

BYOD Setup, 8:30 – 9:00am

1. Install software and check versions

- See email from **Amy Niessen** on **Thursday 3/2**: “2023 Esri Developer Summit PreSummit Workshop – Doing Spatial Data Science Using R and ArcGIS”
- ArcGIS Pro 3.1
- R 4.2.2
- RStudio 2022.12.0+353
- R-ArcGIS Bridge (arcgisbinding)
- In ArcGIS Pro: clone conda env and install r-arcgis-essentials

2. Download seminar data

- <https://github.com/EsriDevEvents/doing-spatial-data-science-using-r-and-arcgis-2023>



ESRI DEVELOPER SUMMIT 2023

Doing Spatial Data Science Using R and ArcGIS

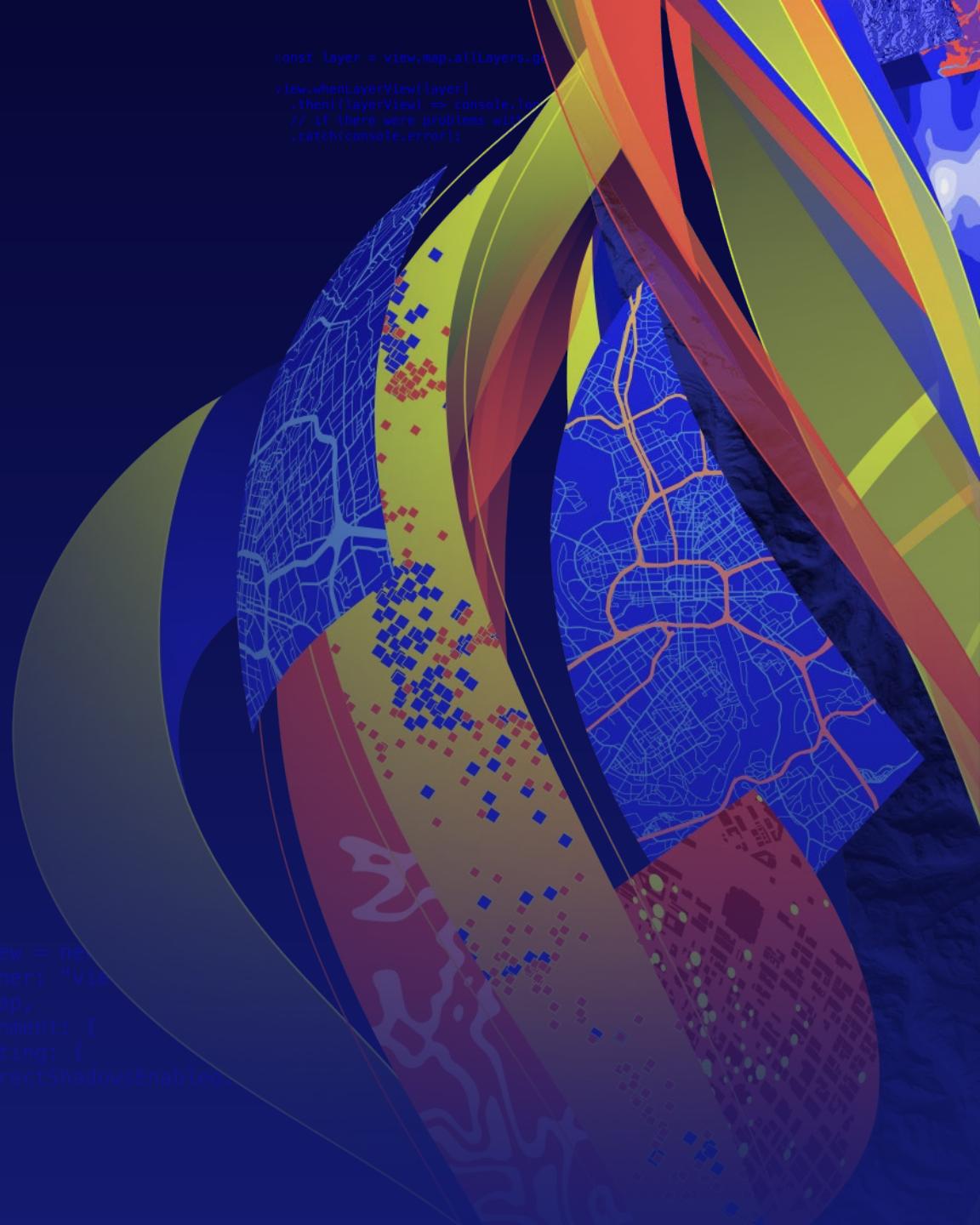
Nick Giner, Stella Li, Martha Bass

Seminar Overview

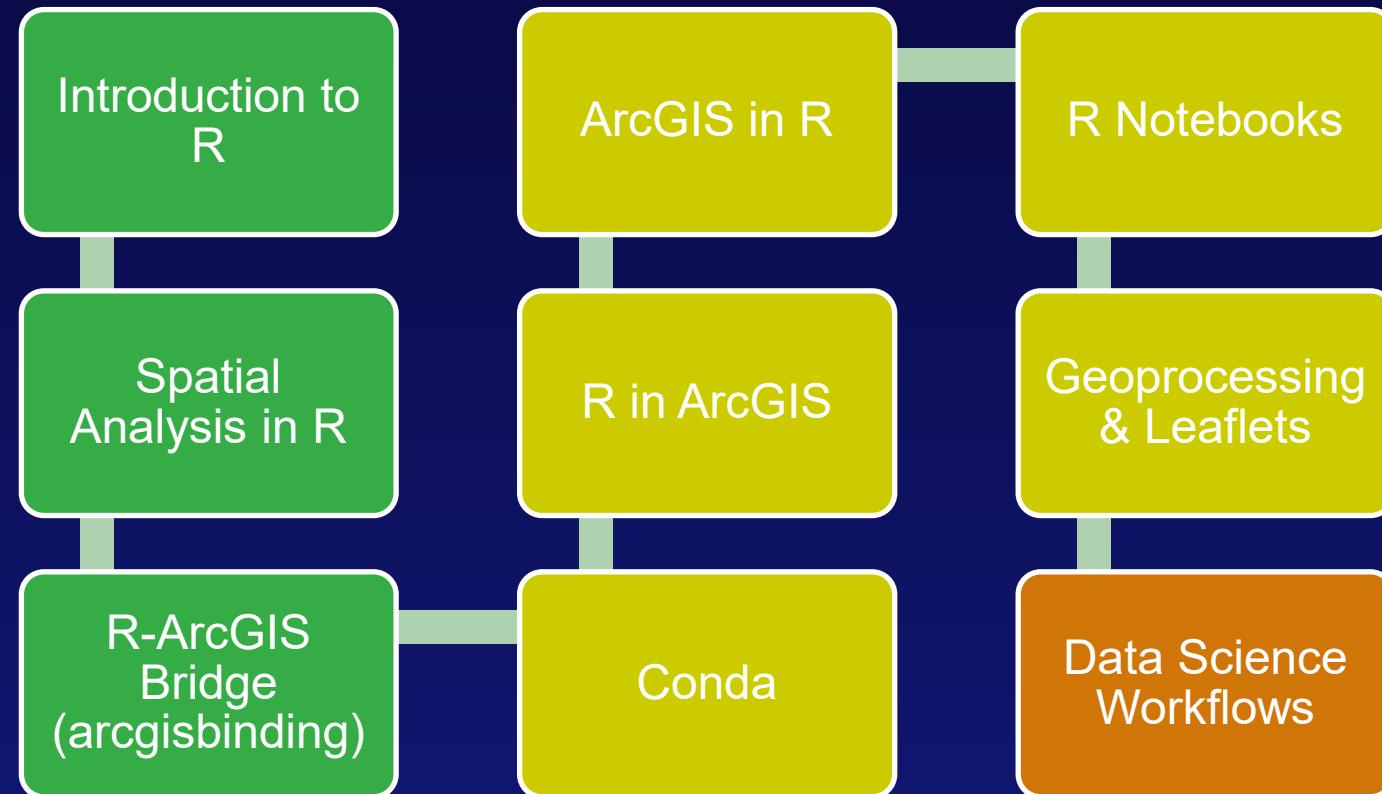
Today's Schedule

```
const layer = view.map.allLayers.get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(
    // if there were problems with
    .catch(console.error);
```

```
const view = new View({
  container: "view",
  map: map,
  environment: {
    lighting: {
      directShadowsEnabled: true
    }
  }
});
```



