

Data Mining

Practical Session #1

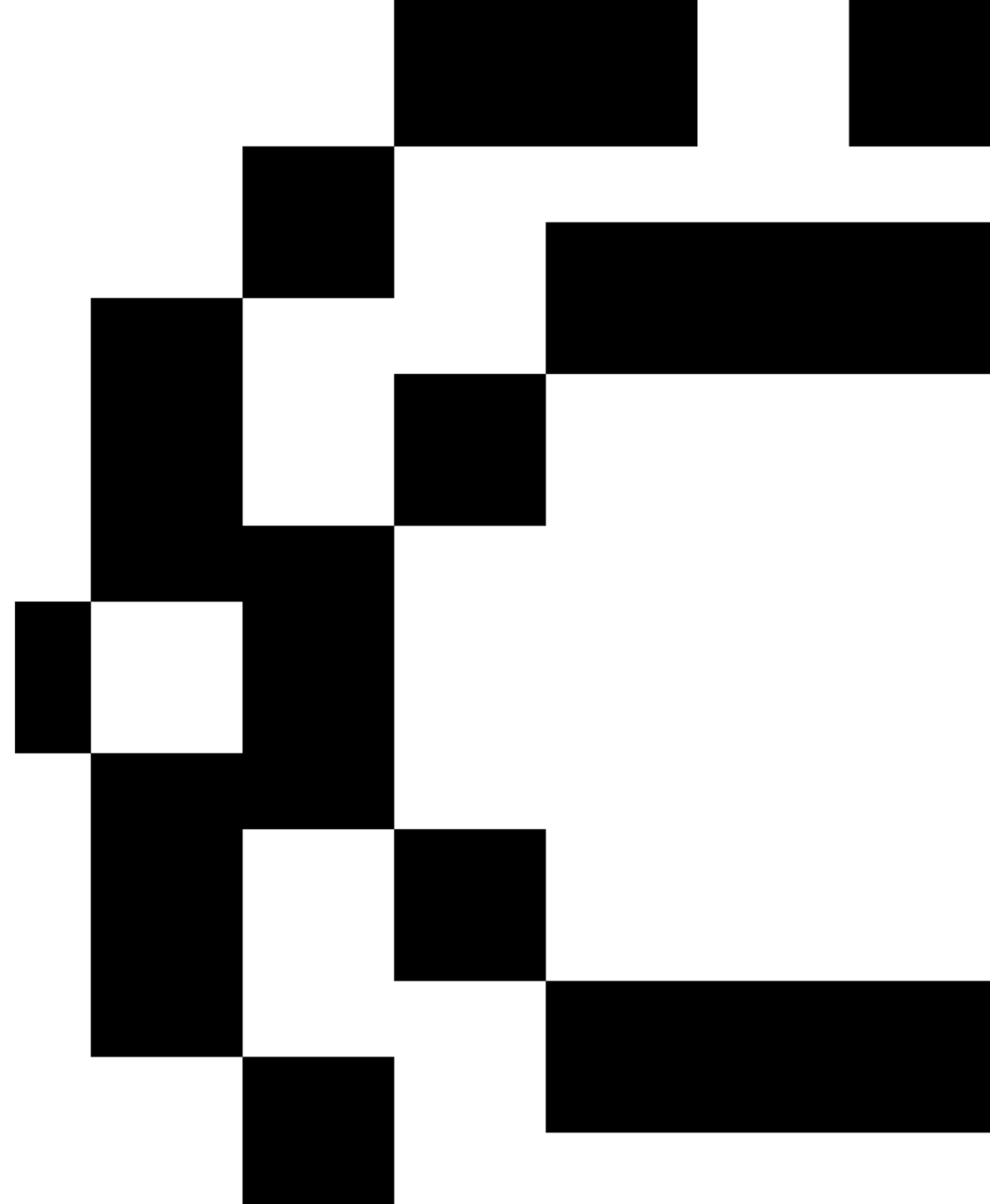
Fall Semester 2025-2026

Master in Data Science and Advanced Analytics

Ana Pedro Caleiro – acaleiro@novaims.unl.pt

Gaspar Pereira – gppereira@novaims.unl.pt

Farina Pontejos - fpontejos@novaims.unl.pt



Please download the lab materials in the
meantime

Practical Session Preparation

Download Anaconda Navigator

- <https://www.anaconda.com/download>
- Documentation: <https://docs.anaconda.com/free/anaconda/install/>

