Extending ArcGIS with Conda and R

Shaun Walbridge

https://github.com/esrioceans/oceans-workshop-2016

Download PDF

Extending ArcGIS

Extending ArcGIS

- ArcGIS is a *system of record*. Combine data and analysis from many fields and into a common environemnt.
- Why extend? Can’t do it all, we support over 1000 GP tools — enabling *integration* with other environments to extend the platform.
- Discussed SciPy and what’s “in the box”, now discuss the broader ecosystem of open source.

Today: R and Conda

- R-ArcGIS Bridge
  - Short Introduction
  - Demo
- Conda for managing packages and environments
  - Introduction
  - Demo
  - Exercise

R

Why R?

- Powerful core data structures and operations
  - Data frames, functional programming
- Unparalleled breadth of statistical routines
  - The *de facto* language of Statisticians
- `CRAN`: 6400 packages for solving problems
- Versatile and powerful plotting

R Data Types

Data types you’re used to seeing…

`Numeric` - `Integer` - `Character` - `Logical` - `timestamp`

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`

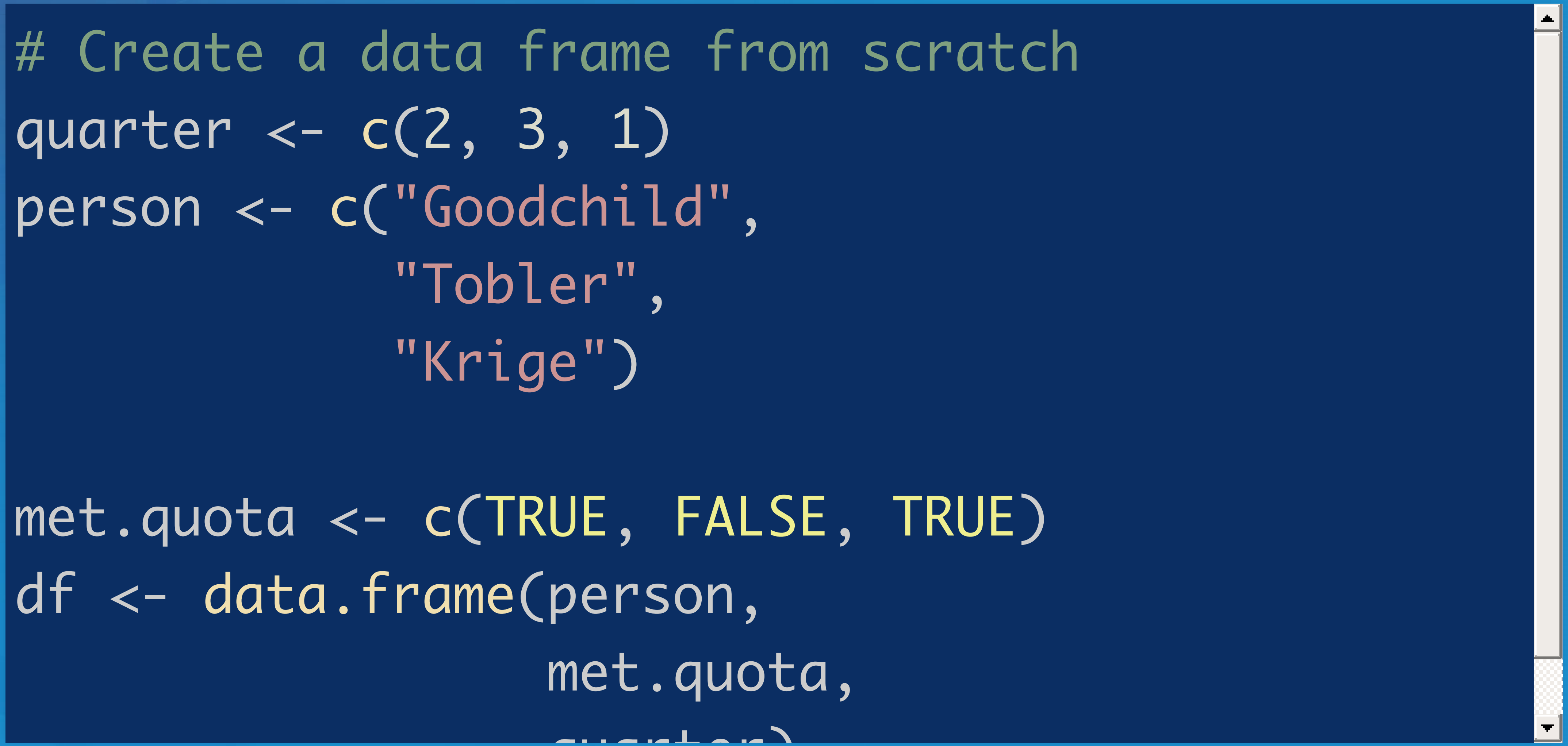
Data Frames

- 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)

Data Types

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

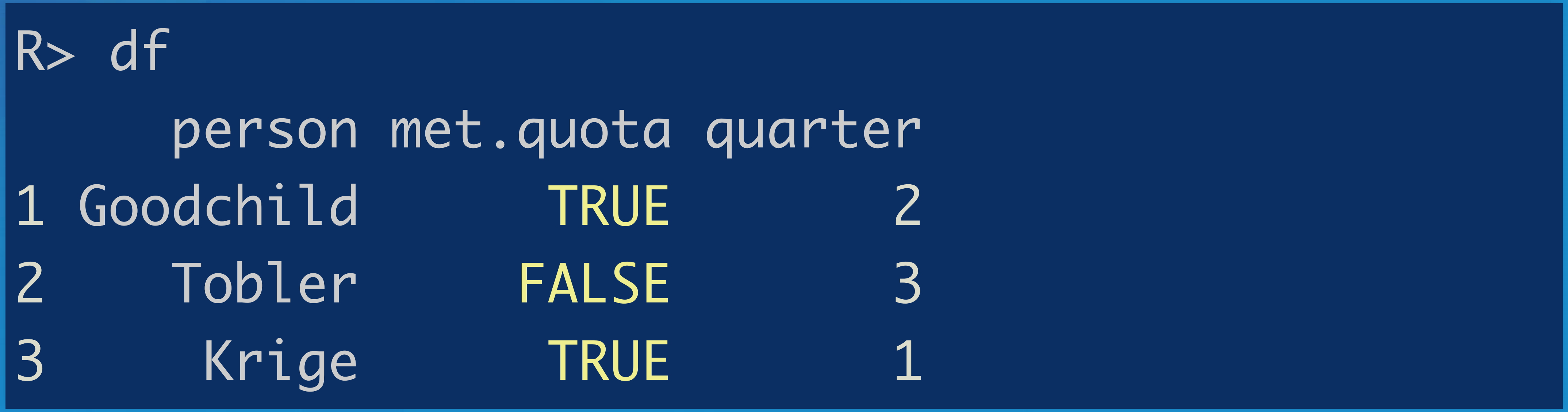
Data Types



```
# 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)
```

Data Types



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

`sp` Types

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

Entity + Attribute model

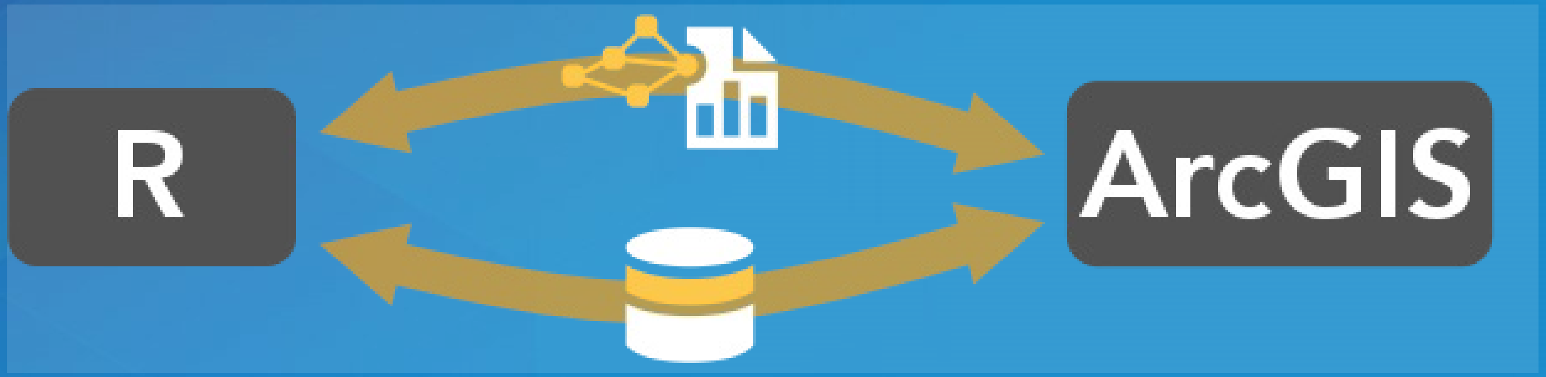
Statistical Formulas

```
fit.results <- lm(pollution ~ elevation + rainfall
```