Course in a Nutshell



Workshop Schedule - Morning

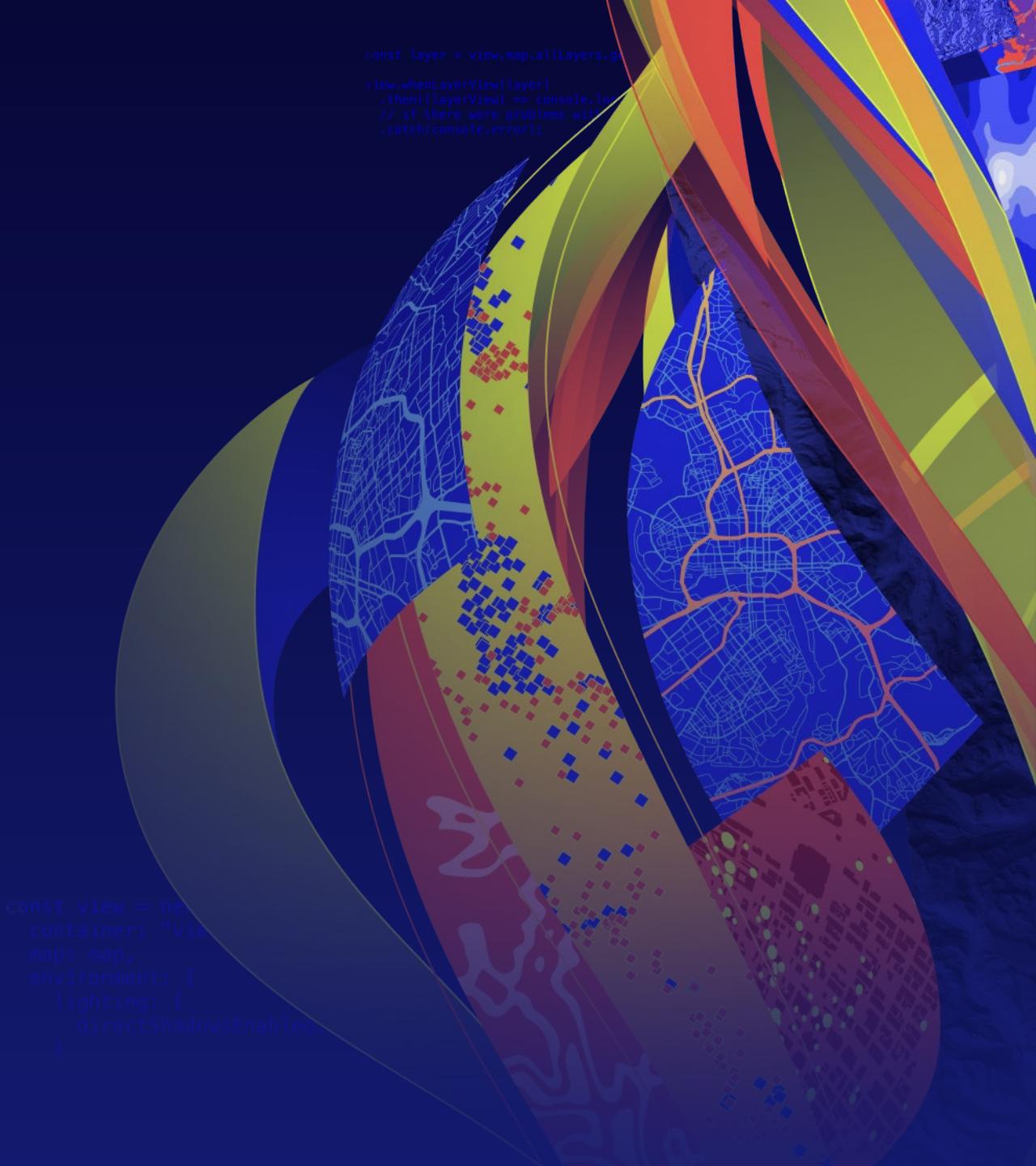
Section Title	Details	Time
1. Introduction to R Language	R Language R Dataframe R Studio	9:00 am – 9:30 am
2. Spatial Data Science in R	Spatial data representations Sf, sp, raster	9:30 am – 10:00 am
Break		10:00 am – 10:15 am
3. The R-ArcGIS Bridge	Setting Up Spatial data I/O Manipulating data	10:15 am – 11:15 am
4. R Script Tools	Hello World Wrapping R functionality	11:15 am – 12:00 pm
Lunch Break		12:00 pm – 1:00 pm

Workshop Schedule - Afternoon

Section Title	Details	Time
5. Getting Started with Conda	Setting Up r-arcgis-essentials	1:00 pm – 1:30pm
6. Getting Started with R Notebooks	Starting an R Notebook Markdown/Code	1:30 pm – 2:30 pm
Break		2:30 pm – 2:45 pm
7. Geoprocessing and interactive maps	reticulate esri.leaflet	2:45 pm – 3:45 pm
8. Data Science Workflows	Hands-On Problem Solving	3:45 pm – 4:30 pm
9. Q&A and Resources		4:30 pm – 5:00 pm

The R Language

An Overview



Why Use R?

- 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 19,261 available packages (doubled since 2016!)
- Versatile and powerful plotting

Language features: core types

- 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

- 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

R Resources

- 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/>

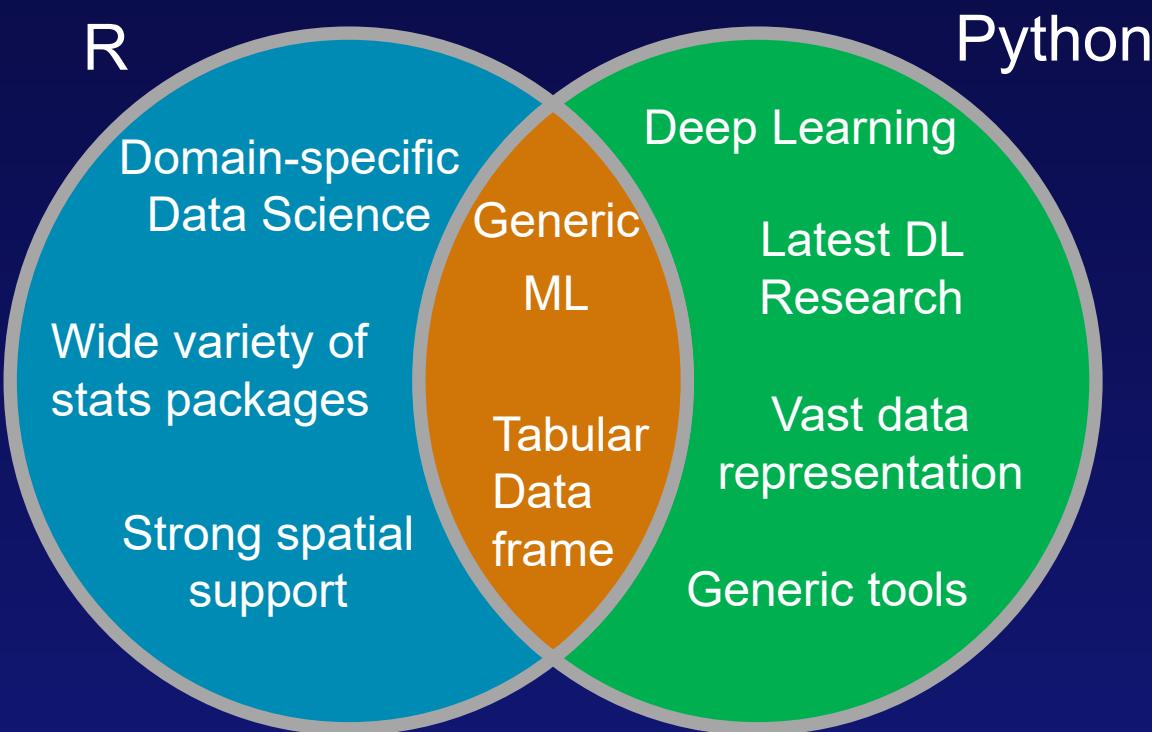
Strengths of the R Language

- Breadth of available geospatial analysis packages
 - 19,261 total R packages as of February 2023
 - <https://cran.r-project.org/web/views/Spatial.html>
- Discipline-specific data science functionality
 - Spatial data science
 - Biostatistics
- Ease of programming
- Ecosystem built for open-science

Weaknesses of R Language

- Performance
 - R is considerably slower than Python, about 5-10 times
 - Loops are notoriously slow
- Memory
 - Memory bottlenecks occurs very frequently with medium size (1, 2 GB) data
 - Inefficient R code is not as forgiving as Python
- Security (will not be an issue addressed in this course)

R and Python for Geocomputation



- R serves vast amounts of field-specific data science packages
 - Biostatistics
 - Geostatistics
 - Econometrics
- Python has general purpose data science libraries
 - Deep Learning (Tensorflow, ...)
 - Machine Learning (scikit-learn, ...)
 - Used for analysis & build scalable software

How to work with R?

RStudio

- IDE specifically designed for R
- Easiest way to get started
- Free
- Integrated Visualization and Variable Space
- Support for R Markdown and & notebooks

Jupyter

- Native support for R notebooks
- Interactive analysis
- Free
- Setup takes some effort

Other IDEs (Vim, VSCode,...)

