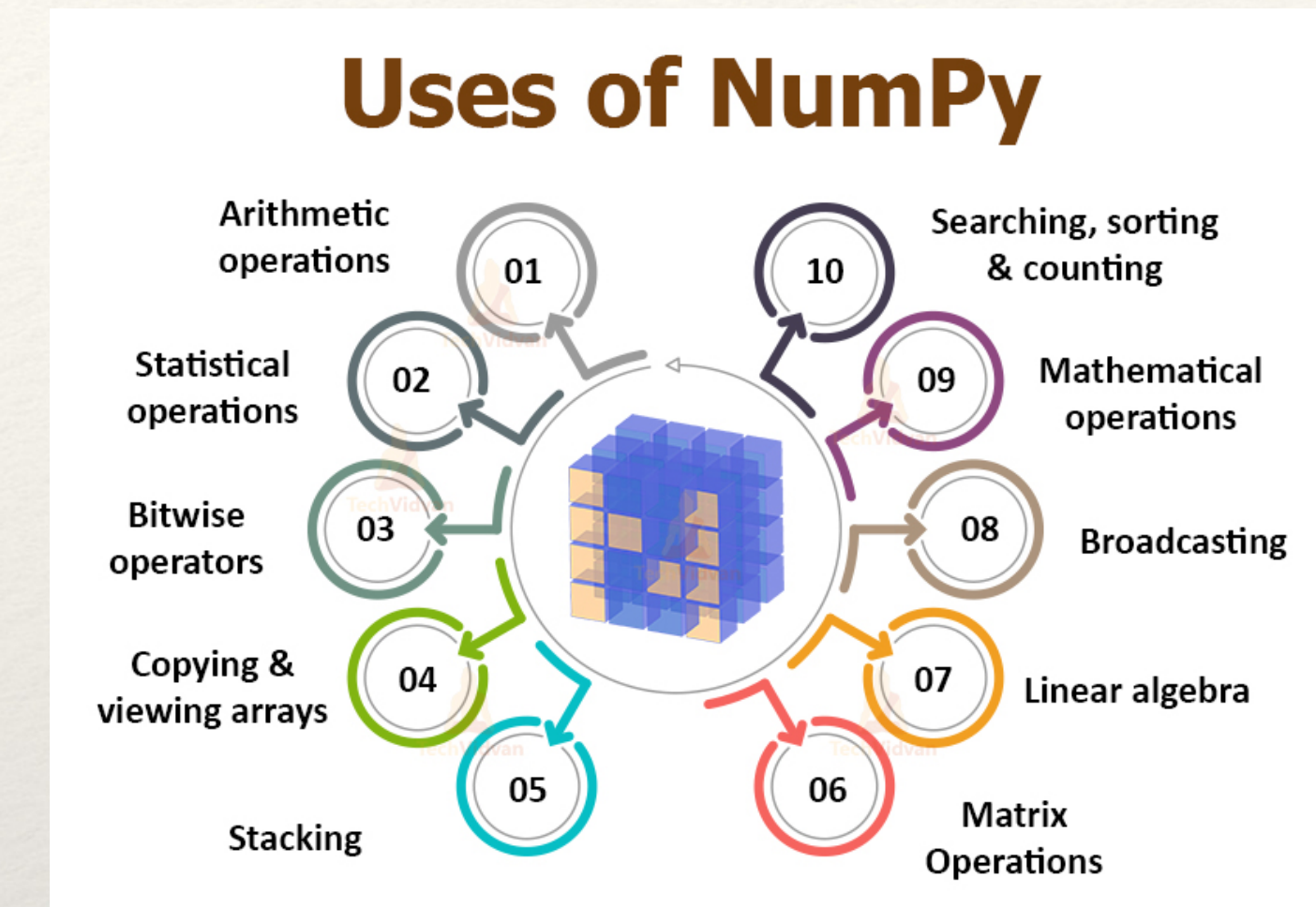


What is numpy?

- ❖ A library for python
- ❖ Optimized for CPU usage so it can run on large data-sets
- ❖ Introduces n-dimensional arrays
- ❖ has functions for working in domain of linear algebra, fourier transform, and matrices.
- ❖ IS REQUIRED to use SCIPY (next slide)



Installing numpy and scipy + running from command shell

miniconda = a python environment

(start the miniconda powershell)

You can now run:

conda install numpy

conda install scipy

SUGGESTION: make a folder under C:\ called pythonprograms to store your programs AND data!

cd pythonprograms

Now you can run your programs with

python programname.py

[important: you cannot use idle with miniconda...you can just type python from shell to get an interactive window]

```
Anaconda Powershell Prompt (Miniconda3)
conda-4.11.0      | 14.4 MB | ##### | 100%
ca-certificates-2021 | 115 KB | ##### | 100%
intel-openmp-2021.4. | 2.2 MB | ##### | 100%
mkl-service-2.4.0   | 51 KB  | ##### | 100%
blas-1.0           | 6 KB   | ##### | 100%
mkl_fft-1.3.1       | 139 KB | ##### | 100%
openssl-1.1.1l      | 4.8 MB | ##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
(base) PS C:\> conda install scipy
Collecting package metadata (current_repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\hbeverly\Miniconda3

added / updated specs:
- scipy

The following packages will be downloaded:

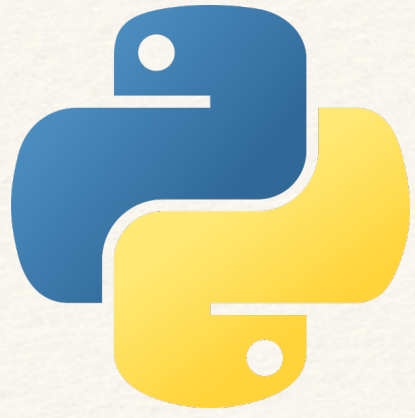
package | build | size
-----|-----|-----
icc_rt-2019.0.0 | h0cc432a_1 | 6.0 MB
scipy-1.7.1 | py39hbe87c03_2 | 13.8 MB
-----|-----|-----
Total: | | 19.8 MB
```


