input <- in_params[[2]]
covariate_input <- in_params[[3]]
axes_input <- in_params[[4]]

prediction_output <- out_params[[1]]

#########################################################
### Load Data and Create Dataframe R Object
#########################################################
arc.progress_label("Loading data...")
arc.progress_pos(40)

presence_path <- arc.open(presence_input)
presence_raster <- arc.raster(presence_path)

env_path <- arc.open(env_input)
env_raster <- arc.raster(env_path)

#########################################################
### Ecological Niche Factor Analysis Data Formating
#########################################################
arc.progress_label("Formating datasets...")
arc.progress_pos(60)
```

Essential R Script Tool Components

Demo – Shaun Walbridge

Microsoft R and R-ArcGIS Bridge

Patterns and best practices

Using the R-ArcGIS Bridge with Microsoft R

Microsoft Open R

- **Microsoft Open R is a publicly available R-version**
- **Contains almost all CRAN libraries**
  - **It lags CRAN in functionality, Microsoft follows CRAN releases to update**
- **Provides integration to other Microsoft tools such as R-Server**

Using the R-ArcGIS Bridge with Microsoft R

ArcGIS Bridge- Microsoft R Connection

- **Connection to argisbinding package is same as CRAN version of R**
- **Can be used as the background R version within ArcGIS Pro**
- **Usage from Pro is exactly the same as CRAN**

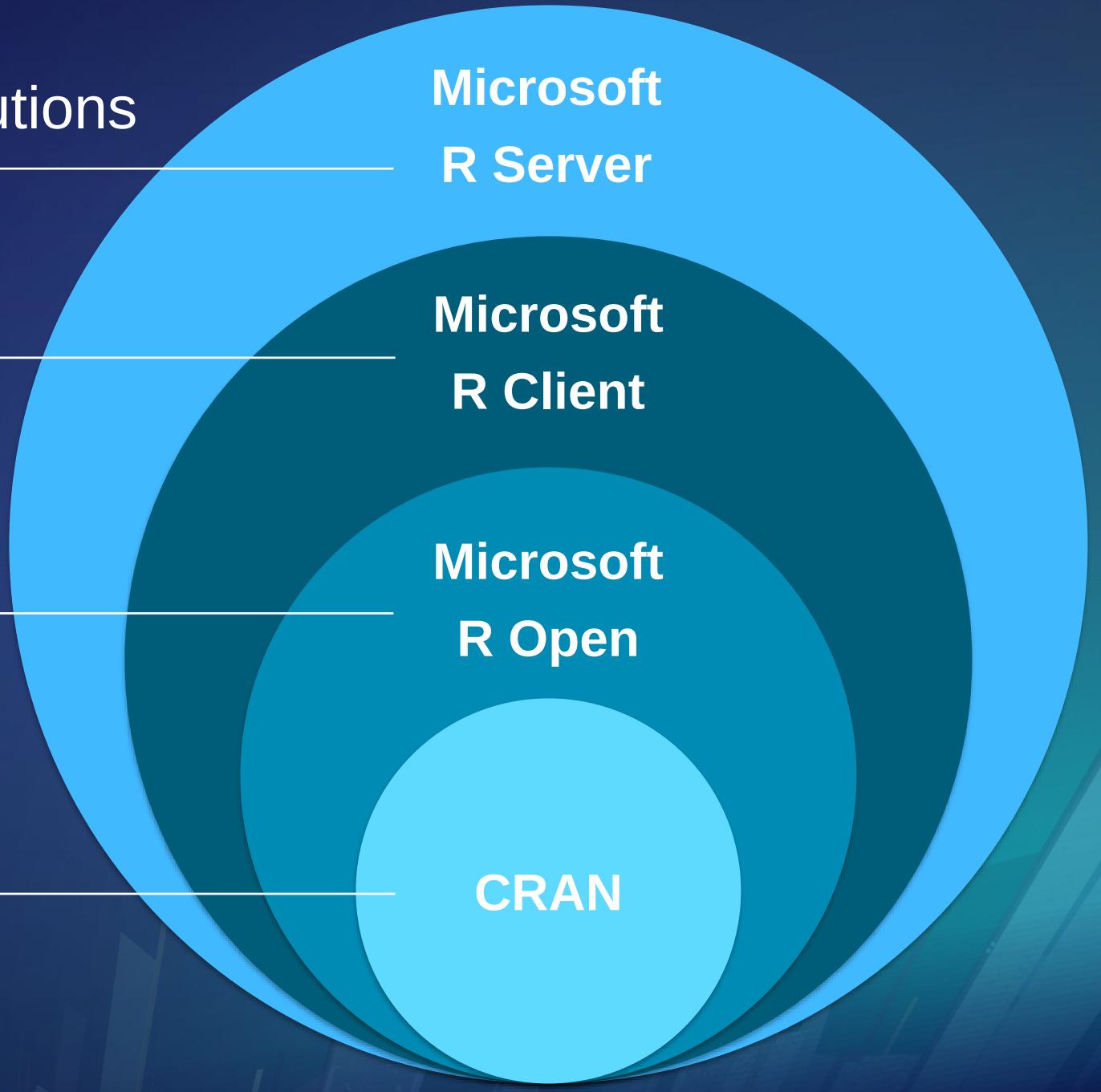
Comparing Different R Distributions

- Allows multi-thread parallelization

- Allows parallelization and remote computing
- Processing limited to 2 threads
- All processing is handled locally

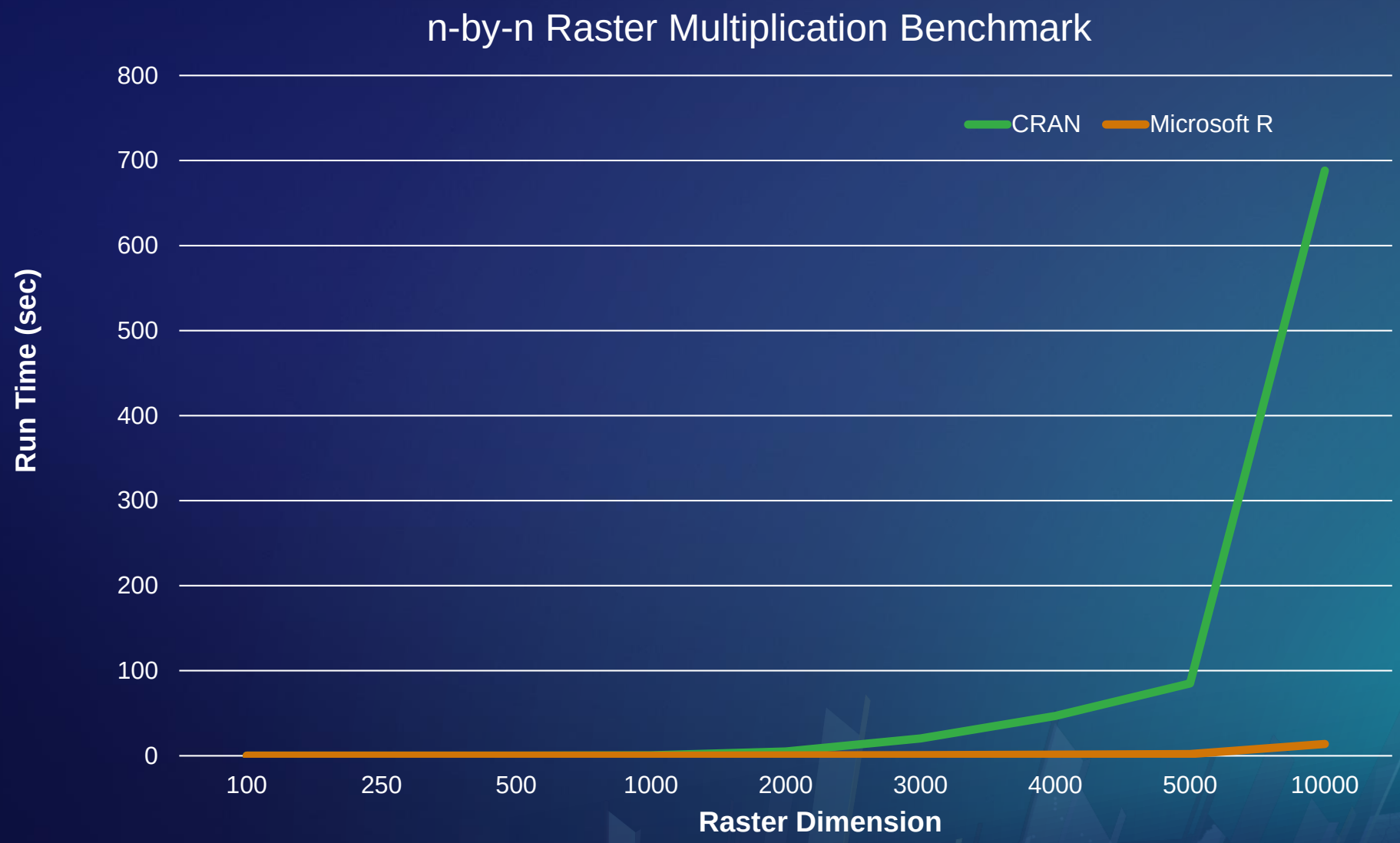
- Microsoft's implementation of R
- Publicly available
- Efficient matrix operations

- Open-source
- Contains newest libraries



Matrix Multiplication Benchmark on Rasters

Comparing Open R to CRAN R



Using the R-ArcGIS Bridge

Microsoft R

- **Image convolutions are matrix multiplications**
- **Window-based operations and image operators speed up drastically**