- Editors that can be pointed to an R installation
- Vim and Visual Code are free
- Cumbersome to setup for non-developers
- Text editor with syntax marking

Installing Packages

- Easiest method to install packages
`install.packages("package_name")`
- For packages that do not exist in CRAN (main repository for R packages)
Download the R package repo
Extract to Folder containing packages
- Devtools
 - Tool used for building a package that require compilation

Defining R Variables

- `var <- my_fun(input1, input2)`
- Variable `var` contains the output from `my_fun`

Acceptable Variable Names	Unacceptable Variable Names
<code>var1</code>	<code>var\$1</code>
<code>var_1</code>	<code>var@1</code>
<code>var.1</code>	<code>1.Var</code>
	<code>var!1</code>

R Data Types

- Character – Stores str
- Complex – Stores complex numbers. $5 + 4i$
- Integer – Stores integers, numbers without floating points. 2, 5, 100
- Logical – Stores Boolean variables. *TRUE*, *FALSE*
- Numeric – Generic container for any number. 2, 3.14

R Data Structures

- Data Frame
- Factor
- List
- Matrix
- Vector (Atomic Vector)

More R Data Structures

Homogeneous (Single-Type) Data

- Can store one type of data
- Array(n-dimensional)
- Matrix (2D)
- Vector (1D)

Heterogeneous (Multi-Type) Data

- Can store multiple types of data
- List
- `r_list <- ('geo', 1.2, 'e', obj)`
- Data Frame

Date	Lat	Lon	Type
1/1/90	44.55	72.58	Built
1/1/10	44.55	72.58	Water

Summarizing Data Structures

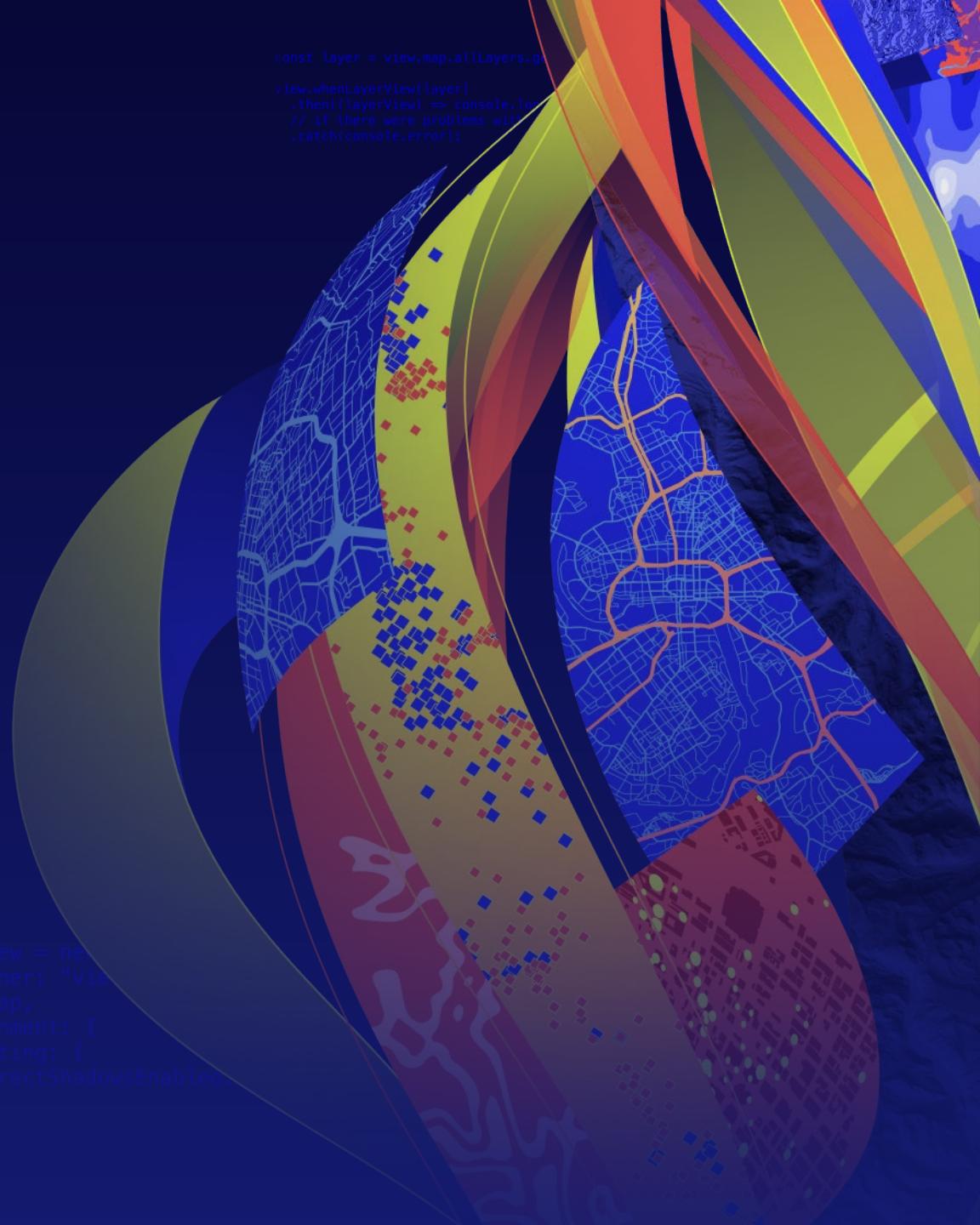
- Common functions to examine variable (var) or data frame (df)
- `typeof(var)`
- `length(var)`
- `class(var)`
- `dim(var)`
- `head(df)`

Getting Documentation Help

- I forgot what this function did
- I do not remember what kind of inputs I need
- Getting help
 - ? <name of the function>
 - ? arc.open

Spatial Data in R

Packages and the Ecosystem



```
const layer = view.map.allLayers.get(0);  
view.whenLayerView(layer)  
.then(layerView => console.log(  
// if there were problems with  
.catch(console.error);
```

```
const view = new View({  
  container: "view",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
});
```

Spatial R Data Types

- Vector Data
 - sf, sp packages
 - Represent spatial information efficiently
 - Accepted by most packages that require spatial dataframes
- Raster Data
 - raster package

Spatial Vector Data Representation in R

- Same pattern: Geometry Info + Attributes
- sf and sp only differ in representing geometry information
- Once vector data is in sf and sp format
 - Spatial operators: join, dissolve, merge, etc.
 - Reprojecting the data (change projection system)
 - Spatially subset the data

Spatial Vectors in R - sf

- The newer spatial vector representation in R
- Uses the simple features (sf) standard of OGC (Open Geospatial Consortium)
- Simple Features are:
 1. Point
 2. Polygon
 3. Linestring
 4. Multipoint
 5. Multipolygon
 6. Multilinestring
 7. GeometryCollection

Raster Data Representation in R

- Raster Info + Attributes
- Raster Info
 - nrows : Number of rows
 - ncols: Number of columns
 - nbands: Number of bands
 - extent
 - projection

Connecting ArcGIS to R

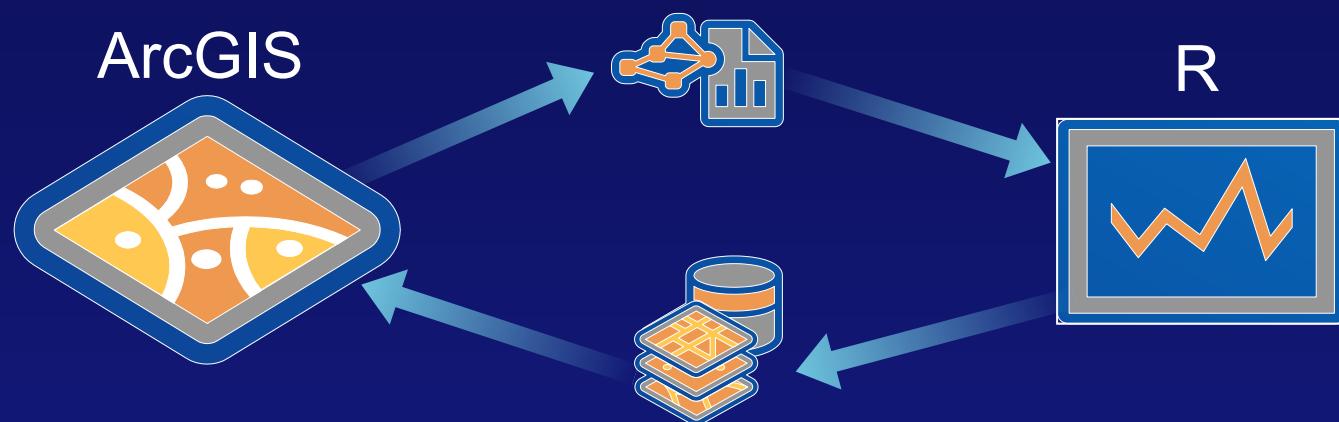
The R-ArcGIS Bridge

```
const view = new View({  
  container: "viewContainer",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
});
```

```
const layer = view.map.allLayers.get(0);  
  
view.whenLayerView(layer)  
.then(layerView => console.log(  
  // if there were problems with  
  // the layer, they would appear here  
  layerView.error));
```

