

# Data Fundamentals



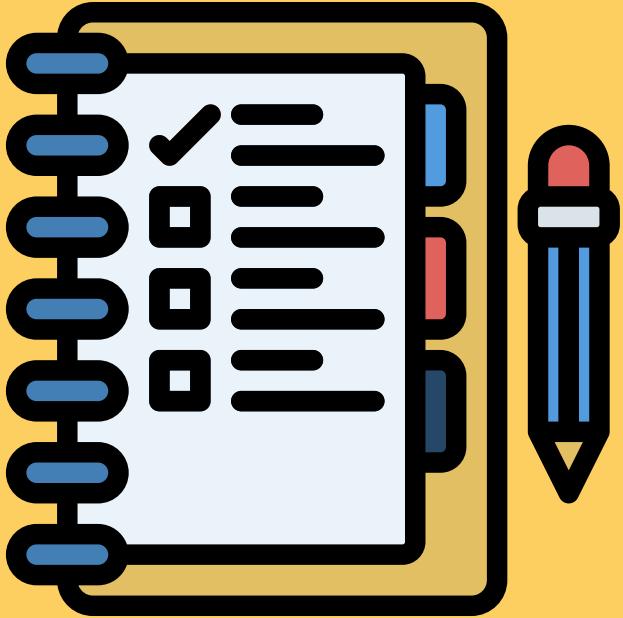
---

Introduction to  
Data Analysis

# GOALS



- Understand the significance and capabilities of Anaconda & Jupyter Notebooks.
- Master the installation process and basic operations with Anaconda.
- Learn how to effectively use Jupyter Notebooks for various applications.
- Engage in a collaborative learning process through the Q&A session.
- Share feedback for continuous improvement.



# AGENDA

- Introduction to Anaconda
- Installing Anaconda
- Key Applications with Anaconda
- Package Management in Anaconda
- Working with Environments
- Best Practices
- Introduction to Jupyter Notebooks
- Installing & Launching Jupyter
- Using Jupyter: Code & Markdown Cells
- Advanced Features in Jupyter
- Applications & Use Cases
- Q&A Session
- Feedback & Evaluation
- Resources

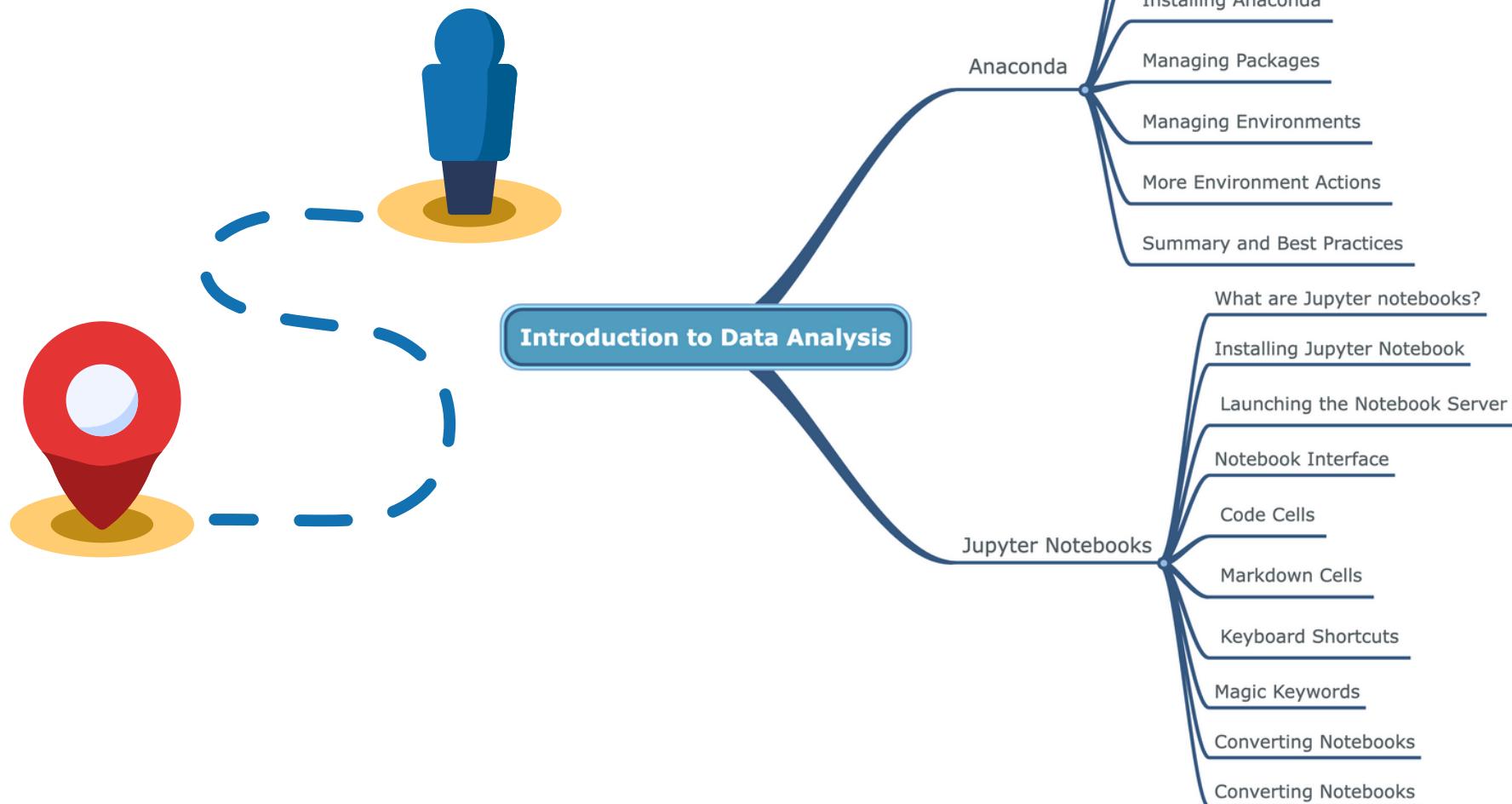


---

Behind every data point, there's a story waiting to be told.

