Getting Started with the R-ArcGIS Bridge

Tutorial Overview

R is an open-source statistical computing language that offers a large suite of data analysis and statistical tools. You can use R to extend the capabilities of ArcGIS and allow for more robust spatial and non-spatial data analysis. In this tutorial, a process will be outlined which will guide you through configuring ArcGIS with the R-ArcGIS bridge to enable you to access ArcGIS datasets in R, install third-party R packages, and execute R tools configured as script tools in an ArcGIS toolbox. By establishing this link, you will be able to develop R programs that interoperate with ArcGIS datasets, and create custom geoprocessing tools that use R functionality within an ArcGIS Pro or ArcGIS Desktop environment.

Skills

By completing this tutorial, you will become comfortable with the following skills:

- Installing and setting up R and RStudio
- Installing the R-ArcGIS Bridge
- Installing and loading third-party R packages.
- Running basic R script tools from an ArcGIS toolbox

Time Required

The following classroom time is required to complete this tutorial:

- 45 - 60 minutes

Materials Required

Technology:

- ArcGIS Pro 1.1+ (2.0+ recommended) or ArcGIS Desktop 10.3.1+ (10.5.1+ recommended)
- Internet browser (e.g., Mozilla Firefox, Google Chrome, Safari)
- R Statistical Computing Language – version 3.2.0+
- RStudio Desktop – version 1.0.153+

Data:

- Data for this tutorial are included as part of the download in the packaged *data* folder.

Data Sources

- City of Toronto:
  https://www1.toronto.ca/wps/portal/contentonly?vgnextoid=1a66e03bb8d1e310VgnVCM10000071d60f89RCRD

Production Date

The Education and Research Group at Esri Canada makes every effort to present accurate and reliable information. The Web sites and URLs used in this tutorial are from sources that were current at the time of production, but are subject to change without notice to Esri Canada.

- Production Date: November 2017

Background Information

ArcGIS offers a multitude of statistical tools for data analysis. These tools are especially useful when working with GIS data, as they can help you extract information that might not be apparent by looking at a map. The tools available in ArcGIS include basic descriptive statistics (e.g., mean, median, standard deviation), and a variety of spatial statistics, many of which can be found in the Spatial Statistics Toolbox. You are encouraged to use these built-in tools where applicable.

While these resources can be beneficial, certain common non-spatial analysis techniques (e.g., ANOVA, correlation, etc.) are not available from within ArcGIS. If you need to apply such techniques, you may have to draw on other statistical software, and exporting data from ArcGIS to a third-party system to perform a simple test can be time consuming and sometimes cumbersome. Being able to communicate with other statistical software systems directly from within ArcGIS helps to alleviate this problem. This is made possible by the R-ArcGIS Bridge by connecting ArcGIS to the R statistical computing software.

R is an open-source statistical computing language that offers a large suite of data analysis and statistical tools. R also has an active and growing user community of data scientists, academics, and software engineers who are constantly implementing new tools and techniques that keep R up to date with the latest research methods. Harnessing these tools can reduce some of the work of performing statistical tests, and is a reason to consider using R with ArcGIS.

Setting up the R-ArcGIS Bridge to establish the link between ArcGIS and R is a relatively straightforward process. By completing this tutorial, you will have successfully installed the R-ArcGIS Bridge. This initial setup is a requirement for associated tutorial that teach you to work with ArcGIS data in R, and create custom R tools that can be execute from ArcGIS.

References and Reading

- Statistical Analysis with ArcGIS
  http://desktop.arcgis.com/en/arcmap/latest/analyze/commonly-used-tools/statistical-analysis.htm

- The Spatial Statistics Toolbox
  http://pro.arcgis.com/en/pro-app/tool-reference/spatial-statistics/an-overview-of-the-spatial-statistics-toolbox.htm

- Advanced Spatial Analysis Labs
  https://hed.esri.ca/resourcefinder/#/title=Advanced%20Spatial%20Analysis%20Labs/lang=en

- R Statistical Computing Language
  http://www.r-project.org/

- R-ArcGIS GitHub Website:
  https://r-arcgis.github.io/

Part A: Installing R and RStudio software

First, you will need to download and install R on your Windows PC. ArcGIS Pro and ArcGIS Desktop are only available for Windows, so this tutorial is aimed at Windows users. Administrative privileges will be required to install the software.

