

Printout

Wednesday, August 28, 2024 9:36 AM

Section 1

MANE 3351

Subsection 1

Lecture 2

Classroom Management

Agenda

- Questions
- Review 1st day
- Introduction to Python
- Discuss lab today
- Call roll

Subsection 2

Resources

Handouts

- [Lecture 2 Slides](#)
- [Lecture 2 Marked Slides](#)

Assignments

- Create free GitHub account

Git

Linux



The screenshot shows the Git website homepage. At the top left is the Git logo (a red diamond with a white 'G') followed by the text 'git --fast-version-control'. To the right is a search bar with the placeholder text 'Search entire site...'. Below the header, there are two paragraphs of text. The first paragraph states: 'Git is a **free and open source** distributed version control system designed to handle everything from small to very large projects with speed and efficiency.' The second paragraph states: 'Git is **easy to learn** and has a **tiny footprint with lightning fast performance**. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like **cheap local branching**, convenient **staging areas**, and **multiple workflows**.' To the right of the text is a diagram illustrating Git's distributed nature. It shows several stacks of papers, each representing a repository. These stacks are connected by colored lines (red, blue, yellow) that represent the branching and merging of code. The diagram shows a central stack with branches leading to other stacks, illustrating how multiple developers can work on different parts of a project simultaneously and then merge their changes back together.

Figure 1: git

Source

GitHub

GitHub

Article [Talk](#)

From Wikipedia, the free encyclopedia

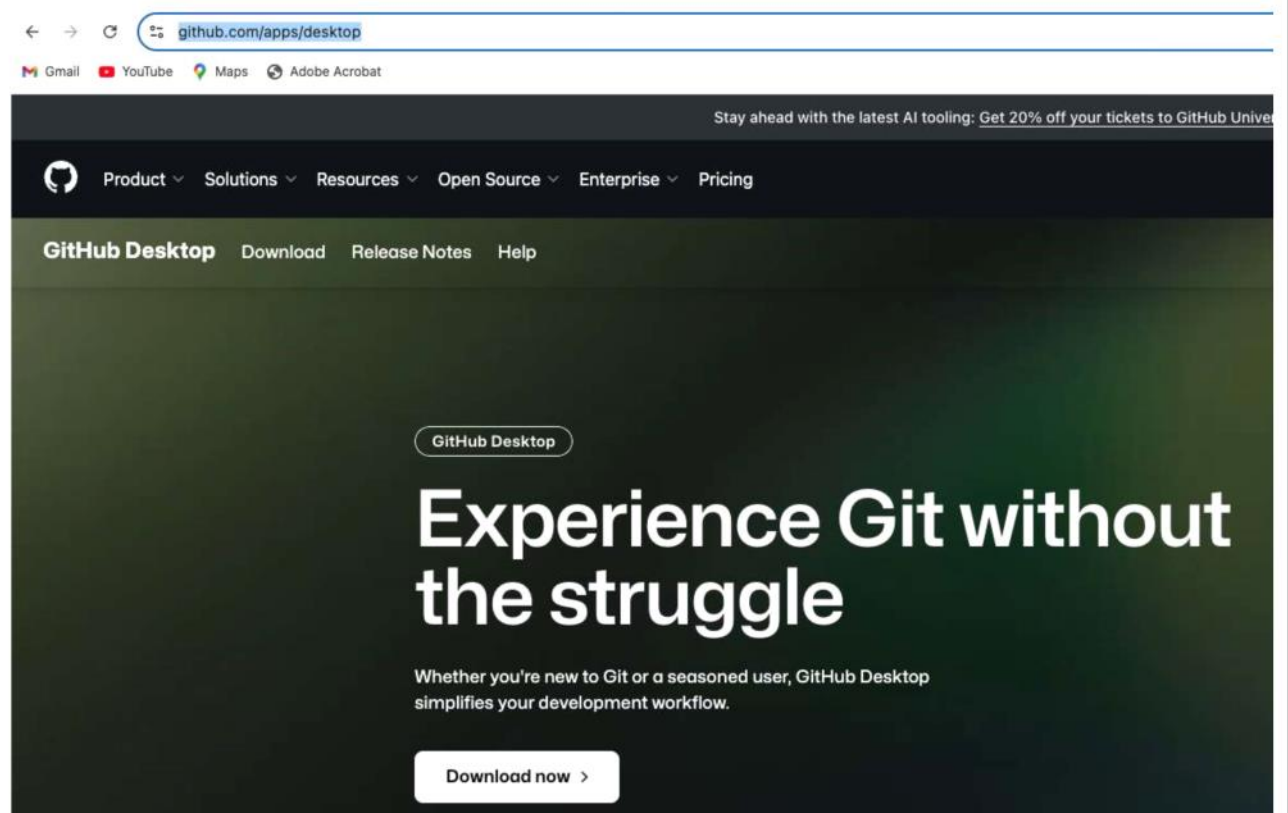
Not to be confused with [Git](#) or [GitLab](#).

free

GitHub (/ˈɡɪthʌb/) is a developer platform that allows developers to create, store, manage and share their code. It uses [Git](#) software, providing the [distributed version control](#) of Git plus [access control](#), [bug tracking](#), [software feature requests](#), [task management](#), [continuous integration](#), and [wikis](#) for every project.^[6] Headquartered in [California](#), it has been a subsidiary of [Microsoft](#) since 2018.^[7]

It is commonly used to host [open source](#) software development projects.^[8] As of January 2023, GitHub reported having over 100 million developers^[9] and more than 420 million [repositories](#),^[10] including at least 28 million public repositories.^[11] It is the world's largest [source code](#) host as of June 2023.