# ROADMAP

---

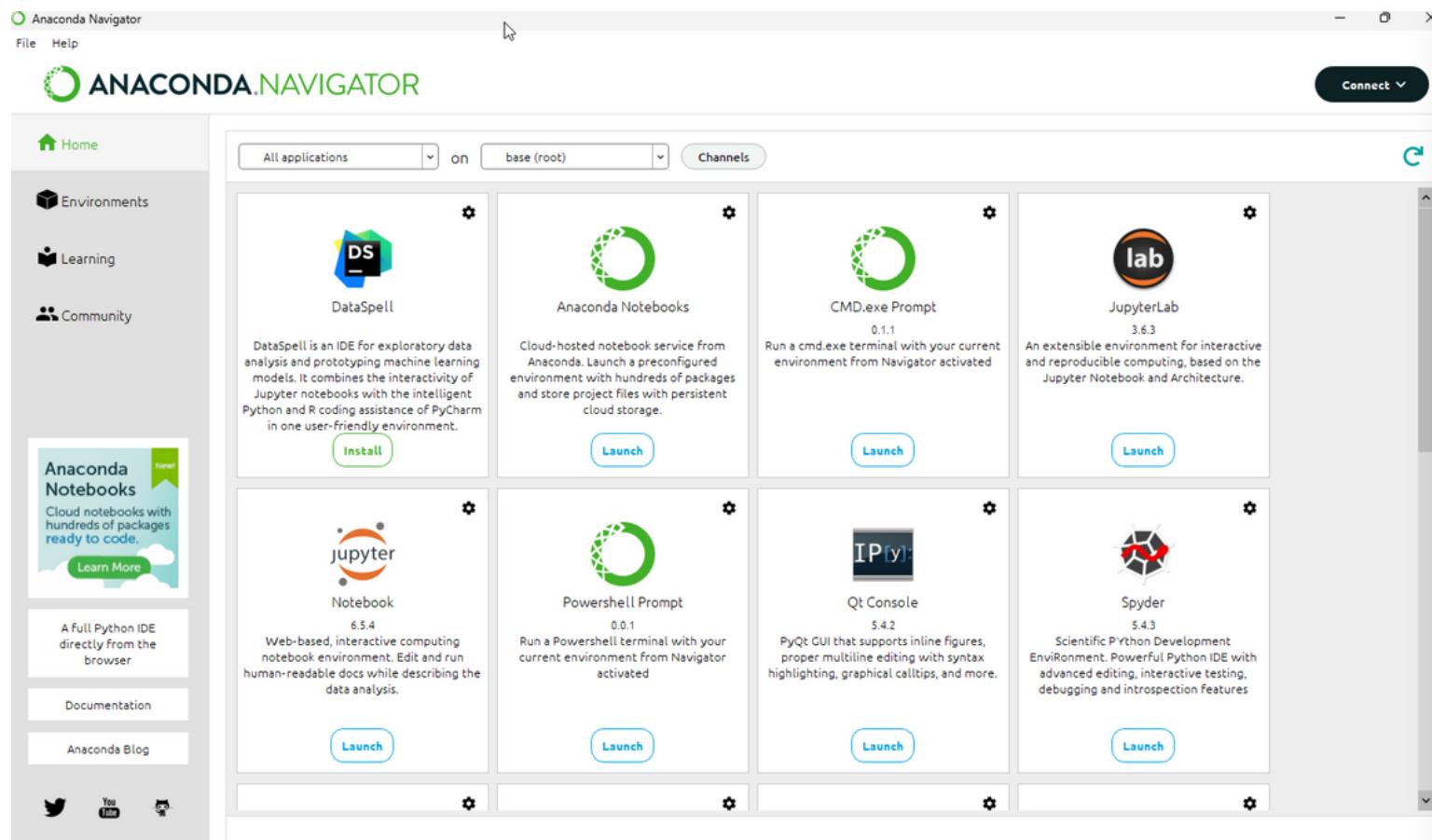


# ANACONDA

---

# INTRODUCTION

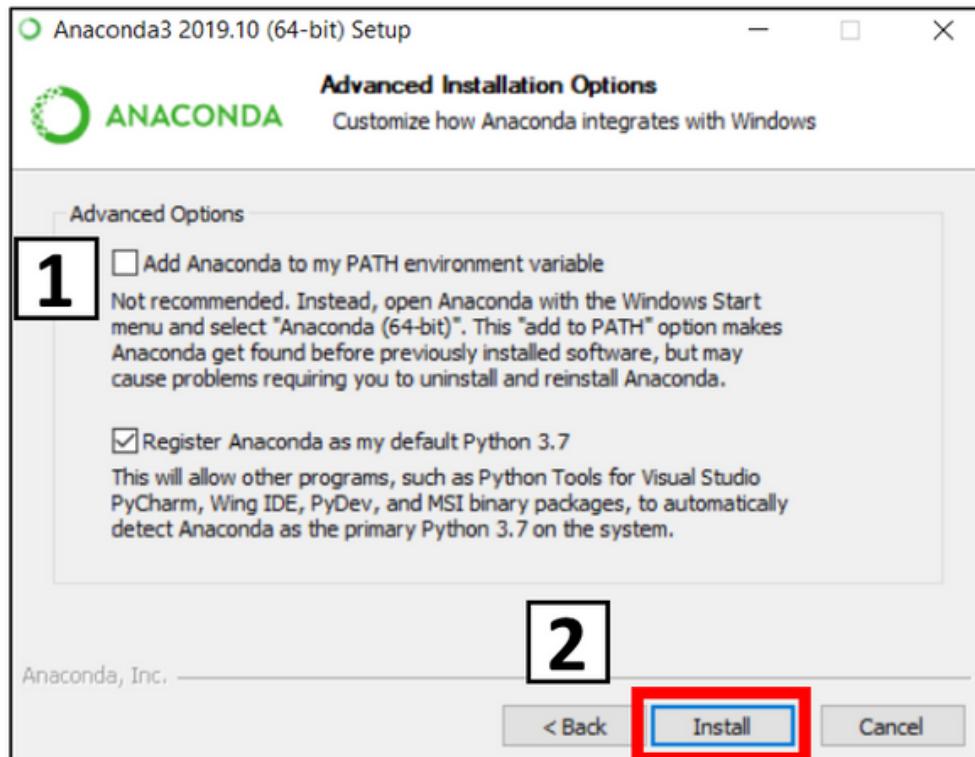
- Comprehensive platform for Python & R.
- Suitable for large-scale data processing, predictive analytics, and scientific computing.
- Seamlessly integrates with other platforms.