R-ArcGIS Bridge

- Store your data in ArcGIS
- Directly access it in R
- Returns R objects back to ArcGIS native data types
- sf, sp, and raster conversion



R ArcGIS Bridge – ‘arcgisbinding’ package

- Spatial data support
 - Vector Data (sf, sp)
 - Raster Data (raster)
- Read/Write functionality
- Converting Functionality
 - Shapefiles, File geodatabase, feature service > sf, sp
 - Raster, Mosaic datasets > raster
- Wrapping R Tool functionality

Different Ways of Setting up the R-ArcGIS Bridge

Geoprocessing Tab

- Fastest way to get setup
- You need to have R installed
- Useful if you have R on your machine

Conda Package Manager

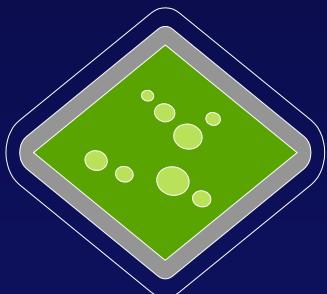
- Easy and streamlined process
- You do NOT need to have R installed
- Independent working environments

GitHub Repository

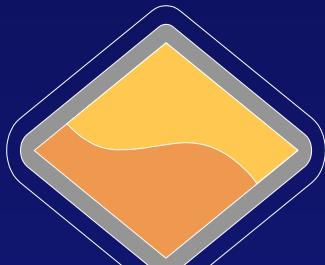
- Least streamlined
- Necessary for ArcGIS Desktop
- Try if everything else fails for ArcGIS Pro

arcgisbinding functions for Vector Data

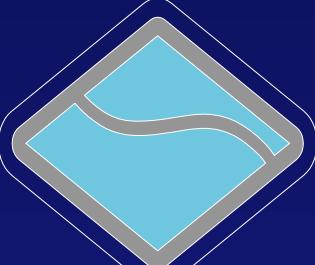
Points



Polygons



Lines



`arc.open`

Metadata	
Location	C:\my.gdb\fs
Type	FeatureClass
extent	...
sr	...

Arc Data frame	
Geometry	Data
POINT	25
POINT	3
POINT	4

arc.open

- Function signature: ***arc.open***(path)
- **Input:** path, full path to data
 - Feature Class (ESRI)
 - Shapefile
 - Feature Service
 - OGC compliant data types
- **Output:** Metadata for data in path

arc.open – data path

Feature service:

A screenshot of an ArcGIS feature service page for 'Coronavirus COVID-19 Cases V2 / Cases_state'. The page includes sections for 'Point layer' (CSSE_covid19), 'Summary', 'Description', and 'Credits (Attribution)'. A 'URL' section at the bottom contains the link <https://services1.arcgis.com/0MSEUqKaxl>. A green arrow points from this URL section to the 'Copy Path' option in the catalog menu.

URL in the middle of the page

`arc.open(copied.data.path)`

Local data:

A screenshot of an ArcGIS Catalog context menu. The 'Copy Path' option is highlighted with a green box and a green arrow pointing to it. The menu also includes other options like 'Cut', 'Copy', 'Delete', 'Rename', 'View Metadata', 'Edit Metadata', and 'Properties'.

Catalog – right click on the data
– Copy Path

arc.select

- Brings in data associated with metadata defined by arc.open
- ***arc.select***(object, fields, where_clause, sr)
- **Input:**
 - object: Output from arc.open. Metadata
 - fields: List of fields to bring in
 - where_clause: SQL expression to bring in a subset of data
 - sr: Spatial reference for reprojecting
- **Output:**
 - Arc format spatial dataframe

arc.data2sf & arc.data2sp

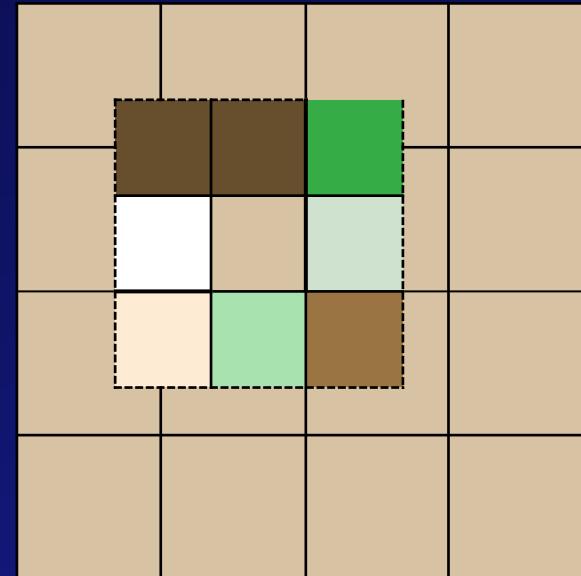
- Converter functions
- Converts arc-type spatial R data frames into sf and sp type spatial dataframes
 - `arc.data2sf(data_arc)`
 - `arc.data2sp(data_arc)`
- **Input:** `data_arc` : arc type spatial R data frame
- **Output:** sp or sf type spatial R data frame

arc.write

- Writes a spatial R data frame into disk
- Input data frame can be a:
 - sf object
 - sp object
 - arc object
- Output for vector data can be:
 - ESRI Feature Class
 - Shapefile

Working with Raster data

- Be mindful that rasters are gridded representations of a variable of interest
 - Elevation
 - Land use
 - Temperature
 -
- When used in analysis, you need to pay attention to
 - Cell size
 - Resampling method applicable to the data stored in the raster
 - Projection
 - If using multiple rasters, these must be aligned



Use `arc.open` too for Raster or Image service

- `arc.open` brings only the metadata of the input, so you can use this for raster datasets too

```
# Read image service from Living Atlas
raster.url <- ''
raster.url <- 'https://env1.arcgis.com/arcgis/rest/services/Sentinel2_10m_LandCover/ImageServer'
raster.open <- arc.open(raster.url)

> raster.open
dataset_type      : RasterDataset
path              : https://env1.arcgis.com/arcgis/rest/services/Sentinel2_10m_LandCover/ImageServer
format            : Image Service
pixel_type        : U8 (8bit)
compression_type: NA
nrow              : 1826152
ncol              : 4007502
extent            : xmin=-180, ymin=-80.01342, xmax=180, ymax=84.03261
WKT               : GEOGCS["GCS_WGS_1984",DATUM["D_WGS_1984",SPHEROID["WGS_1984"...
WKID              : 4326
bands             : 1
      ncol   nrow nodata min  max      mean    stddev
Band_1 4007502 1826152      0   1  11 6.858921 3.908966
```

arc.raster

- Similar to arc.select, brings the data associated to metadata open with arc.open into R
- ***arc.raster***(object, bands, nrow, ncol, extent, pixel_type, resample_type, sr)
- **Input:**
 - object: Output from arc.open. Metadata
 - bands: List of bands to read (default: all bands).
 - nrow: Number of rows for the raster or mosaic dataset. (default: object@nrow)
 - ncol: Number of columns for the raster or mosaic dataset. (default: object@ncol)
 - extent: List (xmin,ymin,xmax,ymax) coordinates defining the extent of the raster to be read.
 - pixel_type: pixel type for the information stored in each raster cell.
 - resample_type: type of resampling to perform on the raster data.
 - sr: Spatial reference for reprojecting (number, not string)
- **Output:**
 - Arc format raster