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

R — ArcGIS Bridge

R — ArcGIS Bridge



- ArcGIS developers can *create 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

R — ArcGIS Bridge

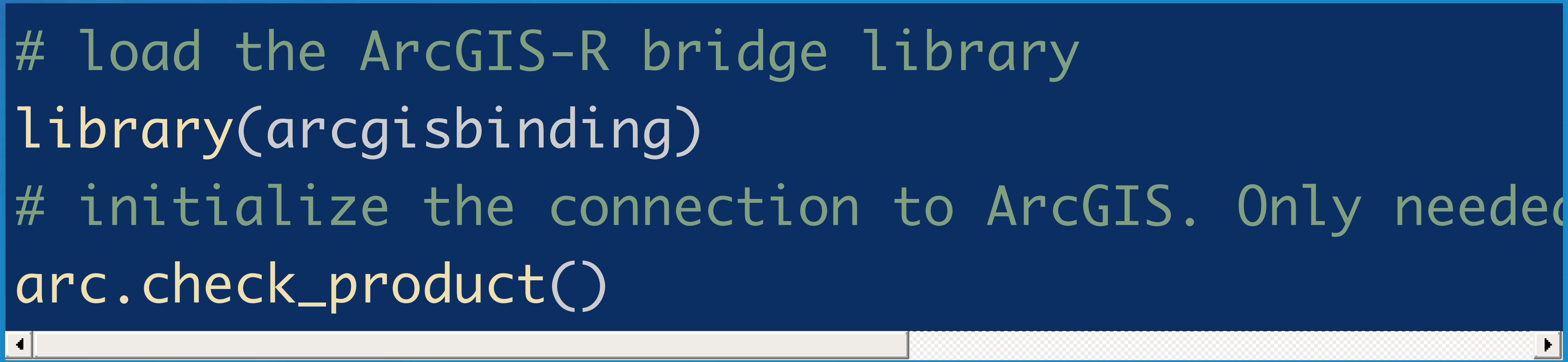
Store your data in ArcGIS, access it quickly in R, return R objects back to ArcGIS native data types (e.g. geodatabase feature classes).

Knows how to convert spatial data to `sp` objects.

Package Documentation

Access ArcGIS from R

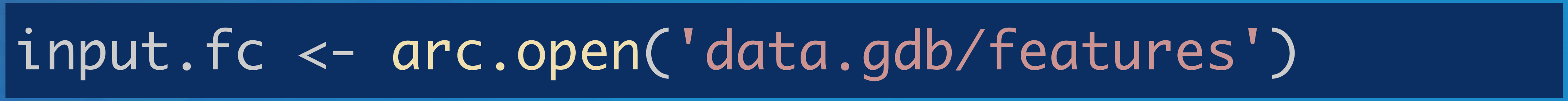
Start by loading the library, and initializing connection to ArcGIS:



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

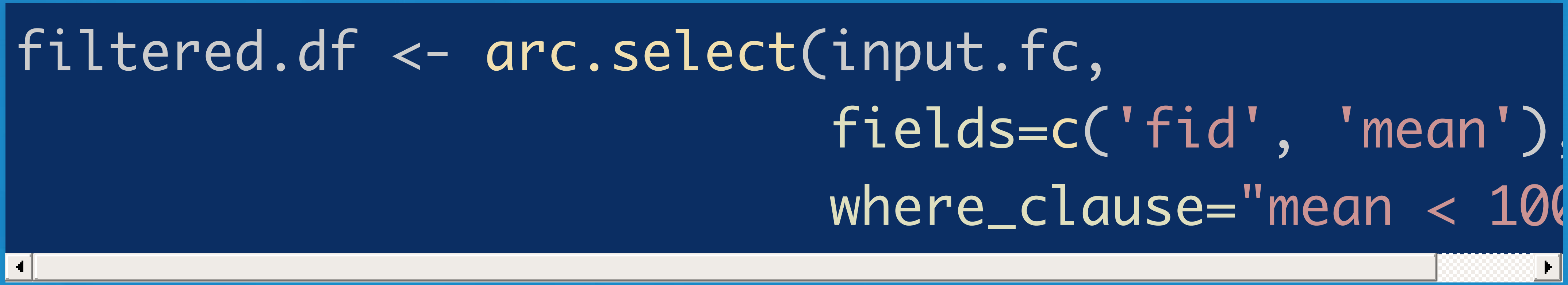
Access ArcGIS from R

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 < 10
```

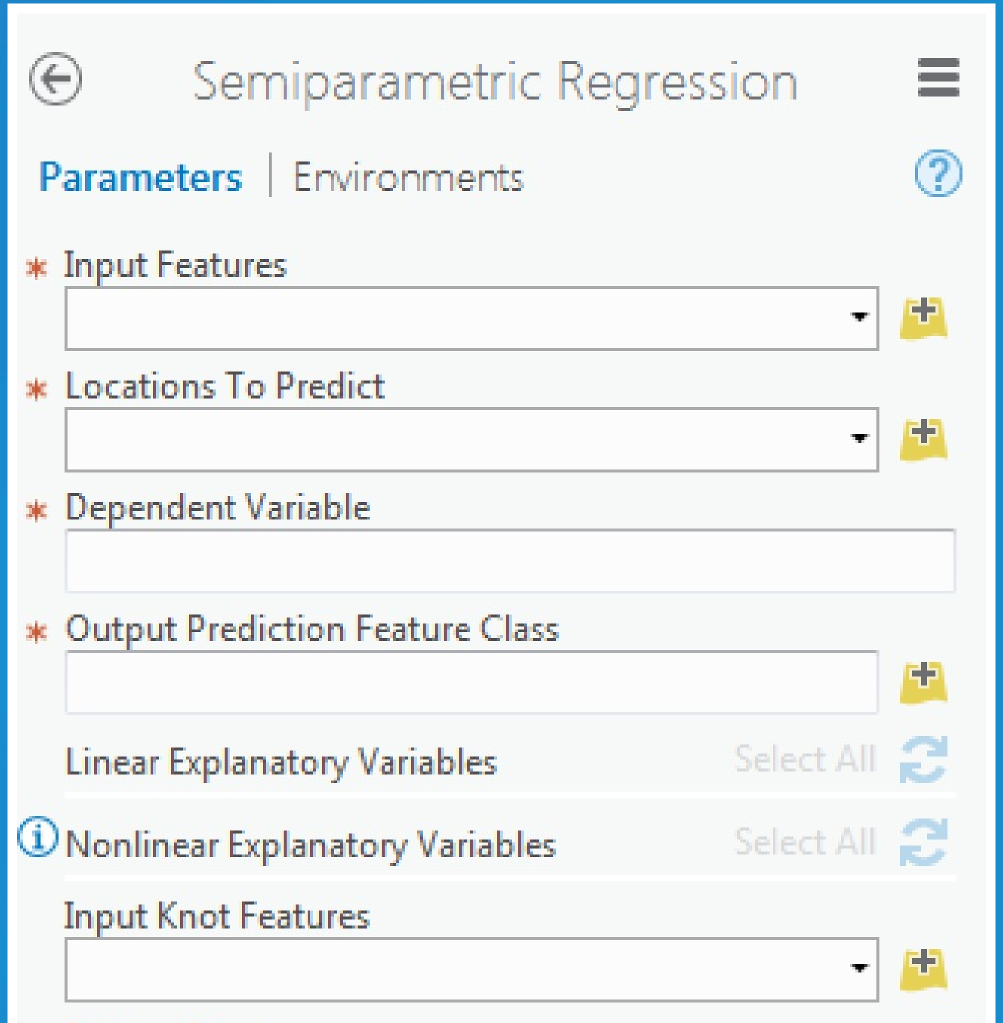
This creates an *ArcGIS data frame* – looks like a data frame, but retains references back to the geometry data.

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)
```