1. Navigate to the R-Project website Windows download page (http://cran.r-project.org/bin/windows/base/). Download the latest version of **R** using the link near the top of the page.

2. Once complete, open the R setup file, and install it in the default location.

   - **Important:** Make sure to note the location on your hard drive that you are installing R to. This is sometimes referred to as the R_HOME directory, and this will be referenced again later in this tutorial.

   - Complete the setup by accepting the remaining default settings.

Note: You will be presented the option to install 32-bit files and/or 64-bit files. 32-bit is required for ArcGIS Desktop 10.3.1 application. 64-bit is required for ArcGIS Pro 1.1+ or with background geoprocessing enabled in ArcGIS Desktop 10.3.1+. If your operating system supports 64-bit, you can install both. You can find out your system type from the System page in the Windows Control Panel.

Open R to make sure the installation was successful:

3. From the Windows **Start Menu**, and click *All Programs > R > R 3.x.x*. The exact name will depend on the version of R you installed, and whether it is 32 or 64-bit (e.g., 32-bit will be **R i386**, and 64-bit will be **R x64**). If you created a Desktop shortcut during installation, you can use this as well.

4. A program entitled **RGui** (Graphic User Interface) should start up with an R console window into which you can enter text.

5. Close **RGui**. If prompted to "Save workspace image?", click *No*.

Next, download and install the **RStudio Desktop** IDE:

6. Navigate to the **RStudio** download page (https://www.rstudio.com/products/rstudio/download/), and download the latest windows installer for the open source version of **RStudio Desktop**.

7. Once complete, run the installer for **RStudio Desktop**, and accept default options.

8. Open **RStudio** - from the Windows **Start Menu**, click *All Programs > RStudio > RStudio*

9. You should be presented with an IDE interface that includes several panels, including an R console window. The R console window should display the version, license, and help information displayed by default (the same text that appears in the console when opened in the **RGui** application).

Note: RStudio is not required to use the R-ArcGIS bridge. However, it provides an integrated development environment that introduces many user-friendly features (syntax highlighting, code completion, debugging, etc.) that are not provided by the default RGui application that comes with R. Portions of this workshop that focus on editing/executing R code will be discussed in the context of an RStudio Desktop project, but can all be completed using only the base R software environment.

# Part B: Installing the R-ArcGIS Bridge

The simplest way to install the R-ArcGIS Bridge is available with **ArcGIS Pro 2.0+**, as it enables you to install the **R-ArcGIS Bridge** directly from ArcGIS Pro's Geoprocessing options. An alternative to this (i.e., for **ArcGIS Pro 1.x** or **ArcGIS Desktop 10.3.1+**) is to install it using geoprocessing tools in a python toolbox that is available in an **r-bridge-install** repository available in the **R-ArcGIS** GitHub account.

At the end of this part of the tutorial, you will have a package named `arcgisbinding` installed into your **R** environment. This is the library that is used from within the **R** programming language to access the functionality provided by the R-ArcGIS Bridge.

*Note: It is not necessary to install the R-ArcGIS Bridge for both **ArcGIS Pro** and **ArcGIS Desktop**. Once it is installed, the `arcgisbinding` package will be available in the **R** environment that can be used by either **ArcGIS Pro** or **ArcGIS Desktop**.*

To install directly from **ArcGIS Pro 2.0+**:

1. Open **ArcGIS Pro** - from the Windows **Start Menu**, click *All Programs > ArcGIS > ArcGIS Pro > ArcGIS Pro*
2. When prompted to open a recent project or create a new project, click the 'About ArcGIS Pro' link in the bottom left-hand corner of the screen – or – with an existing project opened, click the 'Project' tab in the ribbon at the top of the screen:
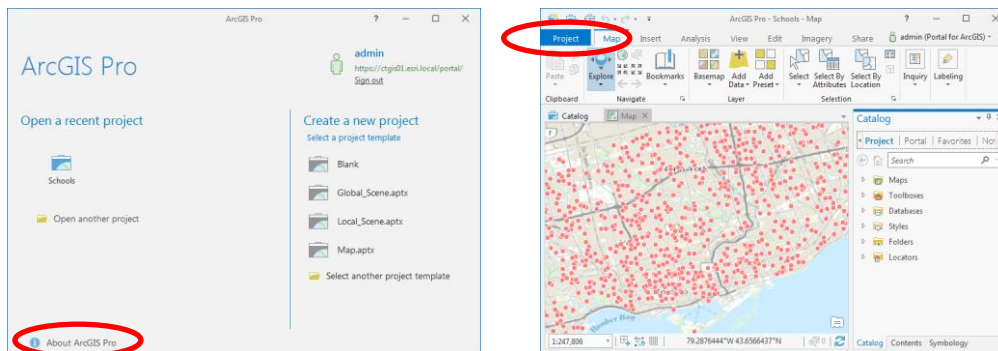
3. From the *About ArcGIS Pro* screen, choose *Options*, and select the *Geoprocessing* section in the left-hand menu of the Options dialog.

