

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



The image is a screenshot of the official Git website. At the top left is the Git logo, which consists of a red diamond shape containing a white icon that looks like a stylized 'b' or a 'g'. To the right of the logo is the word "git" in a large, bold, black sans-serif font. Below the logo, the text "--fast-version-control" is written in a smaller, gray sans-serif font. On the right side of the header is a search bar with a magnifying glass icon and the placeholder text "Search entire site...". The main content area has a light gray background with a subtle grid pattern. On the left, there are two sections of text. The top section is a paragraph about Git's purpose: "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 bottom section is a paragraph about Git's features: "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 3D-style diagram of a network. It shows seven white rectangular blocks, each representing a local repository with a stack of colored files on top. These blocks are interconnected by colored lines: a vertical teal line, a horizontal red line, a diagonal red line, a horizontal teal line, a diagonal yellow line, and a horizontal yellow line. The diagram illustrates the distributed nature of Git's version control system.

Figure 1: git

Source

GitHub

[Article](#) [Talk](#)

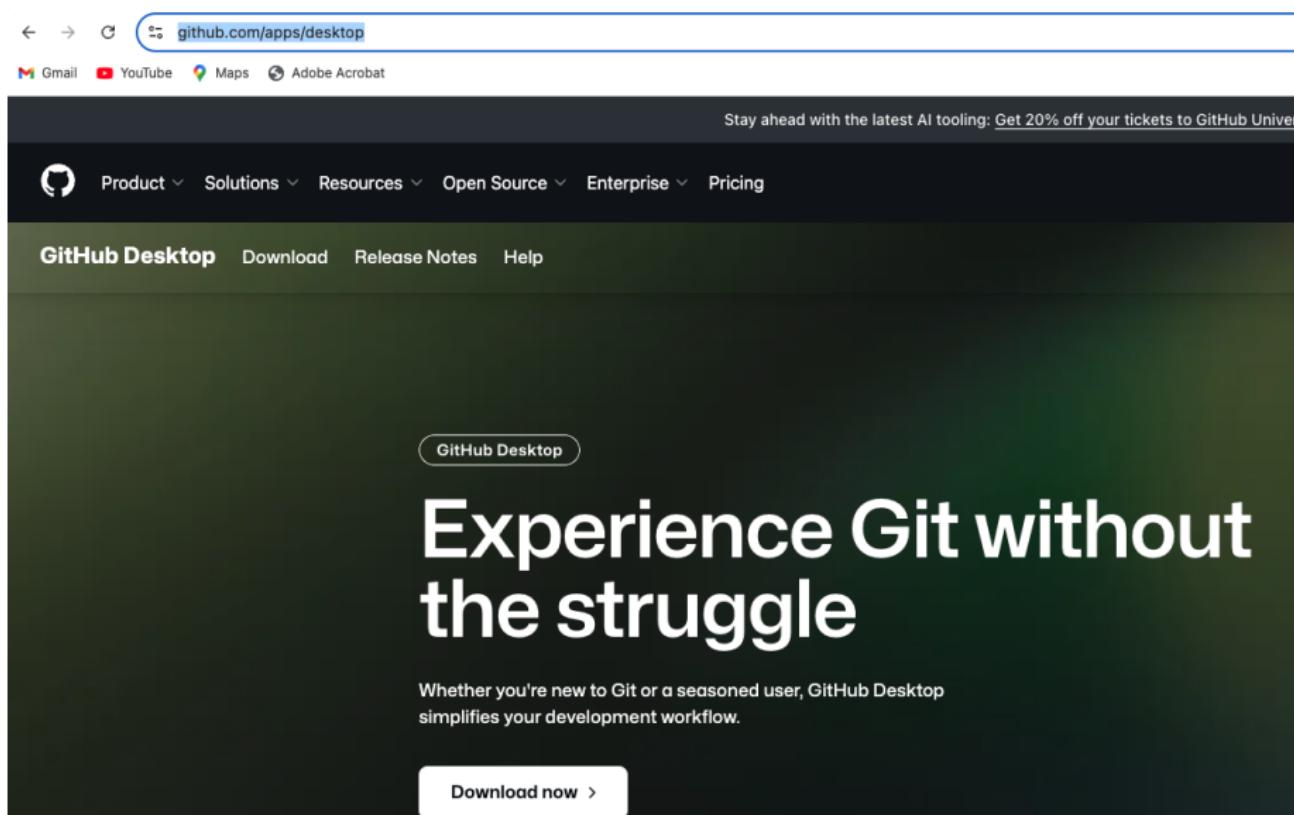
From Wikipedia, the free encyclopedia

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

GitHub ([/githʌb/](#)) 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



The screenshot shows the GitHub Desktop landing page. At the top, there is a navigation bar with links for Product, Solutions, Resources, Open Source, Enterprise, and Pricing. Below this is a secondary navigation bar with links for GitHub Desktop, Download, Release Notes, and Help. The main content area features a large button labeled "GitHub Desktop" and a prominent headline: "Experience Git without the struggle". A subtext below the headline reads: "Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow." At the bottom, there is a call-to-action button labeled "Download now >".

Stay ahead with the latest AI tooling: [Get 20% off your tickets to GitHub Universe](#)

GitHub Desktop Download Release Notes Help

GitHub Desktop

Experience Git without the struggle

Whether you're new to Git or a seasoned user, GitHub Desktop simplifies your development workflow.

Download now >

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

[Home](#) > 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

Search

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.stats

> 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



Plot types User guide Tutorials Examples Reference Contribute Releases



3.9 (stable) ▾



Section Navigation

[Pyplot tutorial](#)

[Image tutorial](#)

[The Lifecycle of a Plot](#)

[Artist tutorial](#)

Home > Tutorials > Pyplot tutorial

Pyplot tutorial

An introduction to the pyplot interface. Please also see [Quick start guide](#) for an overview of how Matplotlib works and [Matplotlib Application Interfaces \(APIs\)](#) for an explanation of the trade-offs between the supported user APIs.

Introduction to pyplot

`matplotlib.pyplot` is a collection of functions that make matplotlib work like MATLAB. Each `pyplot` function makes some change to a figure: e.g., creates a figure, creates a plotting area in figure, plots some lines in a plotting area, decorates the plot with labels, etc.

≡ On this page

[Introduction to pyplot](#)

[Plotting with keyword strings](#)

[Plotting with categorical variables](#)

[Controlling line properties](#)

[Working with multiple figures and Axes](#)

[Working with text](#)

[Logarithmic and other nonlinear axes](#)

Source

Figure 6: matplotlib