esri Canada

2017 Education and Research Conference

# Working with the R-ArcGIS Bridge

Michael G. Leahy PhD

# Workshop Agenda

- Introduction / Overview
- Basic use of R and common R packages
- Use of the R-ArcGIS package in R scripts
- Building geoprocessing script tools using R-ArcGIS
- More Samples/Demos (as time permits…)

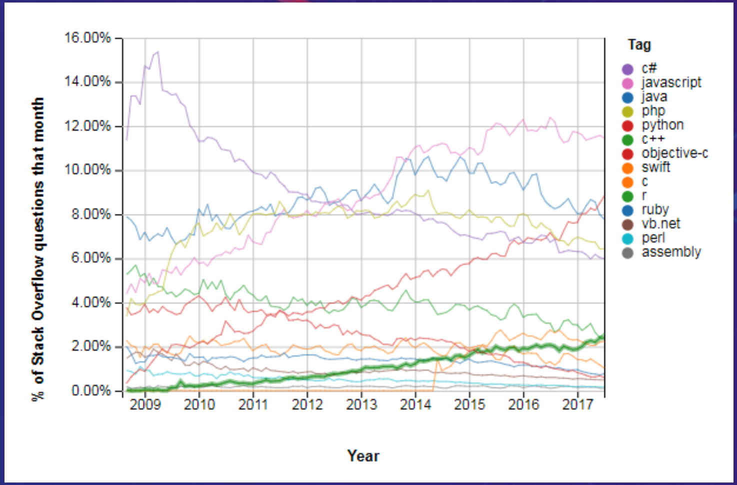
# Introduction to R

# About R

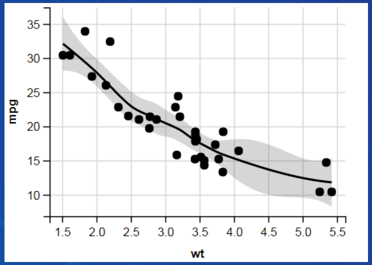
- A system for statistical computation and graphics
  - Open source implementation of S programming language developed at Bell Laboratories
  - Supported by an active user community developing and maintaining a variety of packages
  - Currently over 11,000 packages available from the Comprehensive R Archive Network (CRAN)

# About R

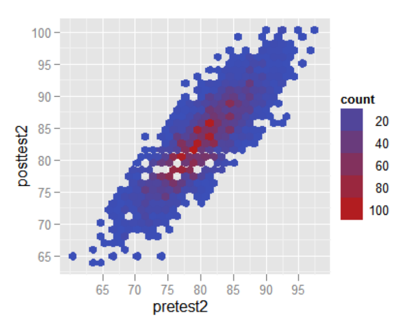
- Why use R?
  - De facto standard language for statisticians - most state-of-the-art statistical methods are available as packages in CRAN
  - Among the fastest growing programming languages
  - Powerful language and packages for generating high-quality plots and graphics (base R plots, ggplot2, ggvis, etc.)



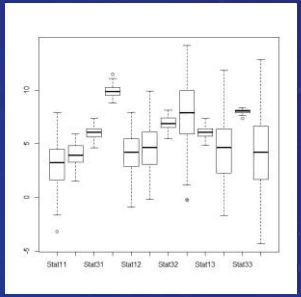
Source: StackOverflow



ggvis.rstudio.com



r4stats.com



www.r-bloggers.com

# R Development Environments

- R-Gui
  - Basic UI installed by default with R

- RStudio - https://www.rstudio.com/
  - Open source and enterprise-ready professional IDE for R
  - Maintained by a team of professional software engineers
  - Syntax highlighting/formatting, code completion
  - Integrated debugging, documentation, plot display
  - Direct execution of code from source editor
  - Organize work in RStudio projects

- R Tools for Visual Studio - https://www.visualstudio.com/vs/rtvs/
  - Full-featured IDE for R integrated within Visual Studio