4. In the *Geoprocessing* options, a section is dedicated to *R-ArcGIS Support*.
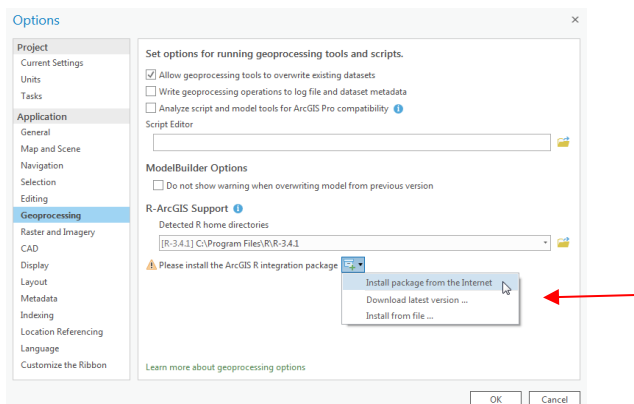
The `R_HOME` directory where you installed **R** in Part A of this tutorial should appear automatically in the list of *Detected R home directories* (it may take a few seconds to appear). If the directory does not appear automatically, you can click the browse button (📂) to the right of the path input, and navigate to the location where you installed R (typically, this will be C:\Program Files\R\R-3.x.x). If you have multiple installations of R, you can choose the version that you want to use.

If the `arcgisbinding` package is not detected with the version of R that you choose, you will be prompted to install it from the menu displayed by clicking the button next to the prompt (🗅). If it is already installed, you will see the version number displayed, and you can update it from the menu displayed from button next version number (🗅).

5. When you choose to install or update, you can do so automatically from the Internet by selecting "Install Package from the Internet" or "Check package for updates" in the corresponding menu. If your machine is not connected to the Internet, you can install from a local copy of the package (`arcgisbinding_x.x.x.x.zip`) package by choosing to "Install from file…" or "Update from file…". This file can be obtained using a machine that is connected to the Internet by downloading the latest `arcgisbinding_x.x.x.x.zip` file from the r-bridge GitHub repo, then copying it to the offline machine running ArcGIS Pro: https://github.com/R-ArcGIS/r-bridge/releases/latest

   Note: a copy of `arcgisbinding_1.0.0.128.zip` is included in the resources folder along with this tutorial.

6. If you have successfully completed these steps, and the installed version of the `arcgisbinding` package is detected and displayed in the ArcGIS Pro Geoprocessing options, you can proceed to Part C of this tutorial.

To install the **R-ArcGIS Bridge** in **ArcGIS Pro 1.x** or **ArcGIS Desktop 10.3.1+**, you can obtain the latest copy of the **r-bridge-install** repository from the **R-Bridge** GitHub account.
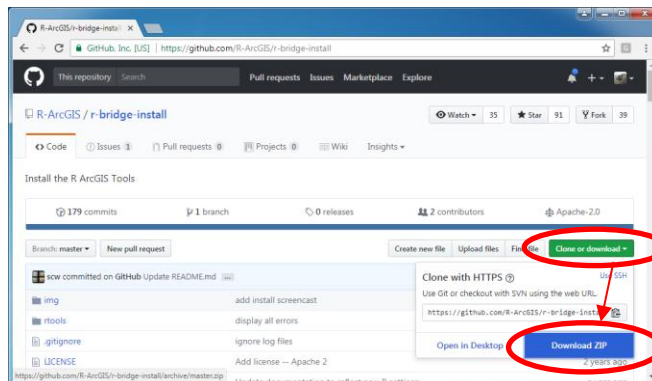
*Note: A copy of the **r-bridge-install** repository is included in the resources folder included with this tutorial.*

1. In a Web browser, visit the **r-bridge-install** repository on GitHub at the following URL:

   https://github.com/R-ArcGIS/r-bridge-install
   Note: a copy of the r-bridge-install repository is included in the resources folder along with this tutorial.