# INSTALLING ANACONDA

- Download from: <https://www.anaconda.com/download/>
- Select Python 3.7+ and appropriate system version (64/32-bit).
- Verify installation for your OS.

After installation, use **conda list** to view default packages.



```
Microsoft Windows [Version 10.0.23481.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mahmoud>conda list
# packages in environment at C:\Users\Mahmoud\anaconda3:
#
#          Name           Version      Build  Channel
_anaconda_depends    2023.07      py311_1
abseil-cpp           20211102.0   hd77b12b_0
aiobotocore          2.4.2       py311haa95532_0
aiofiles              22.1.0     py311haa95532_0
aiohttp               3.8.3       py311h2bbff1b_0
aioitertools         0.7.1       pyhd3eb1b0_0
aiosignal             1.2.0       pyhd3eb1b0_0
aiosqlite             0.18.0     py311haa95532_0
alabaster             0.7.12      pyhd3eb1b0_0
anaconda-catalogs    0.2.0       py311haa95532_0
anaconda-client       1.12.0      py311haa95532_0
anaconda-navigator   2.4.2       py311haa95532_0
anaconda-project     0.11.1      py311haa95532_0
anyio                 3.5.0       py311haa95532_0
appdirs               1.4.4       pyhd3eb1b0_0
argon2-cffi          21.3.0      pyhd3eb1b0_0
argon2-cffi-bindings 21.2.0     py311h2bbff1b_0
arrow                 1.2.3       py311haa95532_1
arrow-cpp             11.0.0      py311h308b814_0
astroid                2.14.2      py311haa95532_0
astropy                5.1        py311h5bb9823_0
asttokens              2.0.5       pyhd3eb1b0_0
async-timeout         4.0.2       py311haa95532_0
```

[Installation guide](#)

# KEY APPLICATIONS WITH ANACONDA

- **Anaconda Navigator:** GUI for managing environments & packages.
- **Conda:** Command-line utility.
- **Python:** Latest version is installed.
- **Anaconda Prompt:** For Windows users.
- Includes over 160 scientific packages.

## DEFAULT ANACONDA APPLICATIONS



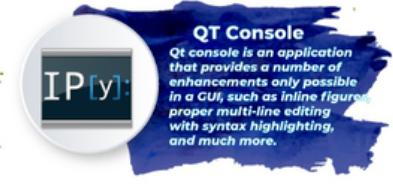
**Jupyter Notebook**

The Jupyter Notebook is an open-source application which runs through a web browser that contain code cells, equations, visualizations, rich text and media.



**VS Code**

Visual Studio Code is a source code editor that includes support for debugging, syntax highlighting, intelligent code completion and code refactoring.



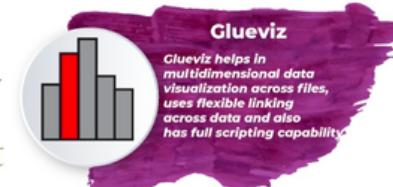
**QT Console**

Qt console is an application that provides a number of enhancements only possible in a GUI, such as inline figures, proper multi-line editing with syntax highlighting, and much more.



**JupyterLab**

JupyterLab is a web-based interactive development environment used for Jupyter notebooks, code, text editors, data & custom components in a flexible, integrated & extensible man-



**Glueviz**

Glueviz helps in multidimensional data visualization across files, uses flexible linking across data and also has full scripting capability.



**R Studio**

R Studio is an IDE that consists of a console, syntax-highlighting editor that supports direct code execution and various tools for plotting, debugging and workspace management.



**Spyder**

Spyder is a powerful scientific IDE designed for scientists, engineers & data analysts which provides real-time code introspection, run configurations for working directory selections, command-line options, console etc.



**Orange 3**

Orange is interactive data visualization toolkit which has a canvas interface in which the user places widgets and creates a data analysis workflow.

# UPDATING PACKAGES

---

Open Terminal/Anaconda Prompt.

## Commands:

- **conda upgrade conda**
- **conda upgrade --all**

```
C:\Users\Mahmoud>conda upgrade conda
Collecting package metadata (current_repodata.json): \ DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): r
repo.anaconda.com:443
DEBUG:urllib3.connectionpool:Starting new HTTPS connection (1): repo.anaconda.com:443
/ DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/noarch/current_repodata.json HTTP/1.1" 304
0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/win-64/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/main/win-64/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/r/noarch/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/msys2/noarch/current_repodata.json HTTP/1.1" 304 0
DEBUG:urllib3.connectionpool:https://repo.anaconda.com:443 "GET /pkgs/msys2/win-64/current_repodata.json HTTP/1.1" 304 0
done
Solving environment: \ |
```

# PIP PACKAGE MANAGER

---

- Pip might be pre-installed with Anaconda.
- Check with **pip --version**.
- If absent, install with **conda install pip**.

