

Esri Developer Summit

March 8–11, 2016 | Palm Springs, CA

Scientific Programming with the SciPy Stack

Shaun Walbridge

Kevin Butler





<https://github.com/scw/scipy-devsummit-2016-talk>

Handout PDF

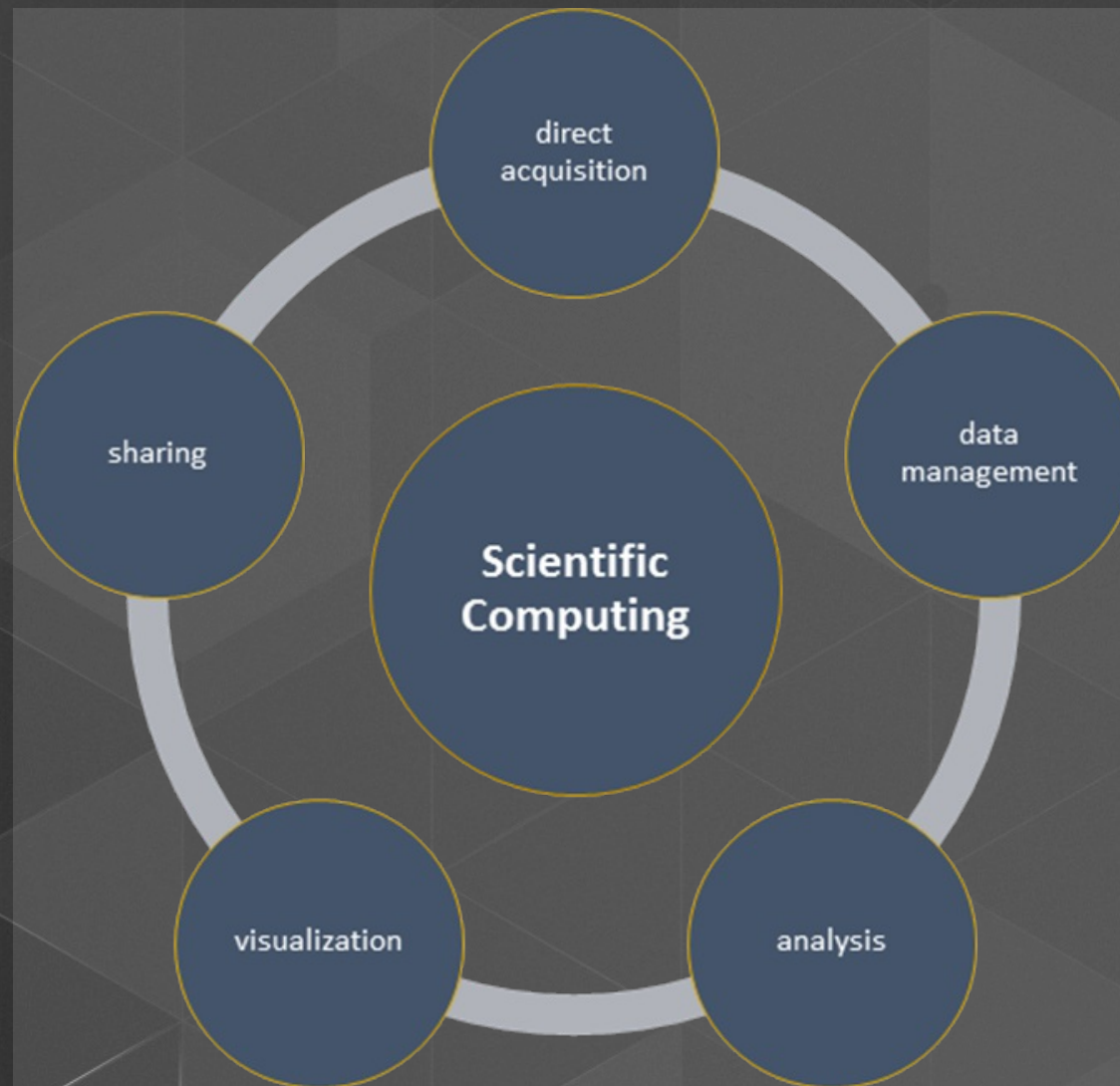
High Quality PDF (28MB)

Resources Section

Scientific Computing

The background features a dark gray grid of hexagons. On the left and right sides, there are clusters of overlapping hexagons in various shades of blue and cyan, creating a 3D effect. The text 'Scientific Computing' is centered in a white, bold, sans-serif font.