2. Click the *Clone or download* button, and select "Download ZIP":

3. Save the zip file and extract its contents to a location on your hard drive (e.g., `D:\r-bridge-install` or `D:/r-bridge-install-master`).

4. Now that the **r-bridge-install** repository files are downloaded, you must choose to install

In **ArcGIS Pro**:

   a. Open **ArcGIS Pro** - from the Windows **Start Menu**, click *All Programs > ArcGIS > ArcGIS Pro*

   *Note: ArcGIS Pro 1.x users may require administrative privileges to run the install process. To do so, Right-click the **ArcGIS Pro** shortcut in the Window **Start Menu**, and click **Run as administrator**. Accept the security prompt that appears. (This is only required setting up the R-ArcGIS Bridge.)*

   b. Open an existing project or create a new project.
   c. Open the *Project* pane (*View > Project Pane*)
   d. In the project section of the *Project* pane, right-click on *Toolboxes* and choose *Add Toolbox…*
   e. Navigate into the folder location containing the **r-bridge-install** repository files, select the `R Integration.pyt` toolbox, and click OK.
   f. In the *Project* pane, navigate into the `R Integration.pyt` toolbox that is now visible in the *Toolboxes* section, and double-click the *Install R bindings* geoprocessing tool.
   g. In the geoprocessing dialog that opens, click Run.

In **ArcGIS Desktop 10.3.1+**:

   a. Open **ArcMap** - from the Windows **Start Menu**, click *All Programs > ArcGIS > ArcMap 10.x.x* (if prompted to open a New Document, click **Cancel**)
   b. Open the *Catalog* window (either click the *Catalog* button (⬛) in the main toolbar, or select *Windows > Catalog* from the main menu)
   c. Click the *Connect To Folder* button (⬛) and navigate to the folder location containing the **r-bridge-install** repository files, and click OK
   d. In the *Catalog* window, navigate into the `R Integration.pyt` toolbox, and double-click on the *Install R bindings* geoprocessing tool.
   e. In the geoprocessing dialog that opens, choose the version of R that you want to install the bindings into (applicable only if you have multiple versions of R installed), and click **OK**

The same steps noted above for **ArcGIS Pro 1.x** and **ArcGIS Desktop 10.3.1+** can be used to update the version of the **R-ArcGIS Bridge** that is installed by executing the **Update R bindings** tool in the **R Integration.pyt** toolbox.
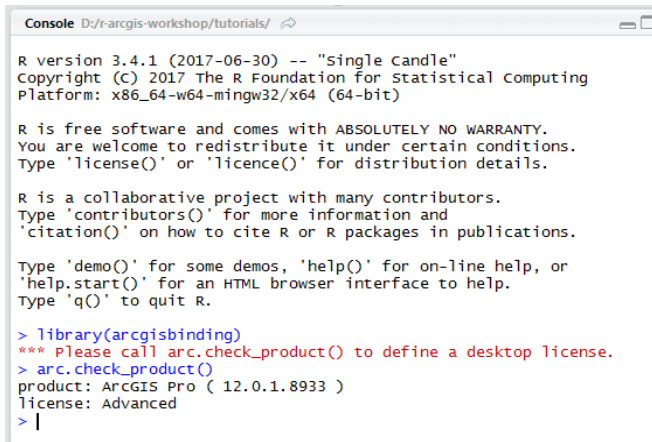
*Note: Both the **Install R bindings** and **Update R bindings** tools will automatically install the bindings from the Internet. If you are not connected to the Internet, you can install a local copy of the `arcgisbinding_x.x.x.x.zip` file by downloading it (from https://github.com/R-ArcGIS/r-bridge/releases/latest) using a browser on a computer connected to the Internet, then placing a copy of the zip file inside the **r-bridge-install** repository files (i.e., in the same folder as the `R Integration.pyt` toolbox). A copy of the zip file that is included in the resource files for this tutorial may be used.*

Now that the `arcgisbinding` package has been installed, verify that it works correctly in the **R** environment.