# MANAGING PACKAGES

---

- **Installing:** `conda install PACKAGE_NAME`
- **Installing multiple:** `conda install numpy scipy pandas`
- **Updating:** `conda update package_name`
- **Removing:** `conda remove PACKAGE_NAME`
- **Searching:** `conda search *SEARCH_TERM*`

# MANAGING ENVIRONMENTS

---

- **Create:** `conda create -n env_name [python=X.X] [LIST_OF_PACKAGES]`
- **Activate:** `conda activate env_name`
- **Deactivate:** `conda deactivate`

# MORE ON ENVIRONMENTS

---

**Saving & sharing:** `conda env export > environment.yaml`

**Creating from a file:** `conda env create -f environment.yaml`

**Listing environments:** `conda env list`

**Removing an environment:** `conda env remove -n env_name`

# BEST PRACTICES

---

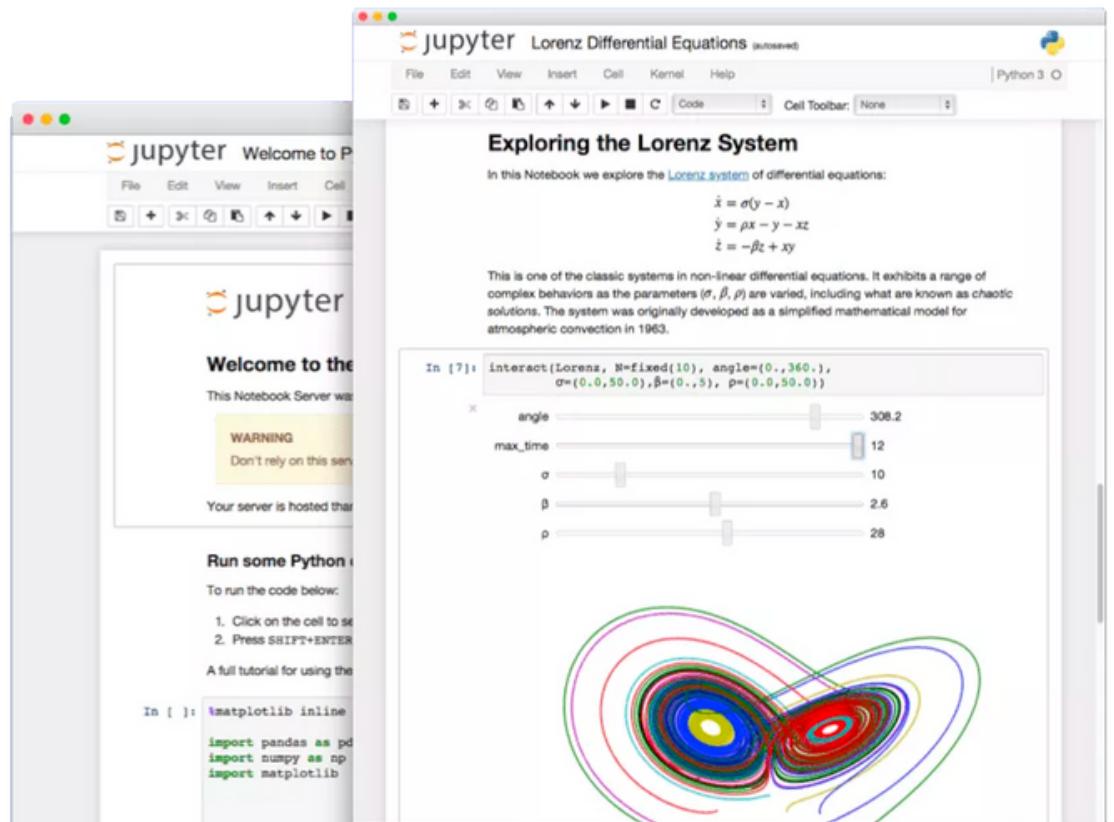
- Create separate environments for different Python versions or projects.
- Share environments for reproducibility.
- Use **pip freeze** for non-conda users.

# JUPYTER NOTEBOOKS

---

# WHAT ARE JUPYTER NOTEBOOKS?

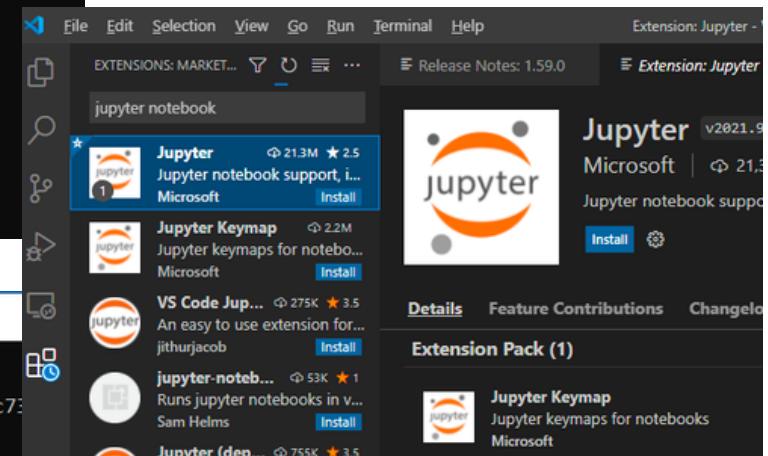
- Interactive computing environment.
- Combines code, markdown, and graphics.
- Supports various languages: Python, R, Julia, etc.
- Ideal for data analysis, visualization, academic research, and more.