Scientific Computing



The application of computational methods to all aspects of the process of scientific investigation – data acquisition, data management, analysis, visualization, and sharing of methods and results.

Python



Why Python?

- Accessible for new-comers, and the [most taught first language in US universities](#)
- Extensive package collection (56k on [PyPI](#)), broad user-base
- Strong glue language used to bind together many environments, both open source and commercial
- Open source with liberal license — do what you want



Why Python?

- Accessible for new-comers, and the [most taught first language in US universities](#)
 - Extensive package collection (56k on [PyPI](#)), broad user-base
 - Strong glue language used to bind together many environments, both open source and commercial
 - Open source with liberal license — do what you want
-
- Brand new to Python? This talk may be challenging
 - Resources include materials that for getting started

Python in ArcGIS

- Python API for driving ArcGIS Desktop and Server
- A fully integrated module: `import arcpy`
- Interactive Window, Python Addins, Python Toolboxes
- Extensions:
 - Spatial Analyst: `arcpy.sa`
 - Map Document: `arcpy.mapping`
 - Network Analyst: `arcpy.na`
 - Geostatistics: `arcpy.ga`
 - Fast cursors: `arcpy.da`

Python in ArcGIS

- Python 3.4 in Pro ([Desktop vs Pro Python](#))
 - `arcpy.mp` instead of `arcpy.mapping`
- Continue to add modules: NetCDF4, xlrd, xlwt, PyPDF2, dateutil, pip
- [Python raster function](#), with a [repository of examples](#) using SciPy for on the fly visualizations

Python in ArcGIS

- Here, focus on SciPy stack, what's included out of the box
- Move toward maintainable, reusable code and beyond the “one-off”
- Recurring theme: multi-dimensional data structures
- Related talks today:
 - [Getting Data Science with R and ArcGIS](#)
 - 2:30PM, this room (Santa Rosa)
 - [Python in ArcGIS Using the Conda Distribution](#)
 - 4:00PM, Mesquite GH

The background features a dark gray grid of thin lines forming a pattern of squares and triangles. In the corners, there are clusters of 3D cubes in various shades of blue and teal, some appearing to be stacked or floating. The word "SciPy" is centered in a white, sans-serif font.

SciPy

Why SciPy?

- Most languages don't support things useful for science, e.g.:
 - Vector primitives
 - Complex numbers
 - Statistics
- Object oriented programming isn't always the right paradigm for analysis applications, but is the only way to go in many modern languages
- SciPy brings the pieces that matter for scientific problems to Python.

Included SciPy

Package	KLOC	Contributors	Stars
matplotlib	118	426	3359
Nose	7	79	912
NumPy	236	405	2683
Pandas	183	407	5834
SciPy	387	375	2150
SymPy	243	427	2672
Totals	1174	1784	

Testing with Nose

- **Nose** — a Python framework for testing
- Tests improve your productivity, and create robust code
- Nose builds on `unittest` framework, extends it to make testing easy.
- Plugin architecture, **includes a number of plugins** and can be extended with **third-party plugins**.