Building R Script Tools



Semiparametric Regression

Parameters | Environments

* Input Features

* Locations To Predict

* Dependent Variable

* Output Prediction Feature Class

Linear Explanatory Variables

Nonlinear Explanatory Variables

Input Knot Features

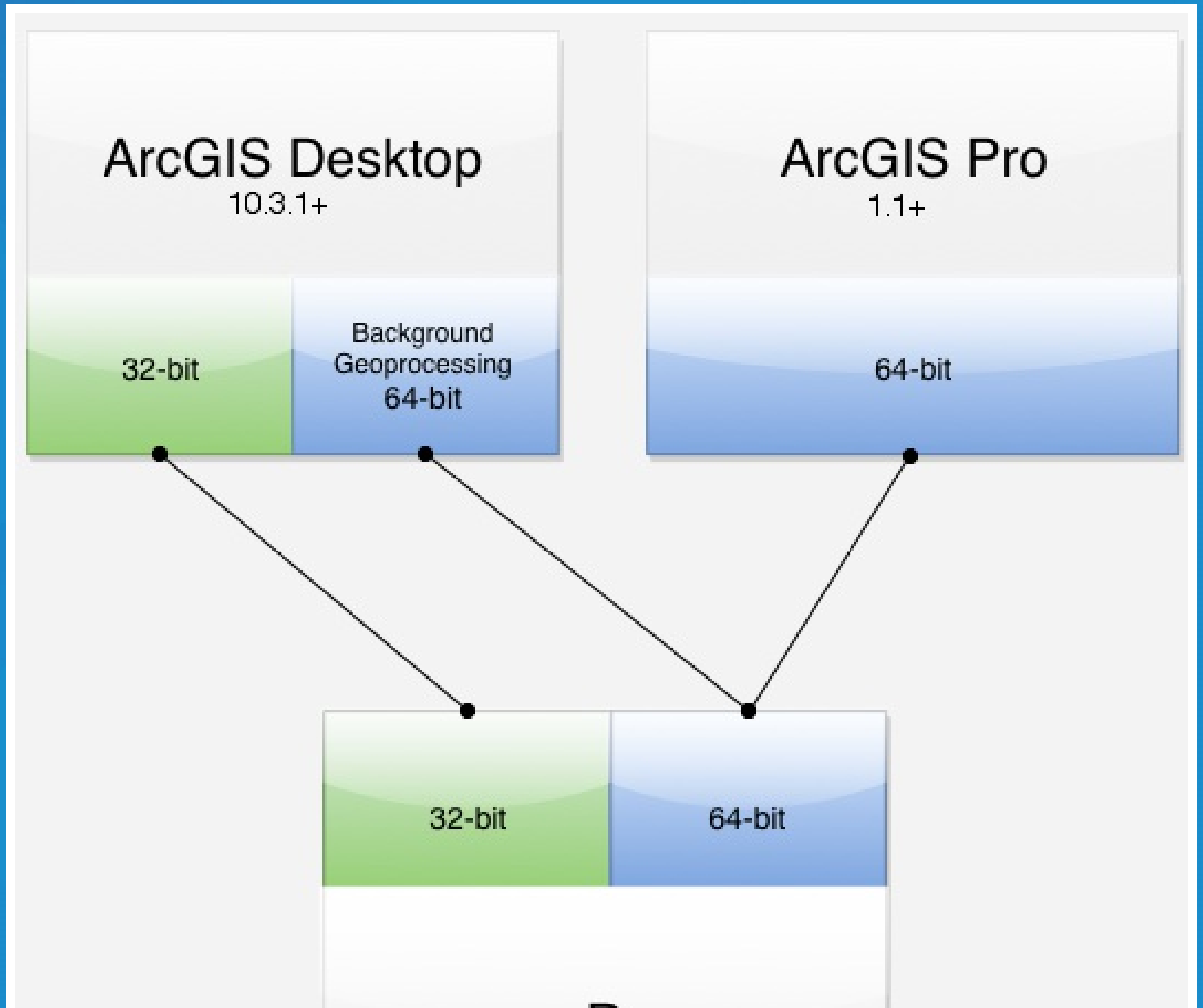
R ArcGIS Bridge Demo

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

How To Install

- Install with the R bridge install
- Detailed installation instructions

Where Can I Run This?