# INSTALLATION PROCESS

- Using Python's package manager: **pip install jupyter**
- Using Anaconda for a comprehensive data science toolkit.

```
Command Prompt - jupyter notebook  
operable program or batch file.  
  
C:\Users\virtuo>pip --version  
pip 21.1.3 from C:\Program Files\WindowsApps\PythonSoftwareFoundation.Py  
\site-packages\pip (python 3.9)  
  
C:\Users\virtuo>pip install notebook  
Collecting notebook  
  Using cached notebook-6.4.0-py3-none-any.whl (9.5 kB)  
Collecting nbformat  
  Using cached nbformat-5.1.3-py3-none-any.whl (178 kB)  
Collecting ipykernel  
  Using cached ipykernel-6.0.3-py3-none-any.whl (122 kB)  
Collecting terminado>=0.8.3  
  
Administrator: Command Prompt  
  
C:\Users\rdc>python -m pip install --upgrade pip  
Collecting pip  
  Downloading https://files.pythonhosted.org/packages/30/db/9e38760b32e3e7f40cce46dd5fb107b8c73  
/pip-19.2.3-py2.py3-none-any.whl (1.4MB)  
   100% |██████████| 1.4MB 3.3MB/s  
Installing collected packages: pip  
  Found existing installation: pip 19.0.3  
    Uninstalling pip-19.0.3:  
      Successfully uninstalled pip-19.0.3  
Successfully installed pip-19.2.3  
  
C:\Users\rdc>python -m pip install jupyter  
Collecting jupyter  
  Downloading https://files.pythonhosted.org/packages/83/df/0f5dd132200728a86190397e1ea87cd76244e42d  
/jupyter-1.0.0-py2.py3-none-any.whl  
Collecting jupyter-console (from jupyter)
```