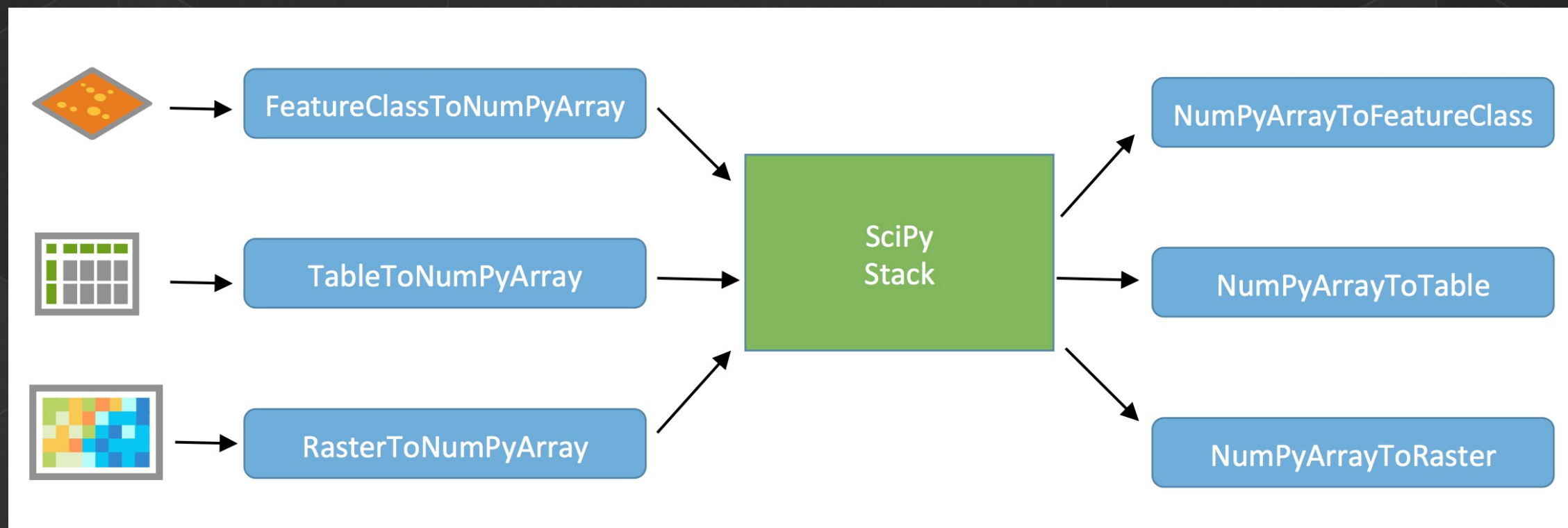
1. An array object of arbitrary homogeneous items
2. Fast mathematical operations over arrays
3. Random Number Generation

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

ArcGIS + NumPy

- ArcGIS and NumPy can interoperate on raster, table, and feature data.
- See [Working with NumPy in ArcGIS](#)
- In-memory data model. Example script to [process by blocks](#) if working with larger data.

ArcGIS + NumPy





Computational methods for:

- Integration ([scipy.integrate](#))
- Optimization ([scipy.optimize](#))
- Interpolation ([scipy.interpolate](#))
- Fourier Transforms ([scipy.fftpack](#))
- Signal Processing ([scipy.signal](#))
- Linear Algebra ([scipy.linalg](#))
- Spatial ([scipy.spatial](#))
- Statistics ([scipy.stats](#))
- Multidimensional image processing ([scipy.ndimage](#))

SciPy: Geometric Mean

- Calculating a geometric mean of an *entire raster* using SciPy ([source](#))

$$\left(\prod_{i=1}^n a_i\right)^{1/n} = \sqrt[n]{a_1 \cdot a_2 \cdots a_n}$$

```
import scipy.stats
rast_in = 'data/input_raster.tif'
rast_as_numpy_array = arcpy.RasterToNumPyArray(rast_in)
raster_geometric_mean = scipy.stats.stats.gmean(
    rast_as_numpy_array, axis=None)
```



Use Case: Benthic Terrain Modeler

Benthic Terrain Modeler

- A Python Add-in and Python toolbox for geomorphology
- Open source, can borrow code for your own projects:
<https://github.com/EsriOceans/btm>
- Active community of users, primarily marine scientists, but also useful for other applications

Lightweight SciPy Integration

- Using `scipy.ndimage` to perform basic multiscale analysis
- Using `scipy.stats` to compute circular statistics