Packages with Conda

Why package management?

- Software is composed of many smaller components, often called *packages* or *libraries*.
- It’s often better to reuse code that solves a problem well rather than recreating it
- But, sharing code is a **hard problem**. Do you have the same packages of the same versions as the developer did?

Why Conda?

CONTINUUM ANALYTICS

- Scientific Python community identified that there was a gap not being addressed by the core Python infrastructure, limiting their ability to get packages into the hands of users
- Industry standard built by people who care about this space — *Continuum Analytics*

Why Conda?

CONTINUUM ANALYTICS

- It solves the hard problem:
  - Handles dependencies for many languages (C, C++, R and of course Python)
  - Built for Python first, but it really solves a much broader infrastructural issue.

Conda

CONDA

- Cross-platform: Simply develop recipes for building and installing software on Linux, OS X and Windows.
- Open source: Esri is using it, you can use it in your own projects for other contexts

What can it install? Not just scientific packages, can install everything from interactive environments like Spyder to Jupyter Notebooks.

CONDA

- *Environments* — Can isolate a Python environment, flexibly make changes withot affecting installed software.
- *Requirements* — include explicit state information, not just the package name.
- Also handles platforms and Jupyter notebooks

Where do packages come from?

Conda packages can come from a variety of locations:

- anaconda.org many thousands of packages
- Repositories (e.g. Anaconda Cloud, self-hosted)
- On disk

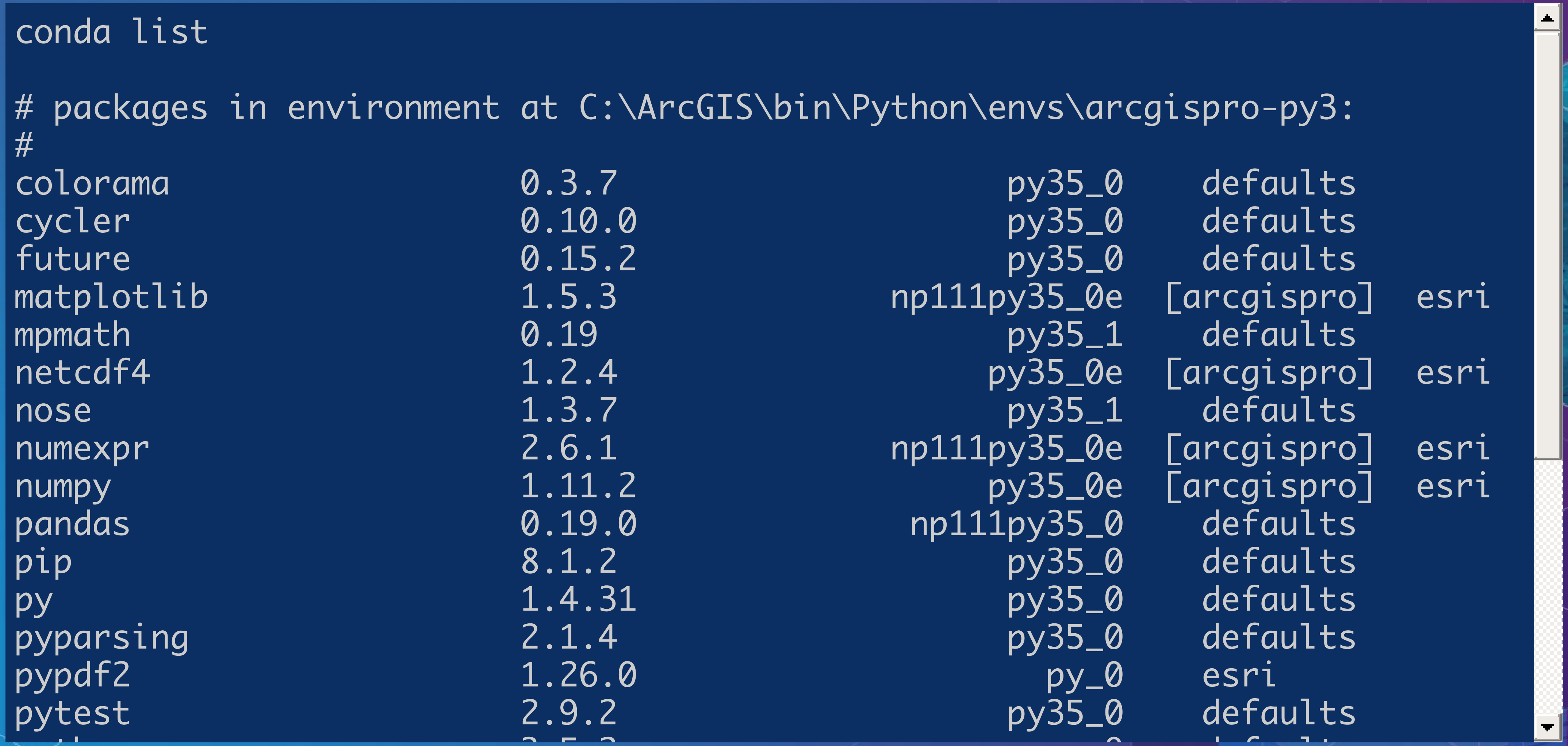
Conda Basics

```
conda --help

conda info
```