# LAUNCHING JUPYTER

---

- Starting: **jupyter notebook**
- Access through a web browser.
- Interface: Home with a list of notebooks.

```
Microsoft Windows [Version 10.0.23481.1000]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Mahmoud>jupyter notebook

[| | | | - -- -| |-- -| | - ---  
| | | | ' - \ - / - ' | - / - )  
\---/ | .--\---,\---,-|\---\---|  
|_|

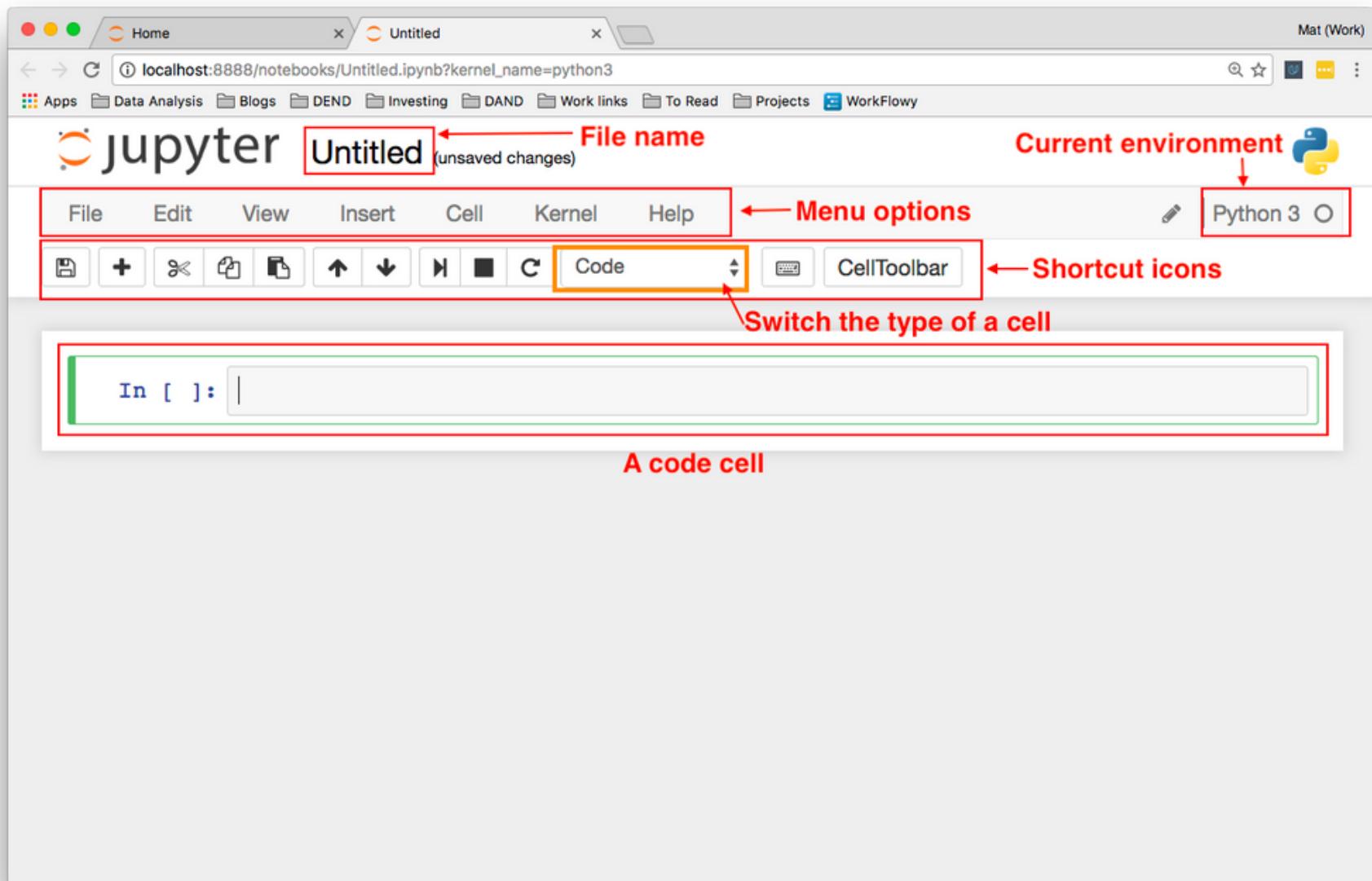
Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions.

https://jupyter-notebook.readthedocs.io/en/latest/migrate\_to\_notebook7.html

Please note that updating to Notebook 7 might break some of your extensions.

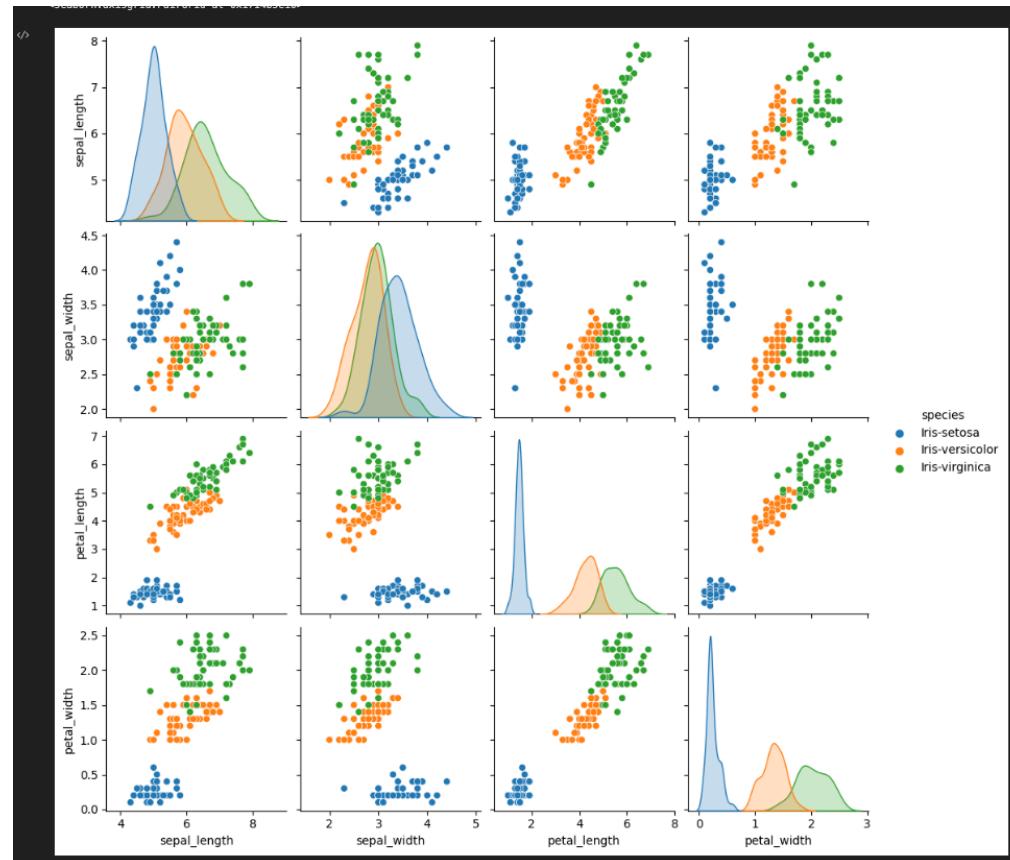
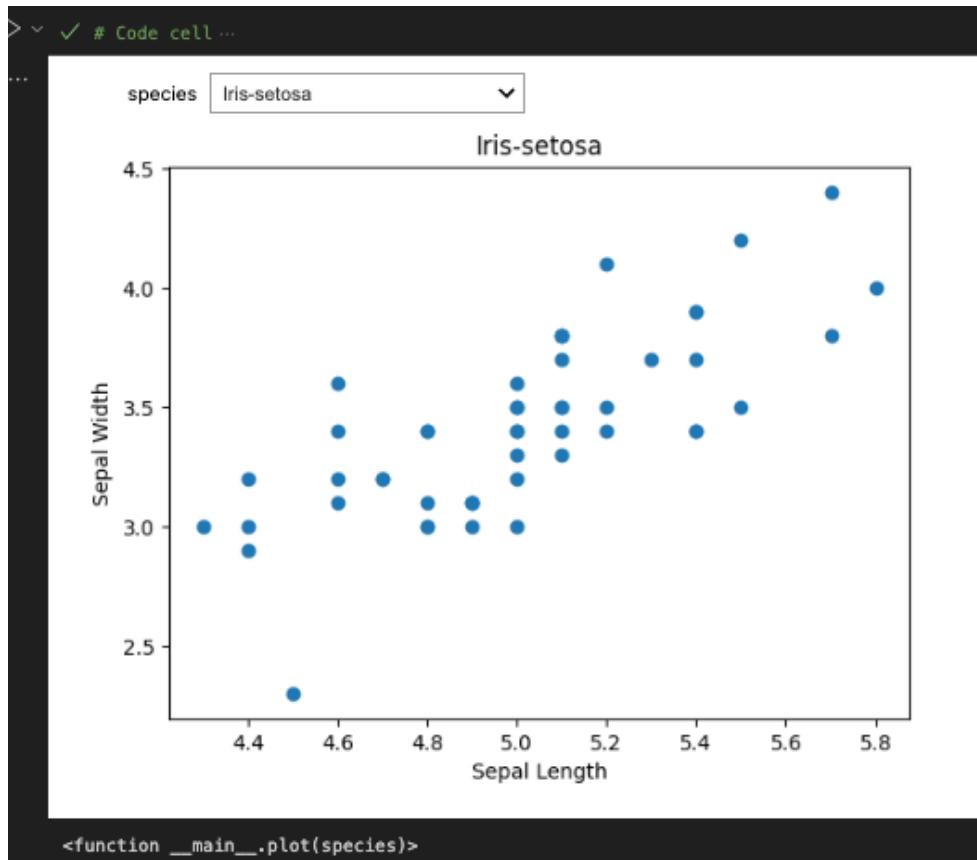
[W 07:40:38.943 NotebookApp] Loading JupyterLab as a classic notebook (v6) extension.
[I 2023-08-11 07:40:38.943 LabApp] JupyterLab extension loaded from C:\Users\Mahmoud\anaconda3\Lib\site-packages\jupyter_lab
[I 2023-08-11 07:40:38.943 LabApp] JupyterLab application directory is C:\Users\Mahmoud\anaconda3\share\jupyter\lab
[I 07:40:41.059 NotebookApp] The port 8888 is already in use, trying another port.
[I 07:40:41.060 NotebookApp] The port 8889 is already in use, trying another port.
[I 07:40:41.060 NotebookApp] The port 8890 is already in use, trying another port.
[I 07:40:41.062 NotebookApp] Serving notebooks from local directory: C:\Users\Mahmoud
[I 07:40:41.062 NotebookApp] Jupyter Notebook 6.5.4 is running at:
[I 07:40:41.062 NotebookApp] http://localhost:8891/?token=197e23c3d500613cdab73dccdac39a52d463ca8c02e051b5
[I 07:40:41.062 NotebookApp] or http://127.0.0.1:8891/?token=197e23c3d500613cdab73dccdac39a52d463ca8c02e051b5
[I 07:40:41.062 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
```