# R – Third-party Packages

- Hadley Wickham's 'tidyverse' set of packages:
  - `ggplot2, dplyr, tidyr, ...`
- Statistical modelling:
  - `car, lme4, caret, mgcv`
- Spatial data / analysis:
  - `arcgisbinding, sp, maptools, spdplyr / mclust, spgwr, spatstat, lctools, deldir ...`
- CRAN Task Views for categorized view of packages (e.g., 'Spatial')
  - https://cran.r-project.org/web/views/

# R Basics

- Basic Object Types

  **`Vector, Lists, Matrices, Arrays, Factors, Data Frames, Time Series`**

- Basic Data Types:

  **`Logical, Numeric, Integer, Character, Complex, Raw`**

- Variable Assignment:

  ```
  x <- 1      # Creates a numeric vector with one value ( = works too… *)
  ```

- Vectors with multiple values:

  ```
  x <- c('apples', "bananas", 'oranges')    # Character vector w/ 3 values
  ```

- Install/load packages:

  ```
  packages.install('sp')        # Install CRAN packages (includes dependencies)
  library(spgwr)                # Load a package in current R workspace
  ```

# R – the **dplyr** package

- Part of Hadley Wickham's 'tidyverse' – a fast and consistent tool for working with tabular data

- Adds a variety of essential methods for data frame manipulation

  **filter, arrange, select, mutate, summarize, ...**

- Works with the forward-pipe operator (from the **magrittr** package), enabling more efficient and concise workflows in R:

```
arranged_data <- survey_data %>% filter(age > 15) %>%
  select(education, income, age) %>% arrange(income, age)
```

*Instead of:*

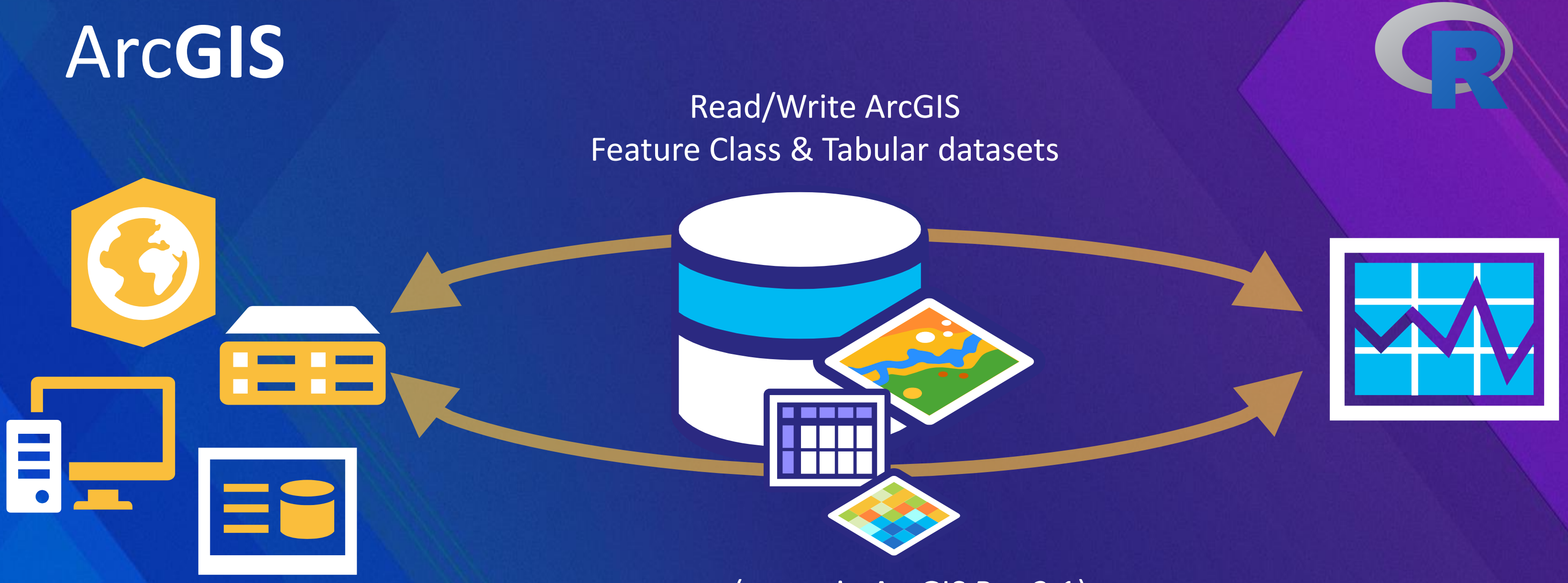
```
filtered_data <- filter(survey_data, age > 15)
selected_data <- select(filtered_data, education, income, age)
arranged_data <- arrange(arranged_data, income, age)
```