Download notebook and other materials

- Moodle / Practical Sessions / Lab 01
- <https://github.com/fpontejos/Data-Mining-25-26>

Optional: Download GitHub Desktop

- <https://desktop.github.com/>
- Sign up for GitHub account: <https://github.com/signup>

About Us

Ana Pedro

Academic Background:

- BSc in Corporate Finance (2024)
- MSc in Data Science & Advanced Analytics (since 2024 – [Ongoing] – NOVA IMS)

Professional Experience:

- Data Consultant Junior (Ongoing)

Thesis Development:

- Large Language Models; Synthetic Population

About Us

Gaspar Pereira

Academic Background:

- BSc and MSc in Medicine (2012-2018 - NOVA Medical School)
- MSc in Data Science & Advanced Analytics (since 2024 - [Ongoing] – NOVA IMS)

Thesis Development:

- Natural Language Processing; Large Language Models;

Resources

- Bibliography
- Class slides and Jupyter Notebooks
- Data Mining Github repo:
 - <https://github.com/fpontejos/Data-Mining-24-25>
 - <https://github.com/fpontejos/Data-Mining-25-26>
- Google, Stack Overflow, documentations, Github and YouTube

Recommended DataCamp courses

Use the **invite link** (on Moodle) to get 6 months free access to all DataCamp features.

Make sure to **use your university email address** to sign up.

Introduction to Python

<https://app.datacamp.com/learn/courses/intro-to-python-for-data-science>

Intermediate Python

<https://app.datacamp.com/learn/courses/intermediate-python>

Our working environment

- We will be using Anaconda: Currently one of the most popular Python distributions.
- Sets up a data science oriented working environment in Python
- It installs a set of libraries (for now, think of libraries as programming tools – like a toolbox in a woodshop)
- But it can be used for many different purposes (all it takes is installing the necessary libraries)

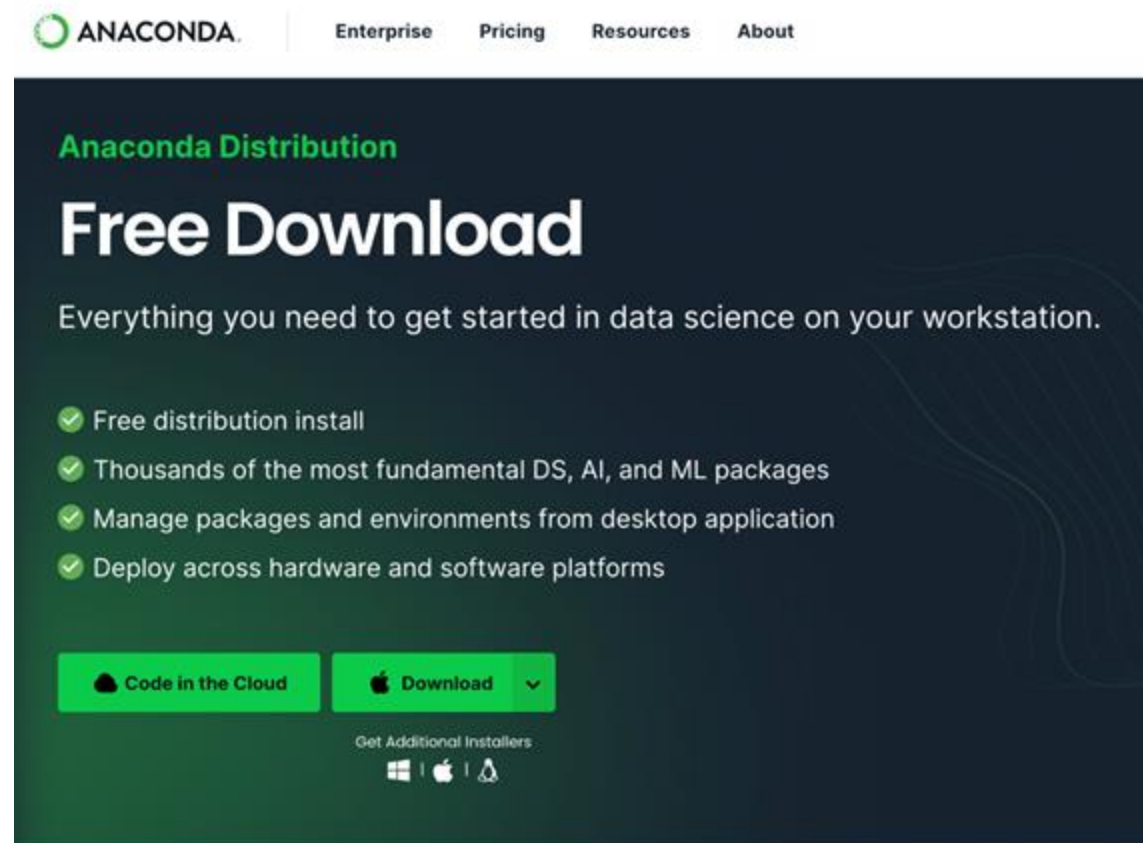
- one of the most popular Python distributions for Data Science
- manages your packages and environments.
- reduce future issues dealing with the various libraries you will be using.
- comes with most of the main libraries for data manipulation
 - Pandas
 - Numpy
 - Matplotlib
 - Scipy
 - ...
- easy to use and install