Working with Private Remote Data- As Easy as 1, 2, 3

`arc.check_portal` and `arc.portal_connect`

`arc.check_portal()` lists portals available to you

- Portals added to the ArcGIS Pro instance

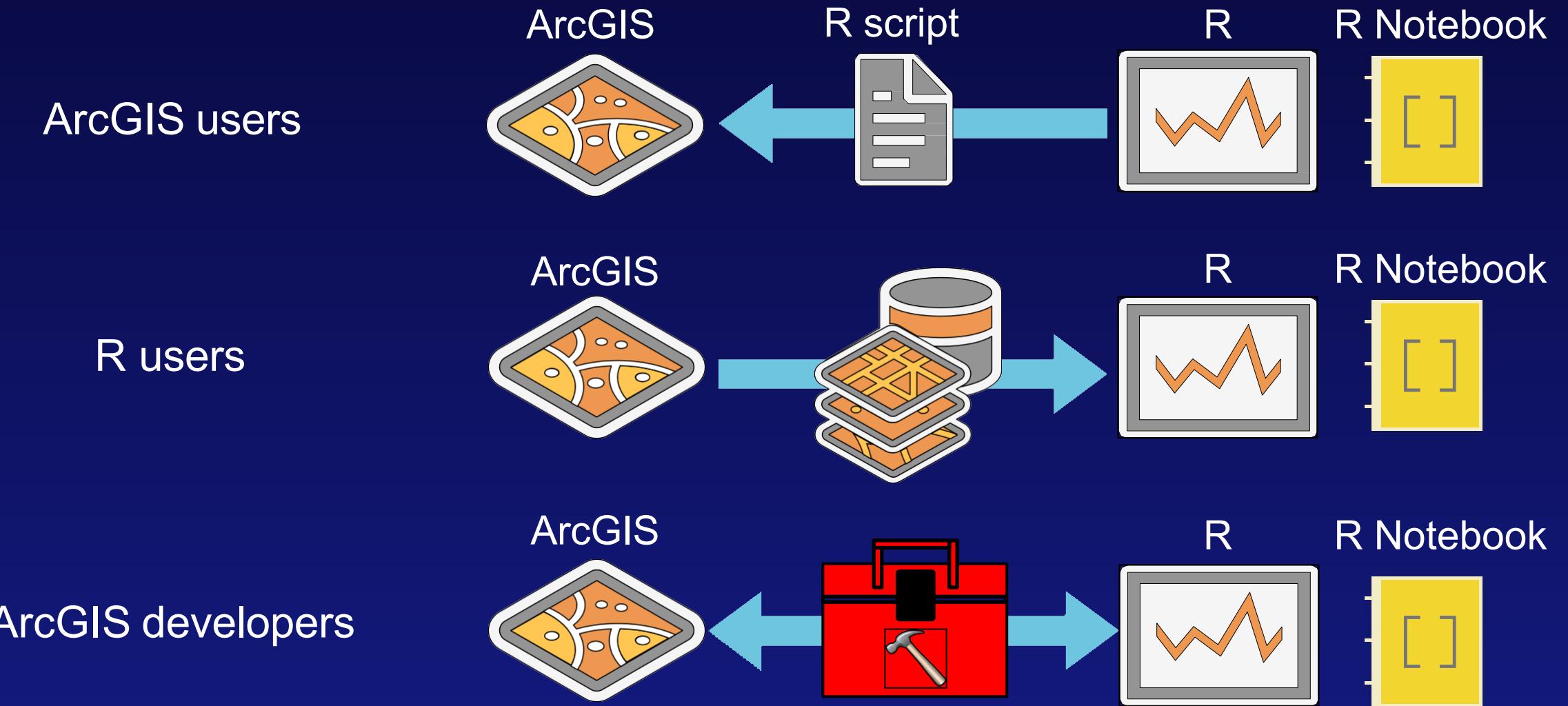
1. Login to your portal

`arc.portal_connect(url, user, password)`

2. Use `arc.open` to bring remote data in

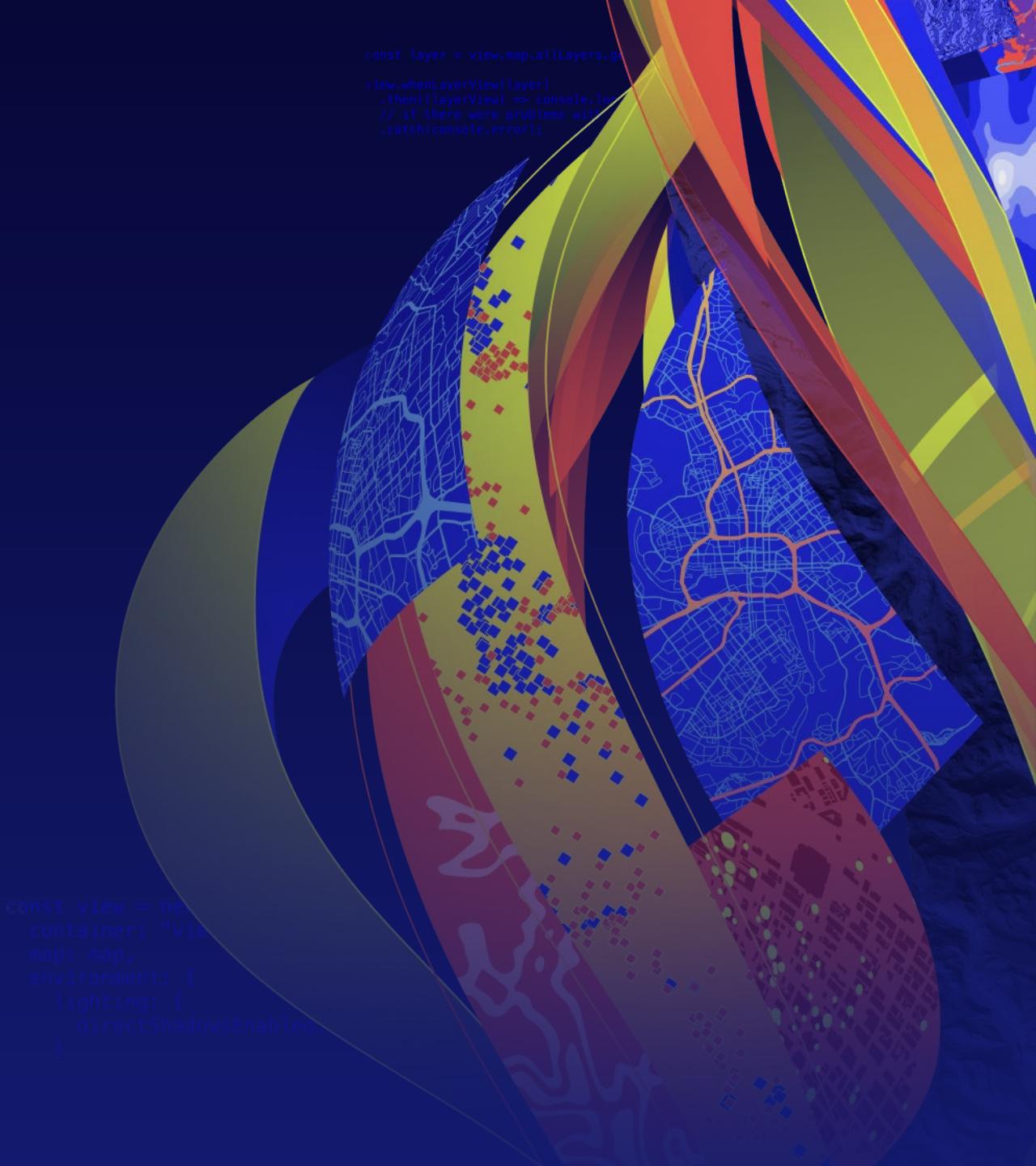
`gis_data <- arc.open(path = url)`

Different Ways to Use R-ArcGIS Bridge



R in ArcGIS

R Script Tools



```
const layer = view.map.allLayers.get(0);
view.whenLayerView(layer)
  .then(layerView => console.log("Layer loaded"))
  // if there were problems with the layer
  .catch(console.error);

const view = new View({
  container: "viewDiv",
  map: map,
  environment: {
    lighting: {
      directShadowsEnabled: true
    }
  }
});
```

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]]

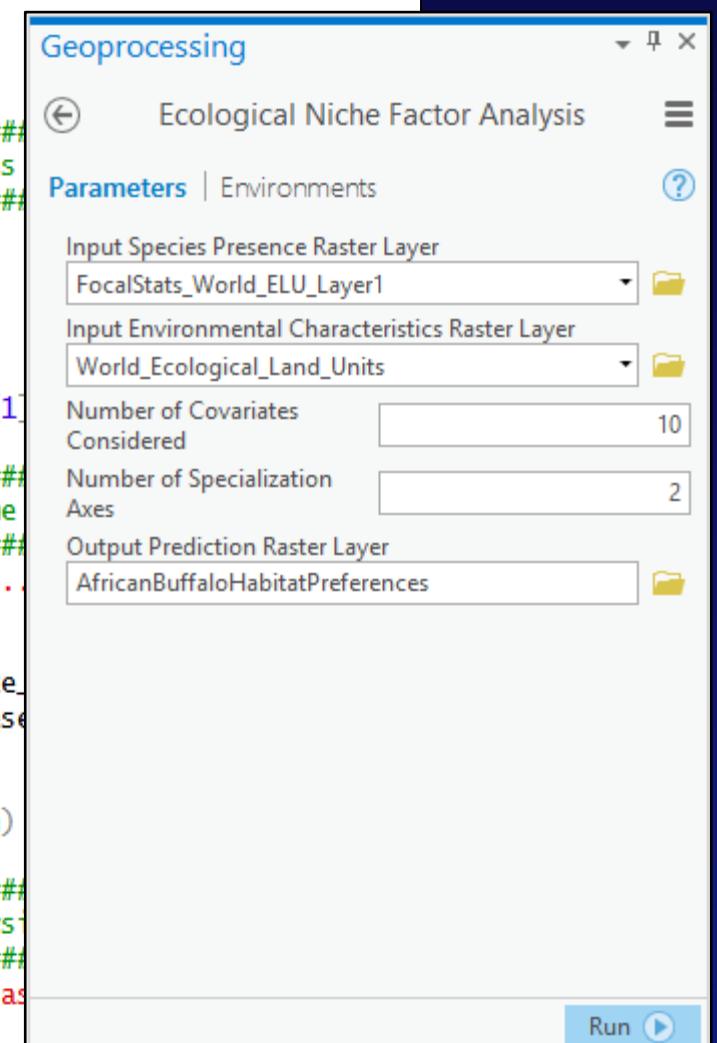
prediction_output <- out_params[[1]]