1. Open the **RStudio** (or **R**) and enter the following command into the console to load the `arcgisbinding` package:

```
library(arcgisbinding)
```

2. You should now see a message printed in the console that prompts you to call `arc.check_product()` – This method must be called to bind the session to an ArcGIS Installation before any of the methods provided by the `arcgisbinding` package will function. Execute this command in the console – when the method is completed, it will display a message indicating which ArcGIS product has been bound to, and `arcgisbinding` methods will now be available to use in the current R session.

```
Console D:/r-arcgis-workshop/tutorials/

R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> library(arcgisbinding)
*** Please call arc.check_product() to define a desktop license.
> arc.check_product()
product: ArcGIS Pro ( 12.0.1.8933 )
license: Advanced
> |
```

3. Test loading ArcGIS datasets in **R**, use the `arc.open()` method to open existing datasets, and the `arc.select()` method to read records from feature layers, feature classes or tables into R data frame objects. To do so, execute the following lines of code in the console (substitute the `'D:/r-arcgis/'` part of the path to the shapefile to reflect where you have extracted the tutorial files on your machine):

```
library(arcgisbinding)
arc.check_product()
to_nbrhds <- arc.open('D:/r-arcgis/data/toronto/neighbourhoods.shp')
to_nbrhds_df <- arc.select(to_nbrhds)
View(to_nbrhds_df)
```

After the `View()` command is executed, the a table will be presented showing the attributes associated with all of the neighbourhood features that were loaded from the shapefile.

Part C: Installing Third-party Packages in R

To perform certain tasks in R, additional functions or data are often required. R relies on its large user community to develop 'packages' that extend additional functionality to R. A 'package' is simply a user-created collection of different functions, data, and/or precompiled code (written in Java, C, Fortran) that seeks to address some problem (e.g., manipulating and reading in geographic data). As of September 2017, there are over 11,000 user-contributed packages available to the public from the Comprehensive R Archive Network (CRAN). You can easily download and use any these packages in from the **R** (gui) or **RStudio** applications, or from any **R** console using the `install.packages()` command.

To install packages in the **RGui** application:

esri Canada

1. Open **R**

2. In the **RGui** menu bar, select *Packages > Install package(s)…* A dialog will appear that prompts you to select a mirror to download the packages from.

3. Choose a mirror located closest to you, and click **OK**.

4. You will now be presented with a list of all the free packages available for R through the Comprehensive R Archive Network (CRAN). Scroll through the list of packages and select **rms** (a package used for regression modeling, testing, estimation, validation, graphics, prediction, and typesetting). Click **OK**, and the package should begin downloading and installing. Multiple packages can be selected at a time by holding the CTRL key and clicking on the desired packages.

5. If you are prompted to use a personal library when trying to install your first package click **Yes**. In the following prompt you will be asked for a location to create a personal library. Accept the default location and click **Yes**. This will create a personal library directory where you will store packages for your user account.

To install packages in the **RStudio** IDE:

1. Open **RStudio**

2. In the **RStudio** menu bar, select **Tools** > **Install packages…** A dialog will appear that provides the following options:

   - Choose to install configured repositories (CRAN and CRANextra by default), or from a package archive file.

   - An input to specify the names of packages to install (or to specify the path to a local copy of a package archive)

   - The library to install the packages into

   - An option to choose whether to install dependencies.

3. Accept the default option to install from the CRAN/CRANextra repositories, leave the library set to the default location, and install dependences.

4. Type **rms** into the packages text field (a package used for regression modeling, testing, estimation, validation, graphics, prediction, and typesetting). Click **Install**, and the package should begin downloading and installing along with its dependencies. Multiple packages can be installed at once by typing all of them into in the dialog separated by spaces or commas.

To install packages from the **R** console:

1. Open **R** or **RStudio**:

2. Type the following command into the console, and press Enter:

```
install.packages('rms')
```

Alternatively, to install multiple packages at once, provide a vector containing package names:

```
install.packages(c('sp', 'maptools', 'spgwr'))
```

3. To avoid being prompted to select a mirror, you can specify a default repository mirror as one of the arguments in the command:

```
install.packages('spgwr', repos='http://cran.r-project.org')
```

Once packages are installed, they must be loaded before any of their functions can be used in an R session or script. To do so, you can execute `library(<packagename>)` to load the package in the current R session.