In the meantime: Download + Install Anaconda Navigator

Download:

<https://www.anaconda.com/download>



Install Anaconda Navigator

Please read the documentation applicable to your system:

<https://docs.anaconda.com/free/anaconda/install/>

Note

When **installing Anaconda**, you have the option to “Add Anaconda to my PATH environment variable.” *This is not recommended* because it *appends* Anaconda to PATH. When the installer appends to PATH, it does not call the activation scripts.

Virtual Environments

<https://docs.conda.io/projects/conda/en/latest/user-guide/concepts/environments.html>

<https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>

- Isolated spaces that contain per-project dependencies (specific collection of installed conda packages)
- Using conda to manage environments:
 - Create, export, list, remove, and update environments
 - Switching or moving between environments (conda activate)
 - You can also share an environment file
- You can also use pip to manage environment

In other words

you may need to use different “versions” of the same types of tools, or entirely different “environments”

Project #1



Cooper (making a barrel)

Project #2



Formwork (setting a structure)

Project #3



Luthier (making a guitar)

In other words

Suppose you will build a Caravel (a Portuguese ship from the 15th century).

However, you are also a Cooper!

To build the Caravel the way they did back then, you will need a specific set of tools, much more rudimentary than the ones you will have at your own workshop.



In other words

- You will need to get them first (i.e., “download” them)
- However, you should not mix these tools with the ones you already have!
 - They are not appropriate to build barrels, and the ones you already have are not appropriate to build Caravels
 - They will create clutter in your workshop (unnecessarily keeping unused packages)
 - You will have duplicate tools with different (version conflicts)
 - Other carpenters may want to build their own replica of you project (reproducibility)
 - If you are working with other carpenters, they will need to use the same types of tools you are using (collaboration)
- These tools (i.e., libraries) and their versions should be specified in the project’s requirements (including the Python version)!

Python Packages



Git and GitHub

<https://guides.github.com/activities/hello-world/>

<https://docs.github.com/en/github/getting-started-with-github>

- What is GitHub?
 - Code hosting platform for version control and collaboration
- What is Git?
 - At the heart of GitHub is an open source version control system (VCS) called Git. Git is responsible for everything GitHub-related that happens locally on your computer.
- Why Git and GitHub?
 - **Optional:** You can use Git and GitHub for collaborating and version control in your projects.
 - Also we have a GitHub repository with all the practical class contents:
 - <https://github.com/fpontejos/Data-Mining-24-25>