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

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

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

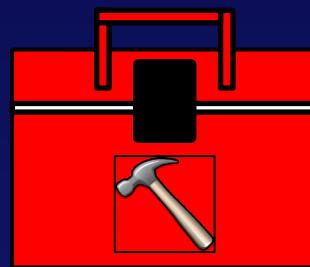
#####
### Ecological Niche Factor Analysis
#####
arc.progress_label("Formating data...")
arc.progress_pos(60)
```



Create Script Tools in ArcGIS

R Script Tools

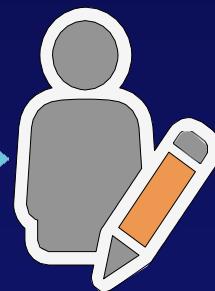
1) Create toolbox



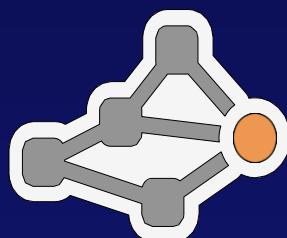
2) Add R script



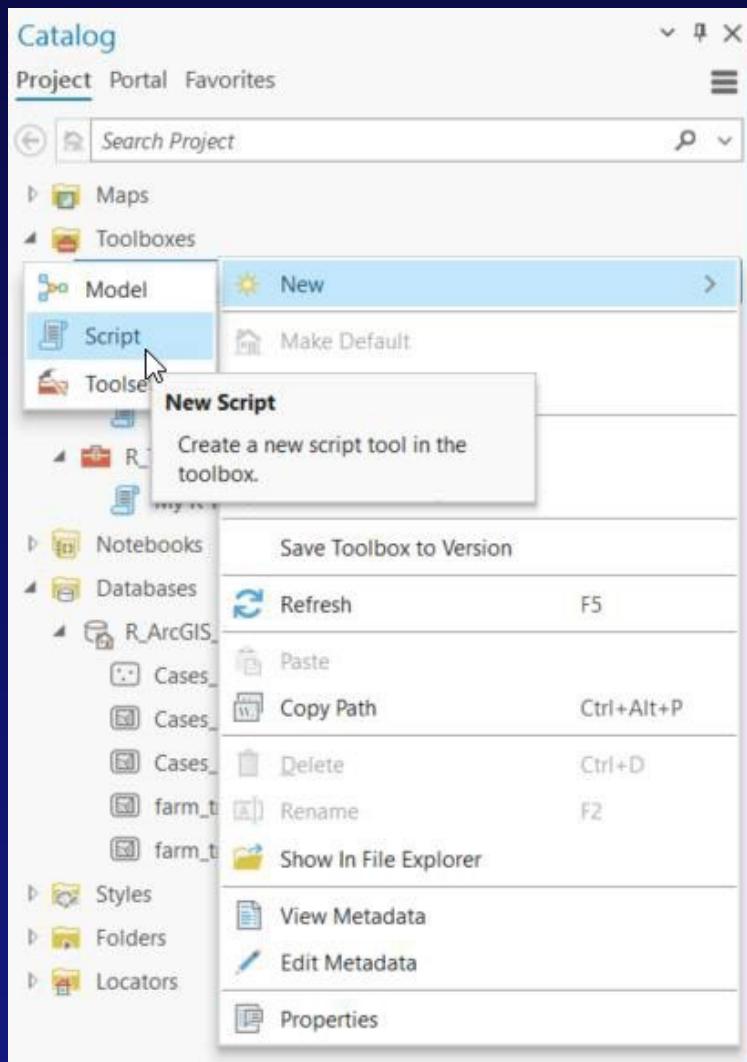
3) Set parameters



4) Perform analysis



Create a script tool



- Catalog – Toolboxes – Project Toolbox
 - Right click and create a new script file in the project toolbox
- Name the tool
 - Label will be the name shown in the toolbox
- Define the parameters
 - Direction: Input/output
 - Data Type
 - Type: required/optional
 - Dependency
- Connect to the r script
 - Execution – Script File
- Click 'OK' to finish editing

Create Script Tools in R

R Script Tools

Generic, reusable script template

Wrapping functionality

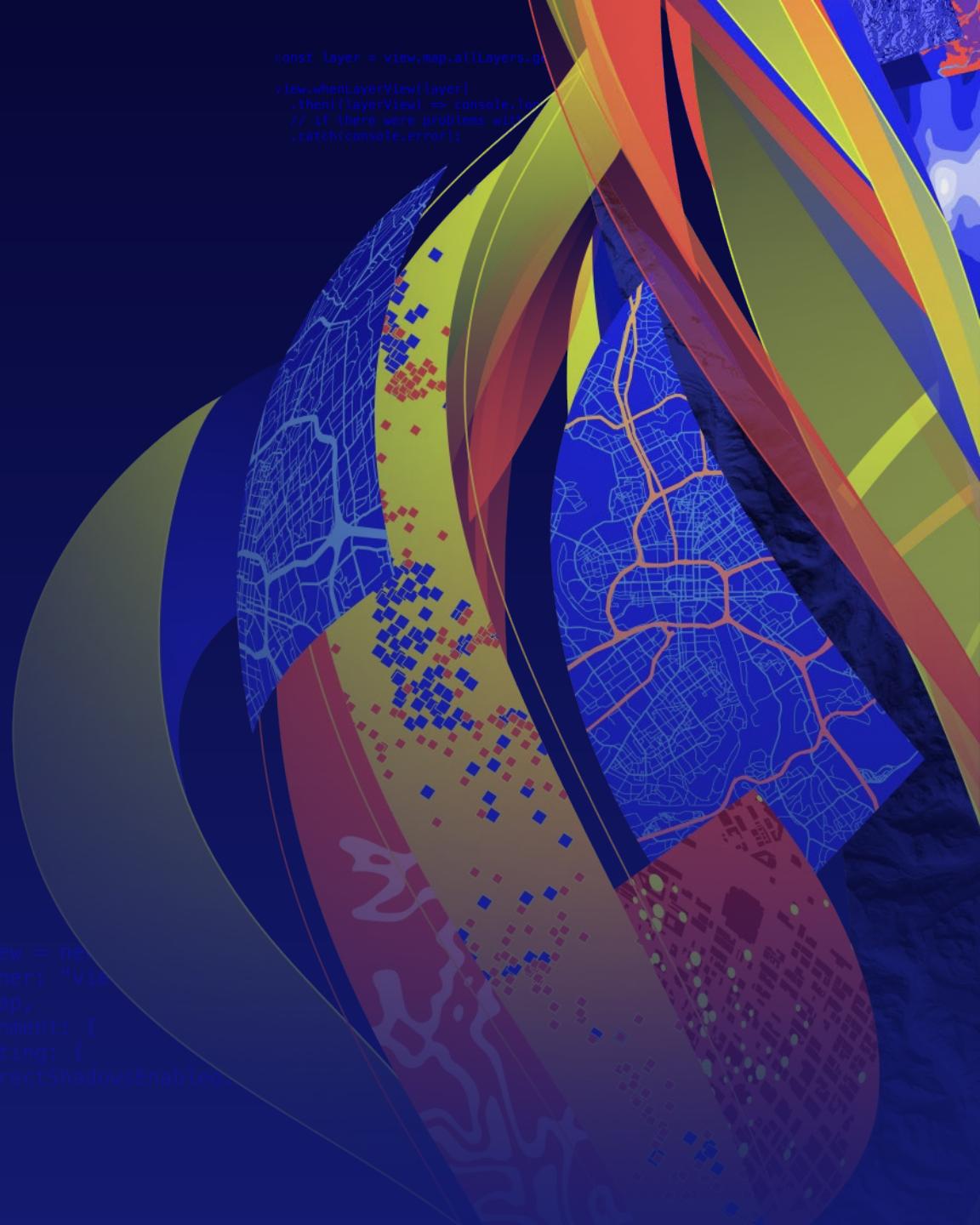
```
tool_exec <- function(in_params, out_params)
{
}
```

Defining input and output parameters