Running numpy (simple examples)

```
Anaconda Powershell Prompt (Miniconda3)

(base) PS C:\> python
Python 3.9.5 (default, May 18 2021, 14:42:02) [MSO
Type "help", "copyright", "credits" or "license"
>>> import numpy as np
>>> arr = np.array([123,55,66,99,34,55,66,99])
>>> np.sort(arr)
array([ 34,  55,  55,  66,  66,  99,  99, 123])
>>> arr
array([123,  55,  66,  99,  34,  55,  66,  99])
>>> onehundred_below = (arr < 100)
>>> arr[onehundred_below]
array([55, 66, 99, 34, 55, 66, 99])
>>> arr.size
8
>>> arr.ndim
1
>>>
```

```
>>>
>>>
>>>
>>>
>>> arr.max()
123
>>> arr.min()
34
>>> arr.sum()
597
>>> arr*2
array([246, 110, 132, 198,  68, 110, 132, 198])
>>> arr/2
array([61.5, 27.5, 33. , 49.5, 17. , 27.5, 33. , 49.5])
>>> np.unique(arr)
array([ 34,  55,  66,  99, 123])
>>> arr
array([123,  55,  66,  99,  34,  55,  66,  99])
>>>
```

Let's Code!

Use numpy to multiply two arrays together and find the minimum product.

```
arr1 = [1,3,5,6,7,19,23,55,777,34325,4346463]
```

```
arr2 = [4354,2342,645,34,4624,234,536,45,3,2,1]
```

Min product?

What are the two values in the product [hint: use index]

https://numpy.org/doc/stable/user/absolute_beginners.html#indexing-and-slicing

What is scipy?

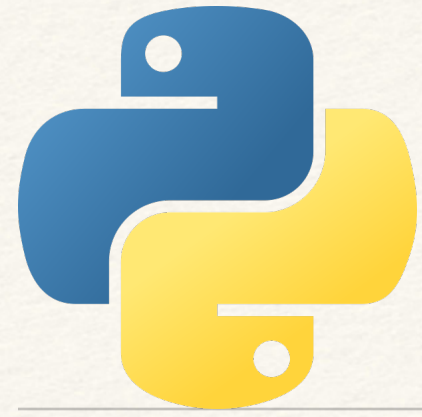
- ❖ A library for python
- ❖ Optimized for CPU usage so it can run on large data-sets
- ❖ Built on numpy
- ❖ Other things (including pandas) built on it.
- ❖ Performs scientific computations (sin, cos, interpolation, stats of all kinds, etc.)



Simple Example: Interpolation using scipy

Anaconda Powershell Prompt (Miniconda3)

```
(base) PS C:\> python
Python 3.9.5 (default, May 18 2021, 14:42:02) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy
>>> from scipy import interpolate
>>> x = numpy.array([1,5,9,14,20,24,30])
>>> y = numpy.array([3,15,17,39,59,71,91])
>>> f = interpolate.interp1d(x, y, axis=0, fill_value="extrapolate")
>>> f(0)
array(0.)
>>> f(1)
array(3.)
>>> f(10000)
array(33324.33333333)
>>>
```

Let's Code!

Find the root(s) of several equations

Equation 1: $y = 2x + 3$

Equation 2: $y = 10x + \cos(2x)$

[hint: from math import cos]

[hint: use root function]

https://www.w3schools.com/python/scipy/scipy_optimizers.php

<https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.root.html>

Basic Website Statistics

Discuss: what is the best metrics for success?

Visits = number of total visits in total, several persons could visit more than once

Unique Visits = only count a visit once per unique visitor.

Pageviews = number of total pages viewed by all visits

Pages/Visit = pageviews / visits

Avg. Unique Pageviews = average number of pages viewed per unique visitor

New Visit = a visit from a new visitor never seen before on the site.

Click-outs = number of clicks leading out of the site

Sales = number of sales that happened on the site.

Conversion Rate = % of visits that lead to a conversion event (such as a sale or click-out, etc.)



2021.07.02_Daily-Activity-Report file

- ❖ Data from 2 real websites (anonymized as etsy and shopify)
- ❖ Goal of these sites is to get user to “Click Out Internal or External” which is considered like a purchase... registration is another goal
- ❖ daily_New_RU = newly registered users on that date.
- ❖ daily_RAU = number of registered users who uniquely visited on that date
- ❖ $\text{daily_Registered_User_Visits}$ = number of visits by registered users on that date
- ❖ daily_PV = total number of pageviews
- ❖ daily_Internal_CO = number of internal click-outs [i.e. a click-out to an internal sale]
- ❖ daily_External_CO = number of external click-outs [i.e. click-out to an external sale]

Quiz

- ❖ Closed notes quiz on Python so far (from weeks 1 + 2).