You will make use of a variety of packages in R as you continue working through the R-ArcGIS tutorial series that follows this *Getting Started* tutorial.

Note: In the resources folder packaged with this tutorial, an `install_packages.R` *script is included that will install all R packages that are used by the series of R-ArcGIS tutorials and included samples that accompany this Getting Started tutorial. The* `install_packages.R` *script may optionally be executed to install these packages, allowing each tutorial to be completed without the need for an Internet connection. This can be accomplished by executing* `source("resources/install_packages.R")` *from any R console. For this to work, you must either specify the full path to the script, the current working directory must be the folder containing the R-ArcGIS tutorial files. You can determine this by executing* `getwd()` *to report the current working directory. If needed, you can set the current working directory by executing* `setwd("C:/path/to/r-arcgis-tutorial")` *with the full path to the folder containing the tutorial files supplied as the argument.*

Part D: Extract Sample Data

The tutorial files include sample datasets that are used throughout this tutorial series. These datasets are packaged in the `data/data.zip` file. Extract the contents of this zip file into the `data/` folder within the tutorial files. This should produce two sub-folders: `data/census/` and `data/toronto/`. Specific steps to extract the files will vary depending on your system configuration.

Using Windows Explorer:

1. Open Windows Explorer and navigate into `data/` folder within the tutorial files
2. Double-click on the `data.zip` file.
3. If no third-party archiving software is installed (e.g., 7-Zip), Windows Explorer will display the contents of the zip file (similar to viewing the contents of regular folder). You should see both `census` and `toronto` folders inside the zip file.
4. Click on one of the folders and press **CTRL+A** (or click and drag the mouse) to select both folders.
5. Press **CTRL+C** (or right-click and choose *Copy*) to copy the two folders.
6. Click the back button in Windows Explorer, or navigate directly back to the `data/` folder containing the `data.zip` file.
7. Press **CTRL+P** (or right-click and choose *Paste*) to paste the folder copied from the `data.zip` file.

Using 7-Zip (with its default shell-extension enabled):

1. Open Windows Explorer and navigate into `data/` folder within the tutorial files
2. Right-click on the `data.zip` file, and choose *7-Zip > Extract Here*

esri Canada

Part E: Running sample R tools in ArcGIS Pro or ArcGIS Desktop

Now that you have the R software and the **R-ArcGIS** bridge installed on your computer, you can test some of the sample R tools available from the R-ArcGIS GitHub account. You can follow the same steps outlined in Part B of this tutorial to download and extract the latest **r-arcgis-install** repository files to obtain the latest **r-sample-tools** repository available at the following URL:

https://github.com/R-ArcGIS/r-sample-tools
*Note: a copy of the **r-sample-tools** repository is included in the resource files for this tutorial.*

To test the Model Based Clustering script tool, for example:

In **ArcGIS Pro 1.1+**:

1. Open the `ModelBasedClustering.arpx` ArcGIS Pro project file included in the **r-sample-tools** repository files (either double-click the file in **Windows Explorer**, or open the file from an open **ArcGIS Pro** session).

2. In the Project pane (1.x) or Catalog pane (2.0+), navigate into *Project > Toolboxes > R Sample Tools.tbx*, and double-click on the *Model Based Clustering* script tool.
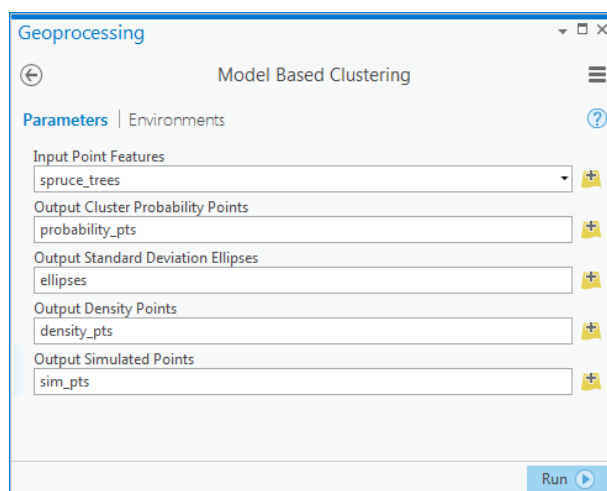
3. In the geoprocessing dialog that opens, select the `spruce_trees` layer in the first input parameter, specify a path for the output probability points layer, and optionally outputs for ellipses, density points, and simulation points layers, then click Run.