```
input_fc <- in_params [[1]]
input_par2 <- in_params [[2]]

output_fc <- out_params [[1]]
```

Conda Integration & R Notebooks



Enablers of R Notebooks with ArcGIS Pro

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

Setting Up R Notebooks with Conda

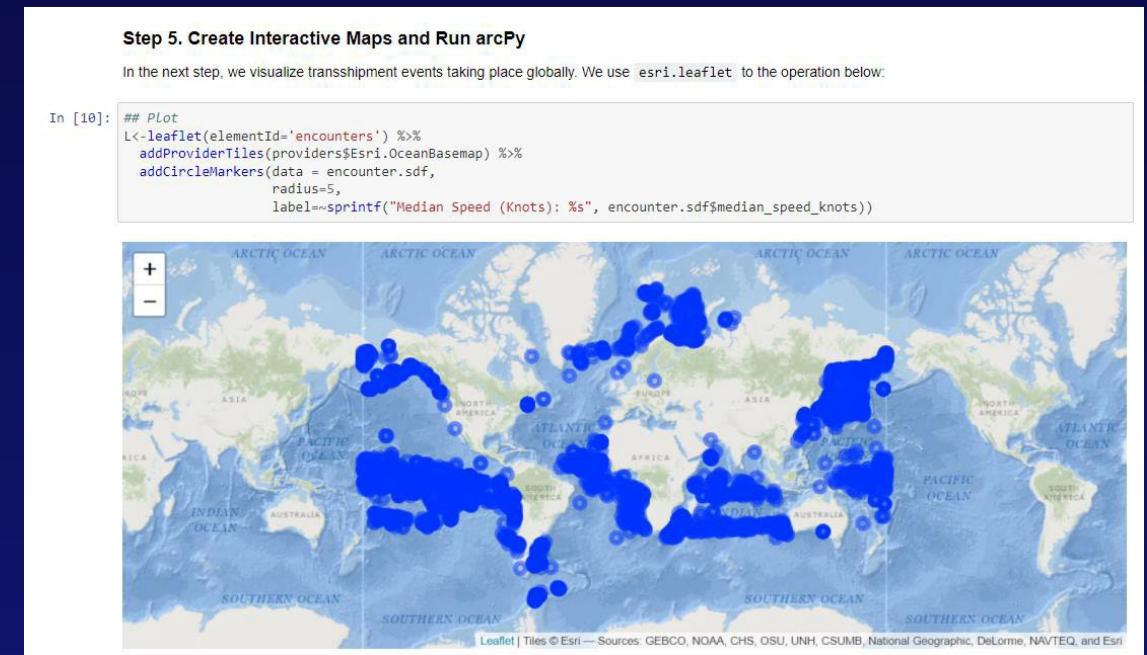
- Handle Python, R and many other languages
- Use cutting edge software from diverse fields (stats, AI, ML + many others)
- Works with R, Microsoft R, R Studio and complex hybrid stacks
- Conda command line, GUI available in Pro



Terraria, isolated environments
Image: Susanna Marsiglia on Unsplash

Notebooks and Interactive Computing

- Intertwine 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



r-arcgis-essentials

Conda Package for R Notebooks

- “Package of packages”
- Clone default ArcGIS Pro environment
 - Includes everything that ArcGIS Pro ships with
- Install r-arcgis-essentials
 - R Notebooks (IR-Kernel)
 - sf package
 - sp package
 - raster package

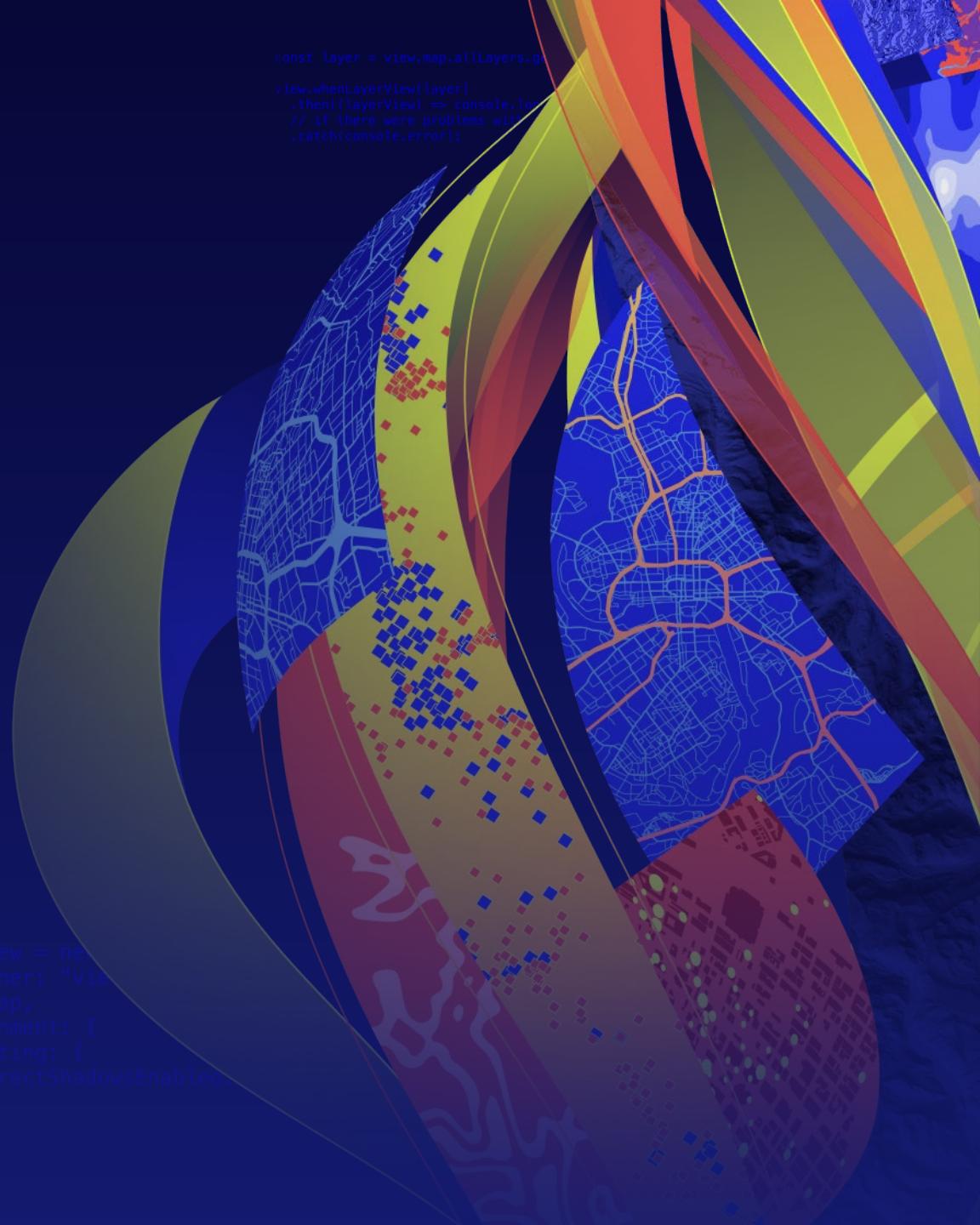
r-arcgis-essentials

- r-mro
- arcgisbinding
- ir-kernel
- sf
- sp
- raster

arcgispro-py3

- arcpy
- arcgis
- numpy
- pandas
- ...

Reticulate and ArcPy



```
const layer = view.map.allLayers.get(0);  
view.whenLayerView(layer)  
.then(layerView => console.log(  
// if there were problems with  
.catch(console.error);
```

```
const view = new View({  
  container: "viewDiv",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
});
```

What is reticulate?

- R package that enables using Python libraries from R
- Enables creating R-centric workflows that use Python libraries
- Ability to call ArcPy functions from R
 - Check out your history items

How do you use it with ArcPy

- Specify the Python to be used

```
reticulate::use_python('path/to/python.exe', required = T)
```

- You can import ArcPy in R workflows

```
ARCPY <- import("arcpy")
```

- Set Environment Variables

```
ARCPY$env$overwriteOutput = TRUE
```

- Check Out Licenses

```
ARCPY$CheckOutExtension("GeoStats")
```

- Run Geoprocessing Tools

```
ARCPY$ga$EmpiricalBayesianKriging("EMU_Global_90m", ...)
```

```
inst view = new SceneView({  
  container: "viewDiv",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
})
```

R-ArcGIS Bridge - Modeling the Impact of Transshipment on Marine Biodiversity
In this notebook, working with arcgisbinding to understand the impact of transshipment on biodiversity will be presented.

Transshipment = Transfer of goods and crew between ships
Reefer = Refrigerated ship that can take the catch from fishing ships
Working Ship = Fishing vessel
Loitering Events: Transshipment-capable vessels (reefers) going slow enough and long enough to transship
Encountering Events: Potential encounter events between fishing vessel and reefer



Importing Necessary R libraries

```
In [1]: library(arcgisbinding, quietly = TRUE)  
*** Please call arc.check_product() to define a desktop license.  
  