Lightweight SciPy Integration

Example source

```
import arcpy
import scipy.ndimage as nd
from matplotlib import pyplot as plt

ras = "data/input_raster.tif"
r = arcpy.RasterToNumPyArray(ras, "", 200, 200, 0)

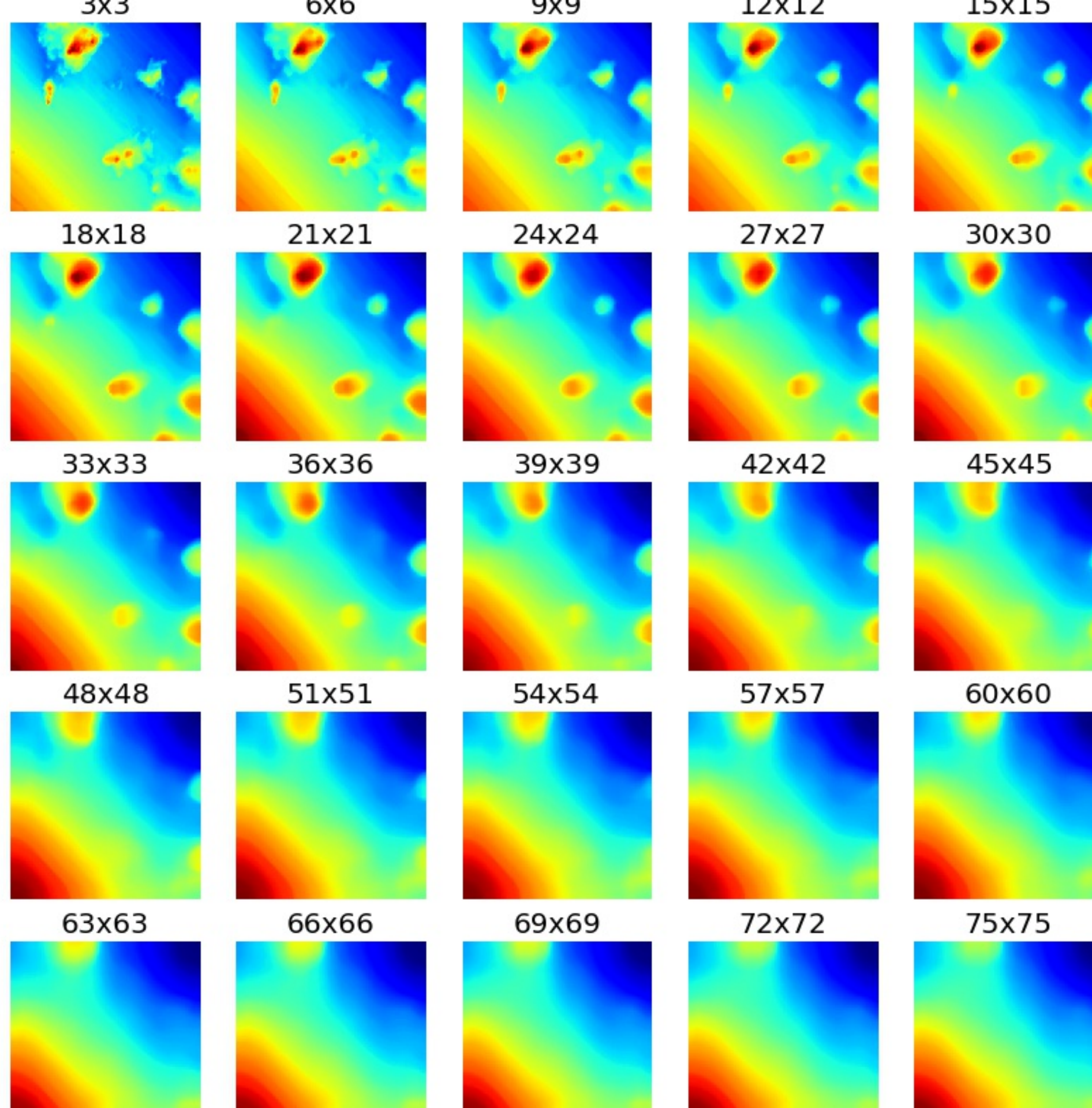
fig = plt.figure(figsize=(10, 10))
```