# NOTEBOOK INTERFACE



# CODE CELLS

- Execute live code.
- Supports in-line visualization.
- Displays output below the cell.



# MARKDOWN CELLS

---

- Documenting the process.
- Formatting with headers, links, italics, bold, etc.
- Embedding images and videos.

## Jupyter Notebook Comprehensive Demo

Welcome to a comprehensive demonstration of Jupyter Notebook features in one block.

### Documenting the Process

Every step, idea, or experiment can be detailed and explained using markdown cells, aiding reproducibility and transparency.

For more detailed documentation, we can:

- Make use of *italics* for emphasis
- Make certain words or phrases **bold** for standout
- Hyperlinks: [Visit Udacity!](#)
- Or even `inline code snippets` for small references

### Visual Inclusion

#### Images

While this markdown cannot directly run code, you can always embed static images:



#### Videos

You can also embed videos, for instance, a YouTube video:



### Concluding Remarks

Incorporating all these elements ensures:

1. Improved readability
2. Engaging presentation
3. Comprehensive content delivery

# ADVANCED FEATURES

---

- **Keyboard Shortcuts**: Enhancing productivity.
- **Magic Keywords**: Special commands (%timeit, %matplotlib inline, etc.)
- **Extensions**: Customizing and enhancing functionality.

```
#This will give you information about a single run of your code in terms of CPU times:  
[10] %time x = [i for i in range(1000000)]  
✓ 0.0s  
... CPU times: user 38.7 ms, sys: 3.54 ms, total: 42.2 ms  
Wall time: 12.8 ms  
  
# This is used to time a particular piece of code (a cell magic).  
# It runs the code multiple times to get a more accurate estimate:  
[11] %timeit x = [i for i in range(10000)]  
✓ 9.4s  
... 116 µs ± 1.96 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)  
  
#Prints the current working directory.  
[12] %pwd  
✓ 0.0s  
... '/Users/mahmoudhassan/Desktop/session11'
```

```
#Allows you to render the cell as HTML. So, you can include raw HTML code in it.  
[1] %%html  
    <h2>This is an HTML heading</h2>  
✓ 0.0s  
  
This is an HTML heading  
  
##This renders the cell contents as LaTeX.  
[2] %%latex  
    \begin{align}  
    a = \frac{1}{2} \quad b = \frac{2}{3}  
    \end{align}  
✓ 0.0s  
  
a =  $\frac{1}{2}$    b =  $\frac{2}{3}$ 
```

# APPLICATIONS & USE CASES

---

# DATA CLEANING & TRANSFORMATION

Cleaning a dataset containing sales data and transforming dates into a more usable format.

```
#1. Data Cleaning & Transformation:

import pandas as pd

# Sample data
data = {
    'date': ['2023-01-05', '2023-01-06', '2023-01-07'],
    'sales': [100, 200, 'missing'],
    'product': ['A', 'B', 'A']
}

df = pd.DataFrame(data)

# Replace 'missing' with NaN and convert sales column to float
df['sales'] = pd.to_numeric(df['sales'], errors='coerce')

# Convert date from string to datetime format
df['date'] = pd.to_datetime(df['date'])

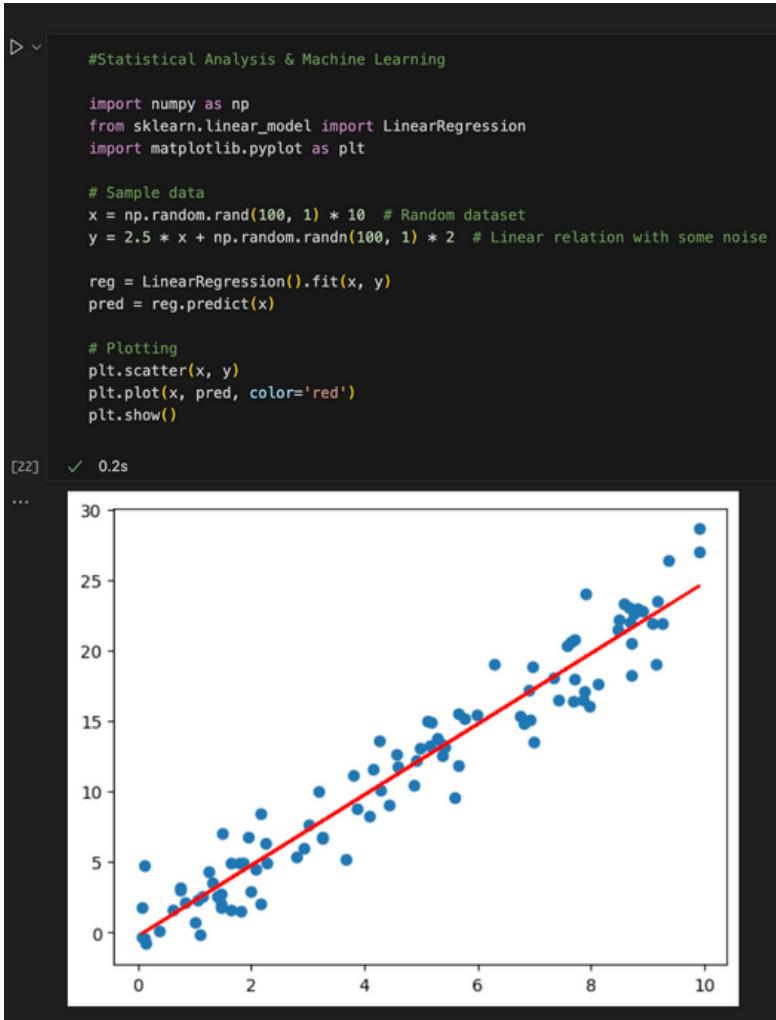
df

[23] ✓ 0.0s
```