Conda info is the starting point — it tells you the state of the environment.

Conda Basics



```
conda list

# packages in environment at C:\ArcGIS\bin\Python\envs\arcgispro-py3:
#
colorama                  0.3.7                    py35_0    defaults
cycler                    0.10.0                   py35_0    defaults
future                    0.15.2                   py35_0    defaults
matplotlib                1.5.3              np111py35_0e  [arcgispro]  esri
mpmath                    0.19                     py35_1    defaults
netcdf4                   1.2.4                   py35_0e  [arcgispro]  esri
nose                      1.3.7                    py35_1    defaults
numexpr                   2.6.1              np111py35_0e  [arcgispro]  esri
numpy                     1.11.2                  py35_0e  [arcgispro]  esri
pandas                    0.19.0              np111py35_0    defaults
pip                       8.1.2                    py35_0    defaults
py                        1.4.31                   py35_0    defaults
pyparsing                 2.1.4                    py35_0    defaults
pypdf2                    1.26.0                     py_0    esri
pytest                    2.9.2                    py35_0    defaults
```

Conda Basics

Activating environments, a couple ways:

- Use the shortcuts
- Manually activate the environment:

```
cd C:\ArcGIS\bin\Python\Scripts
activate arcgispro-py3
```

Conda Basics

- A collection of packages and Python install is called an *environment* or *env*, the building block for managing Python with Conda
- Can have multiple environments and seamlessly switch between them

Conda vs…

Name	Means	Included?
Conda	The command itself	✓
Miniconda	A minimum set of Python packages to build and run Conda.	✓
Anaconda	A distribution 200+ packages built with Conda	
Anaconda Server	Host the full infrastructure internally	

Deeper Dive

How can I use this?

- We already ship you the SciPy stack — powerful and out of the box in all products
- Conda command and a Conda root Python install
- New modules (e.g. `requests`), environment with Pro
- Conda UI

How can I use this?

- We already ship you the SciPy stack — powerful and out of the box in all products
- Conda command and a Conda root Python install
- New modules (e.g. `requests`), environment with Pro
- Conda UI
- Get packages, expand your possibility space
- Package your work: this is an opportunity to distribute it, possibly including commercial side as well.

Can I Run This?

ArcGIS Pro

- ArcGIS Pro 1.3+
  - Is the *the* Python install.
- ArcGIS Pro 1.4
  - UI for Conda

Can I Run This?

- Future:
  - Integration with platform
  - Help make things like MGET even better
  - Share results with organizations and others

Conda Hands-on Activity

Hands-on Exercise