Lightweight SciPy Integration

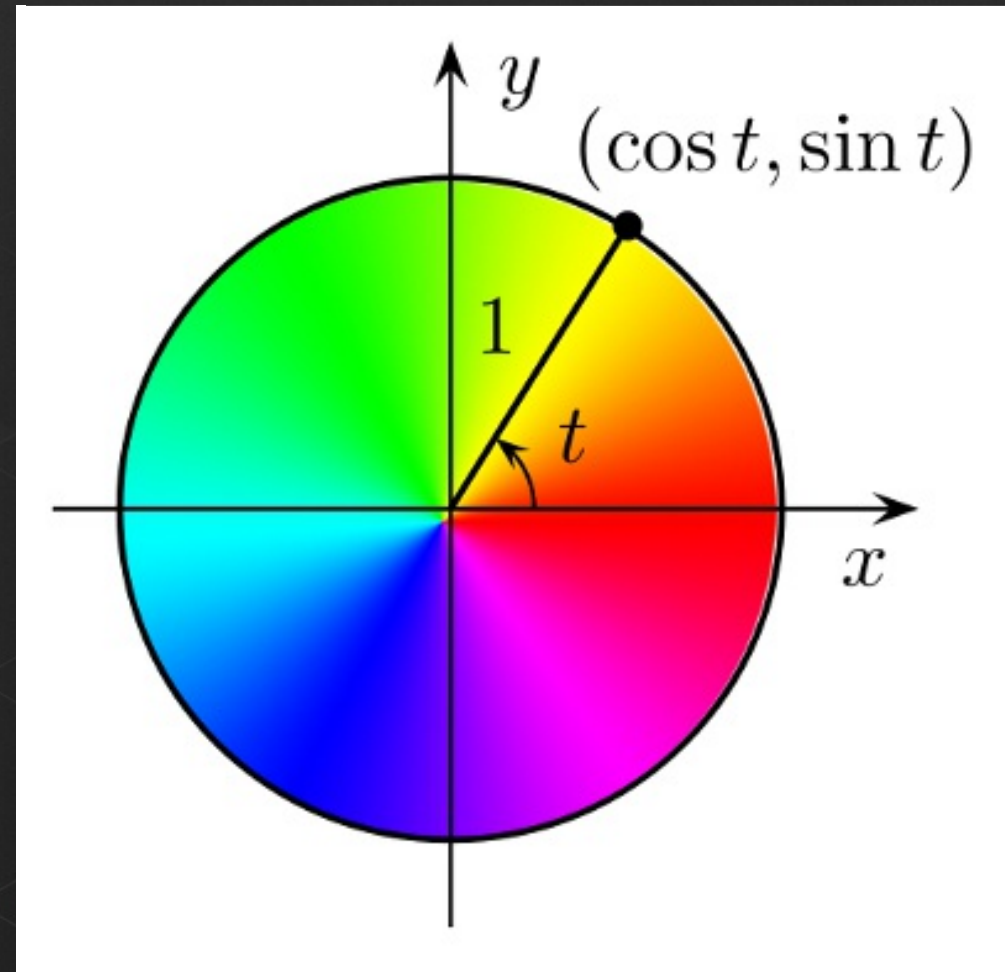
```
for i in xrange(25):
    size = (i+1) * 3
    print "running {}".format(size)
    med = nd.median_filter(r, size)

    a = fig.add_subplot(5, 5, i+1)
    plt.imshow(med, interpolation='nearest')
    a.set_title('{}x{}'.format(size, size))
    plt.axis('off')
    plt.subplots_adjust(hspace = 0.1)
    prev = med

plt.savefig("btm-scale-compare.png", bbox_inches='tight')
```



SciPy Statistics



- Break down aspect into `sin()` and `cos()` variables
- Aspect is a circular variable – without this 0 and 360 are opposites instead of being the same value

SciPy Statistics

Summary statistics from SciPy include circular statistics ([source](#)).

```
import scipy.stats.morestats

ras = "data/aspect_raster.tif"
r = arcpy.RasterToNumPyArray(ras)

morestats.circmean(r)
morestats.circstd(r)
morestats.circvar(r)
```

Demo: SciPy

Multidimensional Data



NetCDF4

- Fast, HDF5 and NetCDF4 read+write support, OPeNDAP
- Hierarchical data structures
- Widely used in meteorology, oceanography, climate communities
- Easier: Multidimensional Toolbox, but can be useful

(Source)

```
import netCDF4
nc = netCDF4.Dataset('test.nc', 'r', format='NETCDF4')
print nc.file_format
# outputs: NETCDF4
nc.close()
```

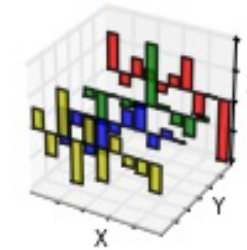
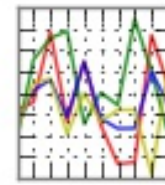
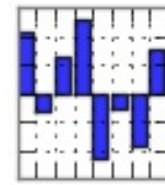
Multidimensional Improvements