- **Integrates to bigger data platforms of Microsoft such as Azure and R-Server**

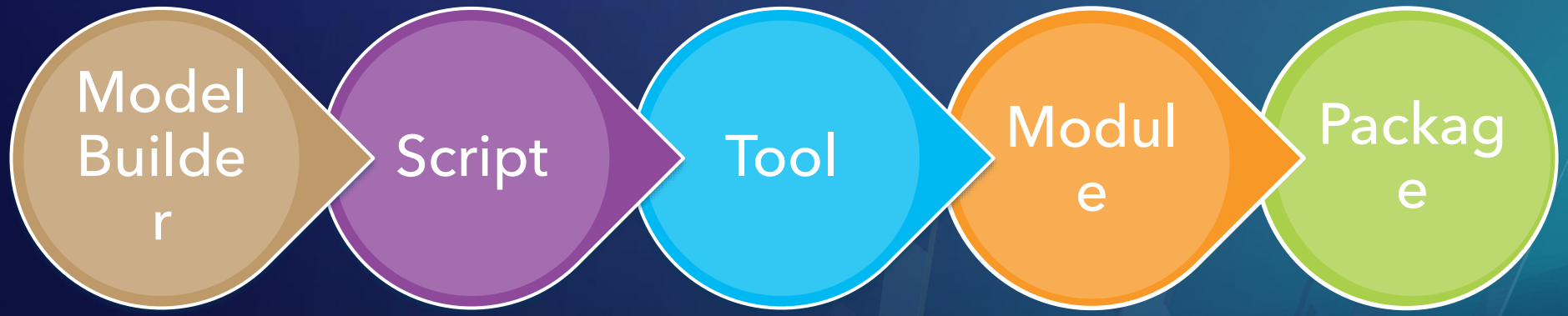
Integration Points for R + Python

Patterns and best practices

Conda for managing software

Conda

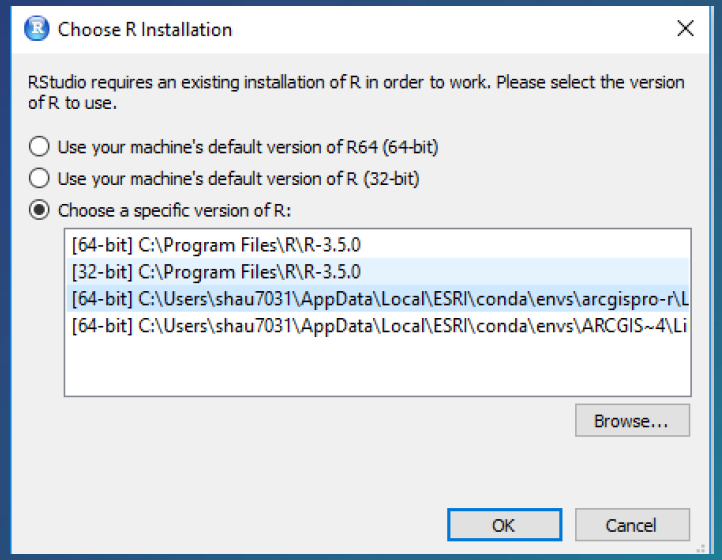
- If you don’t need it now, no problem – keep things simple
- But a good place to explore, and progress in your coding abilities. Helps future you, but also helps you share with others
- Mix together both R and Python dependencies into a consistent, reproducible environment



Best practices

Conda

- `arcgispro-r` environment
- Use editors that respect environments (Rstudio, Visual Studio Code, ...)
- Start now on building up reproducible habits



Broader Integration Points

R integration options

In order of level of effort:

- Jupyter Notebooks (`+leaflet.esri`)
- Microsoft GeoAI VM
- Geoprocessing Services
  - R runs on the server
  - Share real time outputs from R code
  - Build into live applications
  - Use as a component of a story map
- Conda packages