esri Canada

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

# R-ArcGIS Bridge

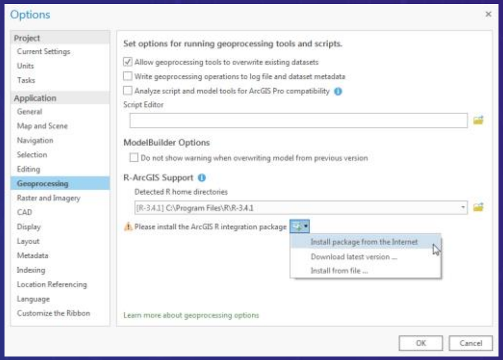
- Enables seamless access to read/write ArcGIS datasets in R
  - Shape Files (read/write)
  - Geodatabase Feature Classes & Tables (read/write)
  - FeatureService layers (read – with ArcGIS Pro)



(raster in ArcGIS Pro 2.1)

# R-ArcGIS Bridge Setup

- Software Required
  - R for windows version 3.2.0 or later (R 3.4.1+ recommended)
  - ArcGIS Pro 1.1+ or ArcGIS 10.3.1+ (ArcGIS Pro 2.0.1+ recommended)
- In ArcGIS Pro:
  - Project -> Options -> Geoprocessing
    - Install from Internet, or
    - Install from file (e.g., **arcgisbinding_1.0.0.128.zip** downloaded previously from GitHub)



Set options for running geoprocessing tools and scripts.

☑ Allow geoprocessing tools to overwrite existing datasets
☐ Write geoprocessing operations to log file and dataset metadata
☐ Analyze script and model tools for ArcGIS Pro compatibility

Script Editor

ModelBuilder Options

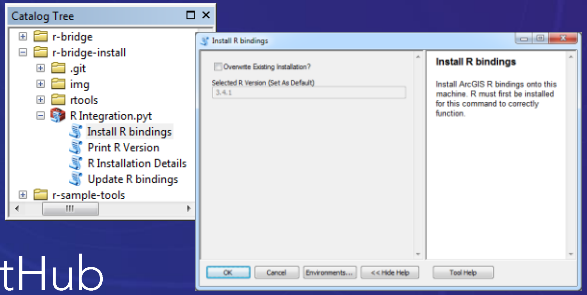
☐ Do not show warning when overwriting model from previous version

R-ArcGIS Support

Detected R home directories: [R-3.4.1] C:\Program Files\R\R-3.4.1

Please install the ArcGIS R integration package

- In ArcGIS 10.3.1+ (e.g., ArcMap):
  - Connect To r-bridge-install folder
  - Open R Integration.pyt toolbox
  - Execute the "Install R Bindings" tool



☐ Overwrite Existing Installation?

Selected R Version (Set As Default): 3.4.1

**Install R bindings**

Install ArcGIS R bindings onto this machine. R must first be installed for this command to correctly function.