- Multidimensional formats: HDF, GRIB, NetCDF
- Access via OPeNDAP, vector renderer, Raster Function Chaining
- *An example which combines mutli-D with time*
- Multi-D supported as WMS, and in Mosaic datasets (10.2.1+)

Pandas

pandas

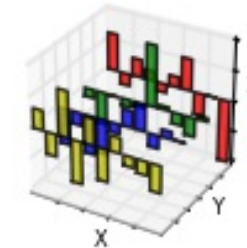
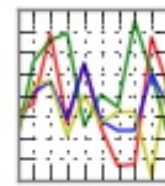
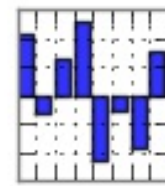
$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



- Panel Data — like R "data frames"
- Bring a robust data *analysis* workflow to Python
- Data frames are fundamental — treat tabular (and multi-dimensional) data as a labeled, indexed series of observations.

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



(Source)

```
import pandas
```

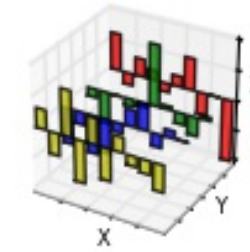
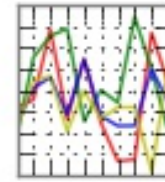
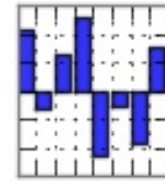
```
data = pandas.read_csv('data/season-ratings.csv')
```

```
data.columns
```

```
Index([u'season', u'households', u'rank', u'tv_households', \
       u'net_indep', u'primetime_pct'], dtype='object')
```

pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



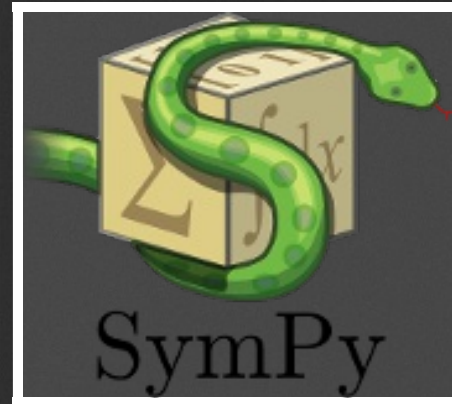
```
majority_simpsons = data[data.primetime_pct > 50]
```

	season	households	tv_households	net_indep	primetime_pct
0	1	13.4m[41]	92.1	51.6	80.751174
1	2	12.2m[n2]	92.1	50.4	78.504673
2	3	12.0m[n3]	92.1	48.4	76.582278
3	4	12.1m[48]	93.1	46.2	72.755906
4	5	10.5m[n4]	93.1	46.5	72.093023
5	6	9.0m[50]	95.4	46.1	71.032357
6	7	8.0m[51]	95.9	46.6	70.713202
7	8	8.6m[52]	97.0	44.2	67.584098
8	9	9.1m[53]	98.0	42.3	64.383562
9	10	7.9m[54]	99.4	39.9	60.916031
10	11	8.2m[55]	100.8	38.1	57.466063
11	12	14.7m[56]	102.2	36.8	53.958944
12	13	12.4m[57]	105.5	35.0	51.094891

Pandas Demo

The background features a dark gray grid of thin lines forming a pattern of squares and diamonds. In the corners, there are clusters of 3D cubes in various shades of blue and teal, some appearing to be stacked or floating. The word "SymPy" is centered in a white, sans-serif font.

SymPy



- A Computer Algebra System (CAS), solve math equations ([source](#))

```
from sympy import *  
x = symbol('x')  
eq = Eq(x**3 + 2*x**2 + 4*x + 8, 0)
```

$$x^3 + 2x^2 + 4x + 8 = 0$$

```
solve(eq, x)
```

$$[-2, -2i, 2i]$$

SymPy Demo

Where and How Fast?

Where Can I Run This?

- Now:
 - ArcGIS Pro (64-bit) [Standalone Python Install for Pro](#)
 - ArcGIS Desktop at 10.4: 32-bit, Background Geoprocessing (64-bit), Server (64-bit), Engine (32-bit)
 - Both now ship with [Scipy Stack](#) (sans IPython)
 - MKL enabled NumPy and SciPy everywhere
 - Older releases: NumPy: ArcGIS 9.2+, matplotlib: ArcGIS 10.1+, SciPy: 10.4+, Pandas: 10.4+
- Upcoming:
 - IPython
 - Conda for managing full Python environments

How Does It perform?