4. The tool should run successfully, with the cluster analysis result layers added to the map.

In **ArcGIS Desktop 10.3.1+**:

1. Open the `ModelBasedClustering.mxd` (double-click on the file in Windows Explorer, or open **ArcMap** and choose File > Open and select the `*.mxd` file)

    *Note: the `*.mxd` may not load successfully in some versions of **ArcMap**. If this happens, start a new*

blank map in **ArcMap**, save it as a new `*.mxd` file inside the **r-sample-tools** repository folder, and add the `spruce_trees` feature class from the sample `data.gdb` file geodatabase as a layer in the map.

2. In the ArcMap catalog window, navigate into the Home directory (which will be the **r-sample-tools** repository folder), open the **R Sample Tools.tbx**, and double-click *Model Based Clustering* script tool.

3. In the geoprocessing dialog that opens, select the `spruce_trees` layer in the first input parameter, specify a path for the output probability points layer, and optionally outputs for ellipses, density points, and simulation points layers, then click OK.

4. The tool should run successfully, with the cluster analysis result layers added to the map.

When executed in ArcGIS Desktop, the script tool might abort and print the following message:

```
*** Please install ArcGIS R integration package
Failed to execute (ModelBasedClustering).
```

If this occurs, it is because **ArcGIS Desktop** is not detecting the same **R** library path where your `arcgisbinding` package is installed. You can identify this path by executing `.libPaths()` from any **R** console, which will display a result like:

```
[1] "C:/Users/<username>/R/win-library/3.4"  "C:/Program Files/R/R-3.4.1/library"
```

The first location in this vector (with your username in the path) is likely the location where the `arcgisbinding` has been installed. Next, you can identify the library paths that **ArcGIS Desktop** is searching by running the check_arcgisdbinding_path.py included in the scripts folder packaged with this tutorial. For example, from a command window, execute the following:

```
C:\Python27\ArcGIS10.5\python.exe check_arcgisbinding_path.py
```

The output will display default 'Documents' directories associated with your profile. A likely result will look something like:

```
C:\Users\<username>\Documents
```

To make the `arcgisbinding` package visible to **ArcGIS Desktop**, create a matching win-library path inside of the Documents path identified above (it may be aliased as 'My Documents' in **Windows Explorer**). For example:

```
C:\Users\<username>\Documents\R\win-library\3.4
```

Copy the `arcgisbinding` package folder from its installed location (i.e., the folder identified by the `.libPaths()` method above), and paste it into the new folder you created in your Documents path. When you try to run the sample script tool now, it should work as expected.

If **ArcGIS Desktop** still does not detect the **arcgisbinding** package, you can place a copy of the package inside of the library directory of the **R_HOME** installation folder (i.e., the second location returned from the `.libPaths()` method above). You will require administrative privileges to copy the files here. This location will ensure that the `arcgisbinding` package is accessible to **ArcGIS Desktop**, and you will be able to successfully run the script tools.

Note: if you employ this workaround, you will need to manually update the copies of the **arcgisbinding** package files if/when you install an updated version of the **R-ArcGIS** bindings.

You can continue experimenting with the **R Sample Tools.tbx** toolbox, for example to view how the script tool can be integrated into an ArcGIS Model Builder workspace (e.g., the Model Based Clustering Enhanced tool). Try experimenting with the Semiparametric Regression tool. You can also inspect and experiment with the R code behind the script tools (see the `*.R` files in the `r-sample-tools/scripts` that are referenced by the script tools defined in the toolbox).

Future Considerations

R is a powerful tool with an abundance of resources that can be drawn from to extend the functionality of ArcGIS. This tutorial has outlined the process of configuring R with ArcGIS, but you should continue working with R in improve your analytical skills and to create your own tools.

To learn how to create your own **R** tools that will run within **ArcGIS Pro** or **ArcGIS Desktop**, check out the **Building an R Script Tool** tutorial. If you need to learn more about R first, check out the **R Basics** tutorial.

http://hed.esri.ca