- Detailed instructions: r-bridge-install on GitHub
  - https://github.com/R-ArcGIS/r-bridge-install

esri Canada

# R-ArcGIS: arcgisbinding package

- Opening ArcGIS datasets in R:

```
gis_data <- arc.open(path="D:/data/gis_data.shp")
```

- Read data into R data frames:

```
R_data <- arc.select(gis_data, fields="*", where_clause="",
                     selected="TRUE", sr=NULL)
```

- Convert to/from **sp** data frames:

```
sp_data <- arc.data2sp(R_data)
R_data <- arc.data2sp(sp_data)
```

- Write data frames back to ArcGIS:

```
arc.write(path="D:/data/gis_data.shp", data=R_data)
```

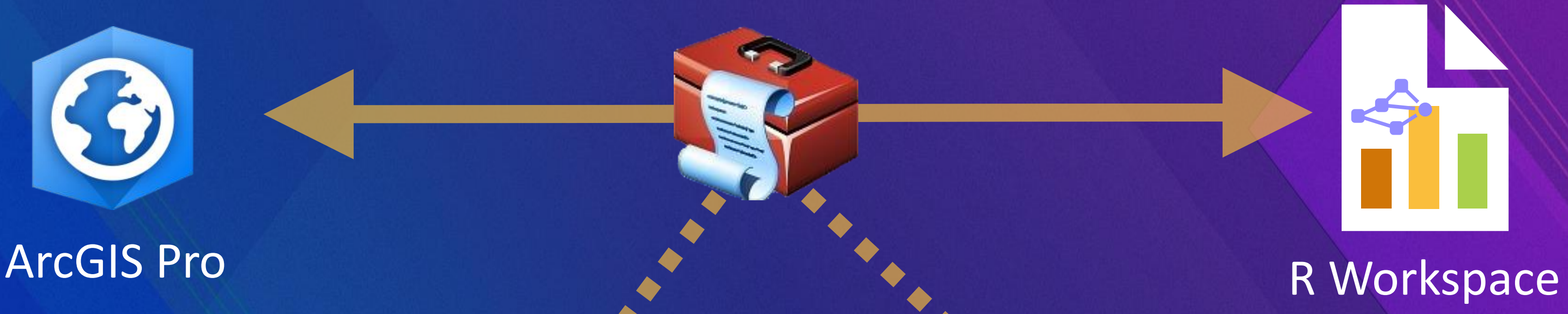
- Convert projections between WKT & Proj.4:

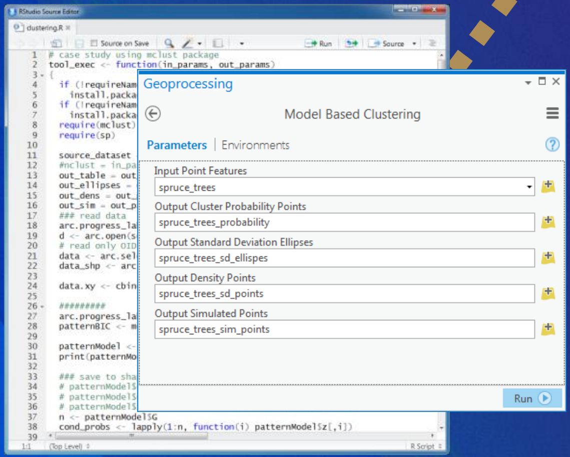
```
arc.fromP4ToWkt("+proj=latlong +datum=wgs84")
arc.fromWktToP4(gis_data$shapefinfo$WKT)
arc.fromWktToP4(4326)
```

esri Canada

# R-ArcGIS: Script Tools

- Enables use of R statistical functionality from the ArcGIS Platform
  - Directly executed as script tools with input/output parameters
  - Integration of R script tools into ArcGIS model builder workspaces
  - Simplification and Automation of complex tasks





```
# case study using mclust package
tool_exec <- function(in_params, out_params)
{
  if (!requireNam
    install.packa
  if (!requireNam
    install.packa
  require(mclust)
  require(sp)

  source_dataset
  #nclust = in_pa
  out_table = out
  out_ellipses =
  out_dens = out_
  out_sim = out_p
  ### read data
  arc.progress_la
  d <- arc.open(s
  # read only OID
  data <- arc.sel
  data_shp <- arc

  data.xy <- cbin

  #########
  arc.progress_la
  patternBIC <- m

  patternModel <-
  print(patternMo

  ### save to sha
  # patternModel$
  # patternModel$
  # patternModel$
  n <- patternModel$G
  cond_probs <- lapply(1:n, function(i) patternModel$z[,i])
```