- Built with Intel's [Math Kernel Library \(MKL\)](#) and compilers—highly optimized Fortran and C under the hood.
- Automated parallelization for executed code

[MKL Performance Chart](#)

from future import *



Opening Doors

- Machine learning (`scikit-learn`, `scikit-image`, ...)
- Deep learning (`theano`, ...)
- Bayesian statistics (`PyMC`)
 - Markov Chain Monte Carlo (MCMC)
- Frequentist statistics (`statsmodels`)

Resources



Other Sessions

- Harnessing the Power of Python in ArcGIS Using the Conda Distribution
- Getting Data Science with R and ArcGIS
- Automated Land Surface Temperature Estimation Using Python and ArcGIS Pro
- Writing Image Processing Algorithms using the Python Raster Function
- Python: Working with Scientific Data
- Python: Developing Geoprocessing Tools
- Integrating Open-source Statistical Packages with ArcGIS

New to Python

- Courses:
 - [Programming for Everybody](#)
 - [Codecademy: Python Track](#)
- Books:
 - [Learn Python the Hard Way](#)
 - [How to Think Like a Computer Scientist](#)

GIS Focused

- [Python Scripting for ArcGIS](#)
- [ArcPy and ArcGIS - Geospatial Analysis with Python](#)
- [Python Developers GeoNet Community](#)
- [GIS Stackexchange](#)

Scientific

Courses:

- [Python Scientific Lecture Notes](#)
- [High Performance Scientific Computing](#)
- [Coding the Matrix: Linear Algebra through Computer Science Applications](#)
- [The Data Scientist's Toolbox](#)

Scientific

Books:

- Free:
 - [Probabilistic Programming & Bayesian Methods for Hackers](#)
 - very compelling book on Bayesian methods in Python, uses SciPy + PyMC.
 - [Kalman and Bayesian Filters in Python](#)

Scientific

- Paid:
 - [Coding the Matrix](#)
 - How to use linear algebra and Python to solve amazing problems.
 - [Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython](#)
 - The canonical book on Pandas and analysis.

Packages

Only require SciPy Stack:

- Scikit-learn:
 - [Lecture material](#)
 - Includes SVMs, can use those for image processing among other things...
- FilterPy, Kalman filtering and optimal estimation:
 - [FilterPy on GitHub](#)
- [An extensive list of machine learning packages](#)

Code

- [ArcPy + SciPy on Github](#)
- [raster-functions](#)
 - An open source collection of function chains to show how to do complex things using NumPy + scipy on the fly for visualization purposes
- [statistics library](#) with a handful of descriptive statistics included in Python 3.4.
- *TIP:* Want a codebase that runs in Python 2 and 3? [Check out future](#), which helps maintain a single codebase that supports both. Includes the `futurize` script to initially a project written for one version.

Scientific ArcGIS Extensions

- PySAL ArcGIS Toolbox
- Movement Ecology Tools for ArcGIS (ArcMET)
- Marine Geospatial Ecology Tools (MGET)
 - Combines Python, R, and MATLAB to solve a wide variety of problems
- SDMToolbox
 - species distribution & maximum entropy models
- Benthic Terrain Modeler
- Geospatial Modeling Environment
- CircuitScape

Conferences

- [PyCon](#)
 - The largest gathering of Pythonistas in the world
- [SciPy](#)
 - A meeting of Scientific Python users from all walks
- [GeoPython](#)
 - The Python event for Python and Geo enthusiasts
- [PyVideo](#)
 - Talks from Python conferences around the world available freely online.
 - [PyVideo GIS talks](#)

Closing

The background features a dark, textured surface with a faint grid of thin, light-colored lines. In the lower-left quadrant, there is a cluster of overlapping triangles in various shades of blue and teal, creating a geometric pattern. A small, solid teal circle is positioned on the grid lines near the top-left of this triangular cluster.

Thanks

- Geoprocessing Team
- The many amazing contributors to the projects demonstrated here.
 - Get involved! All are on GitHub and happily accept contributions.

Rate This Session

iOS, Android: Feedback from within the app



Rate This Session

iOS, Android: Feedback from within the app

Windows Phone, or no smartphone? Cuneiform tablets accepted.