In [14]: library(dismo, quietly = TRUE)
```

Reticulate Demo

```
inst view = new SceneView({  
  container: "viewDiv",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
})
```

R-ArcGIS Bridge - Modeling the Impact of Transshipment on Marine Biodiversity

In this notebook, working with arcgisbinding to understand the impact of transshipment on biodiversity will be presented.

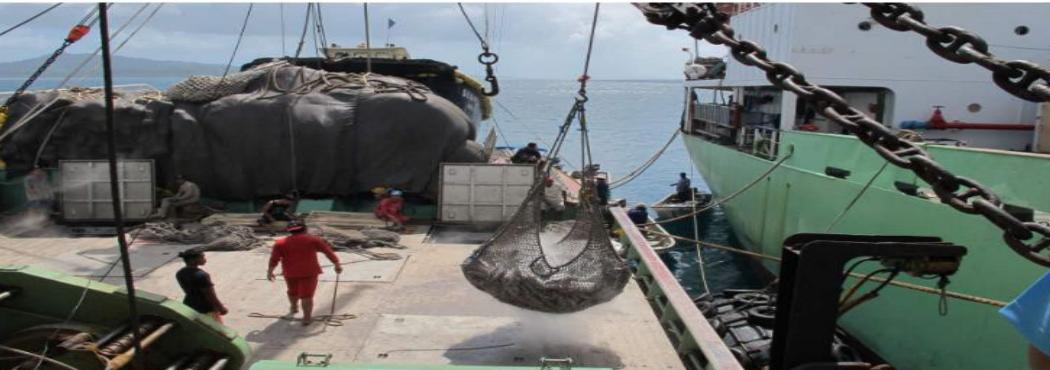
Transshipment = Transfer of goods and crew between ships

Reefer = Refrigerated ship that can take the catch from fishing ships

Working Ship = Fishing vessel

Loitering Events: Transshipment-capable vessels (reefers) going slow enough and long enough to transship

Encountering Events: Potential encounter events between fishing vessel and reefer



Importing Necessary R libraries

```
In [1]: library(arcgisbinding, quietly = TRUE)  
*** Please call arc.check_product() to define a desktop license.  
  
In [14]: library(dismo, quietly = TRUE)
```

R Notebooks Demo

Q&A

R-ArcGIS Bridge Team

```
const layer = view.map.allLayers.get(0);
view.whenLayerView(layer)
  .then(layerView => console.log(
    // if there were problems with
    .catch(console.error);
```

```
const view = new View({
  container: "viewDiv",
  map: map,
  environment: {
    lighting: {
      directShadowsEnabled: true
    }
  }
});
```

R-ArcGIS Bridge at DevSummit 2023

- Find us at the Spatial Analysis and Data Science (SADS) area
 - Dedicated booth for R-ArcGIS Bridge
 - Ask questions
 - Get help on your code/workflow
- Technical Session: “Harnessing the Power of R in ArcGIS with R-ArcGIS Bridge”
 - Wednesday (3/8) 5:30 pm – 6:30 pm
 - San Jacinto, Renaissance Hotel

R-ArcGIS Bridge Resources

Product Landing Page

<https://www.esri.com/en-us/arcgis/products/r-arcgis-bridge/overview>



Learn Path

<https://learn.arcgis.com/en/patterns/get-started-with-r-and-arcgis/>

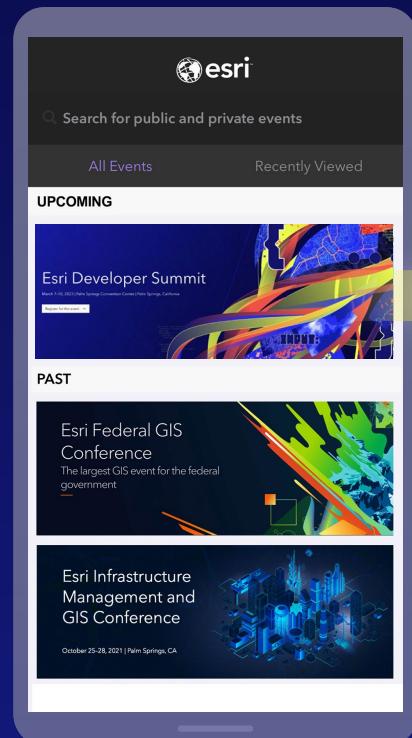


More questions on R bridge? Check the community's GitHub repo:

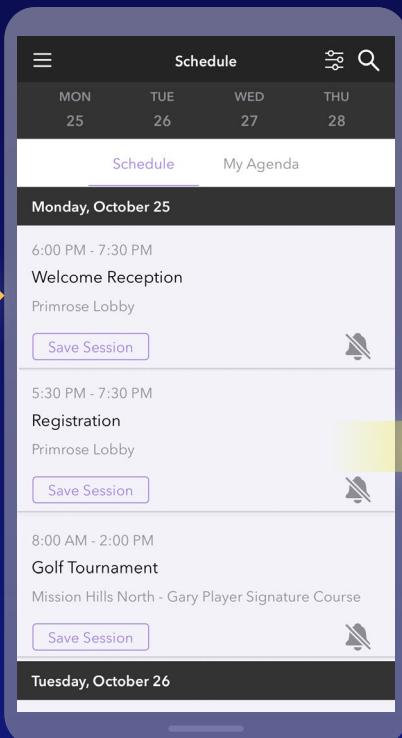
<https://github.com/R-ArcGIS/r-bridge>

Please Share Your Feedback in the App

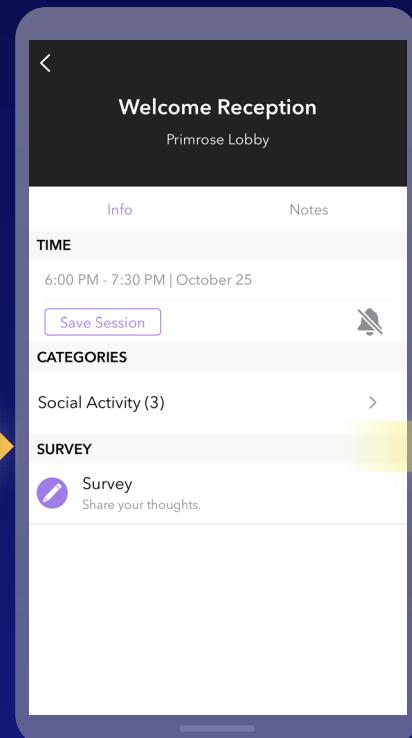
Download the Esri Events app and find your event



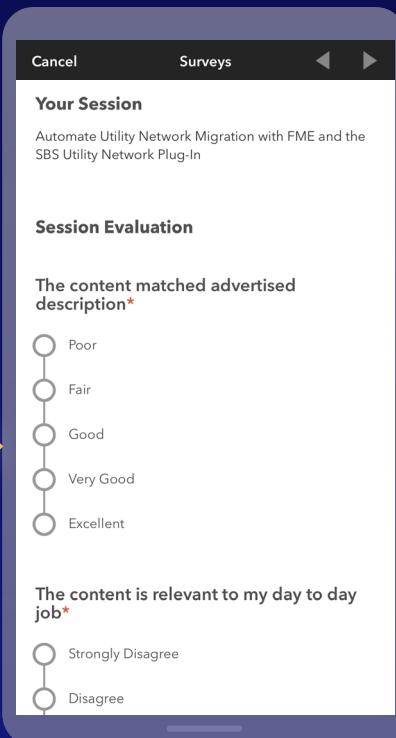
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



Connect With Us On Social

And Join the Conversation Using #EsriDevSummit



twitter.com/EsriDevs #esridevsummit, #esridevsummit23



twitter.com/EsriDevEvents



youtube.com/c/EsriDevelopers



links.esri.com/DevVideos



github.com/Esri



github.com/EsriDevSummit



links.esri.com/EsriDevCommunity

```
const layer = view.map.allLayers.get(0);  
view.whenLayerView(layer)  
.then(layerView => console.log(  
// if there were problems with  
.catch(console.error);
```

```
const view = new MapView({  
  container: "viewDiv",  
  map: map,  
  environment: {  
    lighting: {  
      directShadowsEnabled: true  
    }  
  }  
});
```



esri®

THE
SCIENCE
OF
WHERE®

Copyright © 2023 Esri. All rights reserved.

```
const layer = view.map.allLay...
view.whenLayerView(layer)
.then((layerView) => cons...
// if there were problems
.catch(console.error);
```

</SCRIPT>

```
const view = new SceneView()
container: "viewDiv",
map: map,
environment: {
  lighting: {
    directShad...
```

LIVE
BY
THE
CODE