GitHub Desktop



Python with Jupyter Notebook

- Standard Normal Case 1

```
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as sct
import math

a=0.5

x=np.linspace(-4,4,500)
y=sct.norm.pdf(x,0,1)
y2=0.0*x
maske =(x<a)

plt.plot(x,y,'b')
plt.fill_between(x,y,color='#666666',where=maske)
plt.plot(x,y2,'b')
plt.show()
```

First 4 Lines

- Imports allow external packages to be used
- Most standard packages are included in the Anaconda installation
 - **Matplotlib** “is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers, and four graphical user interface toolkits”
 - **NumPy** “is the fundamental package for scientific computing with Python. It contains among other things: 1). a powerful N-dimensional array object, 2). sophisticated (broadcasting) functions, 3. tools for integrating C/C++ and Fortran code, and 4). useful linear algebra, Fourier transform, and random number capabilities.”
 - **SciPy** “is a Python-based ecosystem of open-source software for mathematics, science, and engineering. In particular, these are some of the core packages: NumPy, SciPy library, Matplotlib, IPython, SymPy, and pandas.”
 - **Math** “provides access to the mathematical functions defined by the C standard.”

Python Libraries

Numpy Linspace



User Guide [API reference](#) Building from source Development Release notes Learn More ▾

numpy.rec.fromrecords
numpy.rec.fromstring
numpy.rec.fromfile
numpy.char.array
numpy.char.asarray
numpy.arange
numpy.linspace
numpy.logspace
numpy.geomspace
numpy.meshgrid
numpy.mgrid
numpy.ogrid
numpy.diag
numpy.diagflat
numpy.tri
numpy.tril
numpy.triu
numpy.vander
numpy.bmat
Array manipulation routines
Bit-wise operations

🏠 > NumPy reference > ... > Array creation routines > numpy.linspace

numpy.linspace

numpy.linspace(start, stop, num=50, endpoint=True, retstep=False, dtype=None, axis=0, *, device=None) [\[source\]](#)

Return evenly spaced numbers over a specified interval.

Returns *num* evenly spaced samples, calculated over the interval *[start, stop]*.

The endpoint of the interval can optionally be excluded.

● **Changed in version 1.16.0:** Non-scalar *start* and *stop* are now supported.

● **Changed in version 1.20.0:** Values are rounded towards `-inf` instead of `0` when an integer `dtype` is specified. The old behavior can still be obtained with `np.linspace(start, stop, num).astype(int)`

Parameters:

start : *array_like*

The starting value of the sequence.

stop : *array_like*

scipy.stats.norm



Installing [User Guide](#) [API reference](#) [Building from source](#) [Development](#) [Release notes](#)

Q Search ⌘ + K

Section Navigation

- scipy
- scipy.cluster
- scipy.constants
- scipy.datasets
- scipy.fft
- scipy.fftpack
- scipy.integrate
- scipy.interpolate
- scipy.io
- scipy.linalg
- scipy.misc
- scipy.ndimage
- scipy.odr
- scipy.optimize
- scipy.signal
- scipy.sparse
- scipy.spatial

🏠 > SciPy API > Statistical functions (`scipy.stats`) > `scipy.stats.norm`

scipy.stats.norm

`norm` = `<scipy.stats._continuous_distns.norm_gen object>` [\[source\]](#)

A normal continuous random variable.

The location (`loc`) keyword specifies the mean. The scale (`scale`) keyword specifies the standard deviation.

As an instance of the `rv_continuous` class, `norm` object inherits from it a collection of generic methods (see below for the full list), and completes them with details specific for this particular distribution.

Methods

<code>rvs(loc=0, scale=1, size=1, random_state=None)</code>	Random variates.
<code>pdf(x, loc=0, scale=1)</code>	Probability density function.
<code>logpdf(x, loc=0, scale=1)</code>	Log of the probability density function.
<code>cdf(x, loc=0, scale=1)</code>	Cumulative distribution function.

Matplotlib

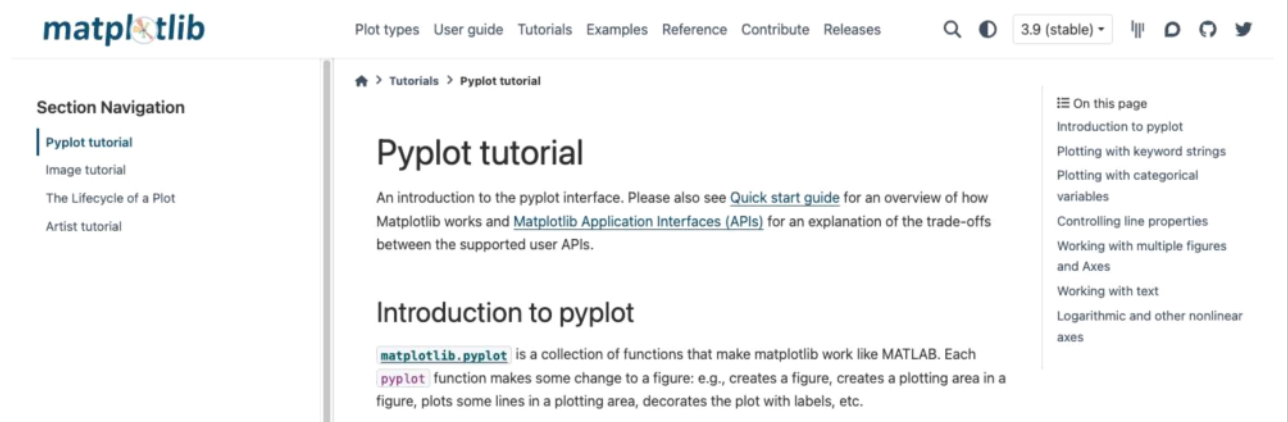


Figure 6: matplotlib

Source