	date	sales	product
0	2023-01-05	100.0	A
1	2023-01-06	200.0	B
2	2023-01-07	NaN	A

# STATISTICAL ANALYSIS & MACHINE LEARNING

Perform linear regression on a set of data.



# VISUALIZATION & REPORTING

---

Visualizing sales data using a bar chart.



# ACADEMIC RESEARCH & TEACHING

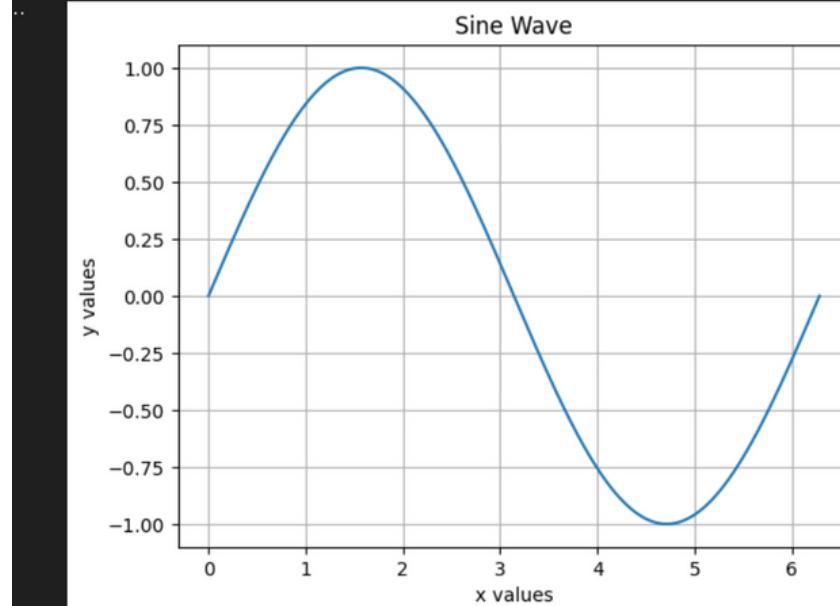
Visualizing sales data using a bar chart.

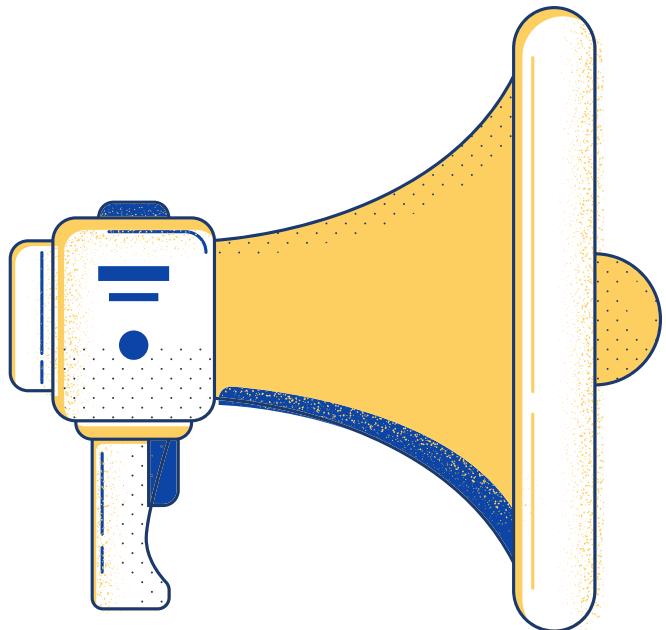
```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 2 * np.pi, 1000)
y = np.sin(x)

plt.plot(x, y)
plt.title("Sine Wave")
plt.xlabel("x values")
plt.ylabel("y values")
plt.grid(True)
plt.show()
```

24]





**Q&A Session:**  
**Let's explore and**  
**understand**  
**together**

# WE NEED YOUR FEEDBACK



Help us improve our sessions! Complete the **Oman Makeen Student Satisfaction Survey**.

Rate my competency, our engagement, and your overall satisfaction.

- Confidential and used solely for session improvement.

**CLICK HERE**

# RESOURCES

---

- [Jupyter Notebooks](#)
- [Conda cheatsheet](#)
- [Anaconda installation](#)
- [IpyWidgets](#)
- [Markdown Cheatsheet](#)
- [Jupyter Keyboard Shortcuts](#)



---

Your presence today has added value  
to our shared learning journey. Thank  
you for joining us!