Geoprocessing

Model Based Clustering

Parameters | Environments

Input Point Features: spruce_trees

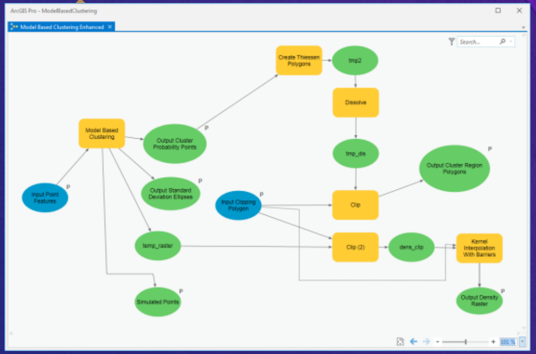
Output Cluster Probability Points: spruce_trees_probability

Output Standard Deviation Ellipses: spruce_trees_sd_ellispes

Output Density Points: spruce_trees_sd_points

Output Simulated Points: spruce_trees_sim_points

Run


ArcGIS Pro - ModelBasedClustering
Model Based Clustering Enhanced
Search...
Create Thiessen Polygons
tmp2
Dissolve
Model Based Clustering
Output Cluster Probability Points
tmp_dis
Output Cluster Region Polygons
Input Point Features
Output Standard Deviation Ellipses
Input Clipping Polygon
Clip
temp_raster
Clip (2)
dens_clip
Kernel Interpolation With Barriers
Simulated Points
Output Density Raster
P
101 %

# R-ArcGIS: Script Tools

- Basic Components:
  - Main entry point for an R script configured as a script tool for ArcGIS:

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

  - Input/output parameters are provided as list objects:

```
in_params <- list( ... )
out_params <- list( ... )
```

  - Use **`arc.open()/arc.select()`** to load feature classes / tables:

```
 in_data <- arc.open( in_params$in_data, ... )
```

# R-ArcGIS: Script Tools

- Output:
  - Use **arc.write()** to output feature classes / tables:

    ```
    arc.write( out_params$out_data, data, ... )
    ```

  - Directly assign other output values, or paths to feature classes / tables specified as derived parameters:

    ```
    out_params$result <- sum(data$values)
    out_params$out_data <- "C:/data/output.shp"
    ```

  - Update Geoprocessing progress indicator:

    ```
    arc.progress_label("Analysis in progress...")
    arc.progress_pos(50)
    ```

  - Output parameters must be returned at the end:

    ```
    return(out_params)
    ```

# R-ArcGIS: Script Tools

- Debugging:
  - Use **print()** or **message()** to display text in the Geoprocessing window:

    ```
    message(paste(c("Input value: ", in_val), collapse=""))
    ```

  - Use **warning()** to display warnings:

    ```
    warning("Input parameter not set, defaulting to x ...")
    ```

  - Use **stop()** to raise errors:

    ```
    stop("Invalid input data.")
    ```

  - Use the R **tryCatch()** method to handle errors and warnings:

```
x <- tryCatch(
  { x <- y     # try something here ... },
  warning = { # handle warnings... },
  error =    { # handle errors... },
  finally = { # do something else afterwards... }
)
```

esri Canada

# R – Resources

- No need to memorize anything:
  - Cheatsheets: https://www.rstudio.com/resources/cheatsheets/
  - Package Vignettes – E.g.:
    - Geographically Weighted Regression (spgwr): https://cran.r-project.org/web/packages/spgwr/vignettes/GWR.pdf
    - Spatial Inequalities (lctools): https://cran.r-project.org/web/packages/lctools/vignettes/SpatialInequalities.pdf
  - Samples, Blogs, GeoNet, Videos/Tutorials, etc.:
    - https://github.com/R-ArcGIS/r-sample-tools
    - https://github.com/R-ArcGIS/CHANS-tools
    - https://www.r-bloggers.com/
    - https://www.r-project.org/help.html
    - https://geonet.esri.com/groups/rstats
    - http://hed.esri.ca/resourcefinder/#/search=r/lang=en