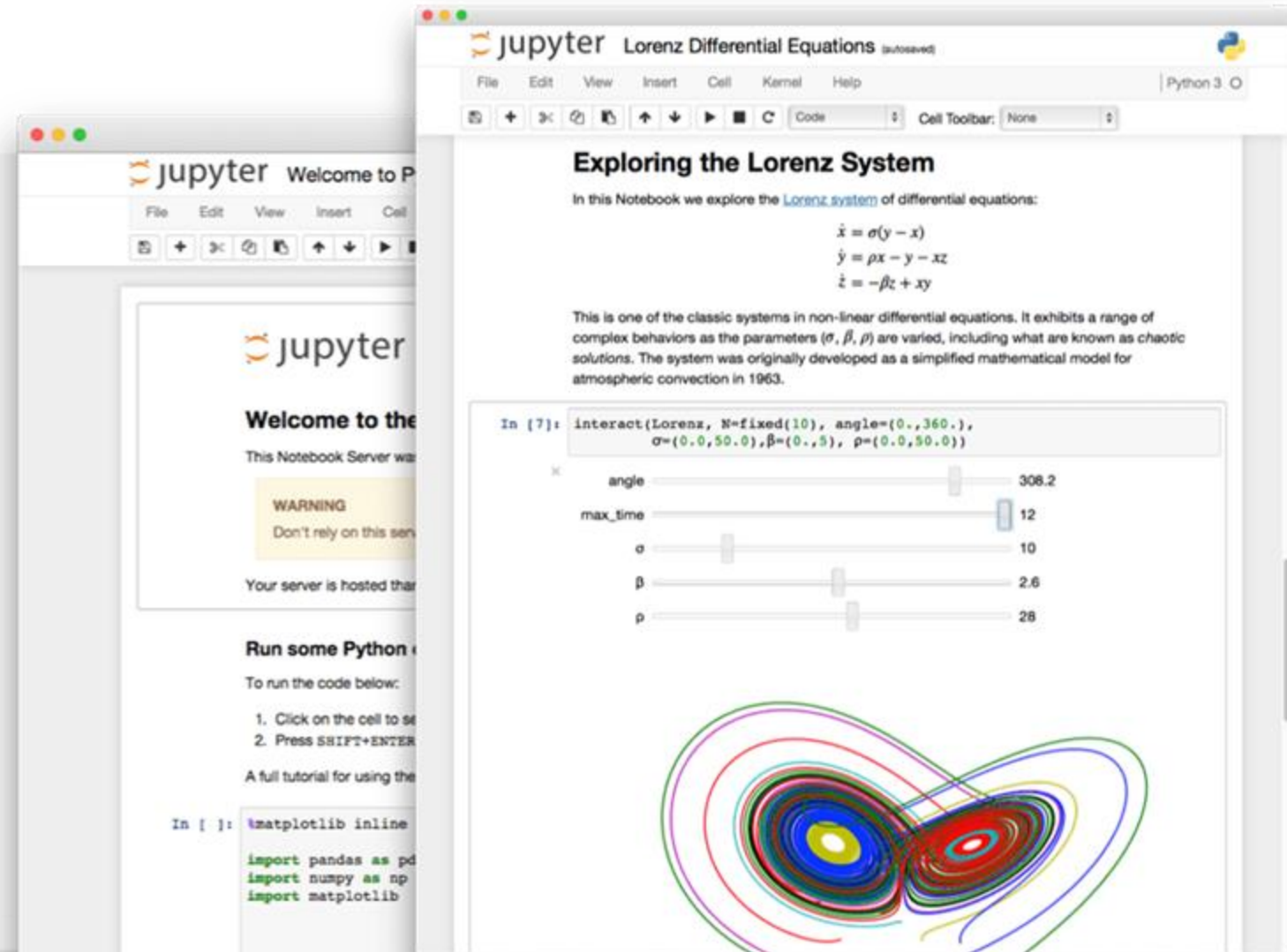
Main ways to access Python

- Python Shell and IPython
 - An interactive environment for writing and running code
- Jupyter Notebooks
 - A notebook that weaves code, data, prose, equations, analysis, and visualization
 - A tool for prototyping new code and analysis
 - A method for creating a reproducible workflow for scientific research
- IDE (Integrated Development Environment):
 - Software that helps you build code

Jupyter notebooks

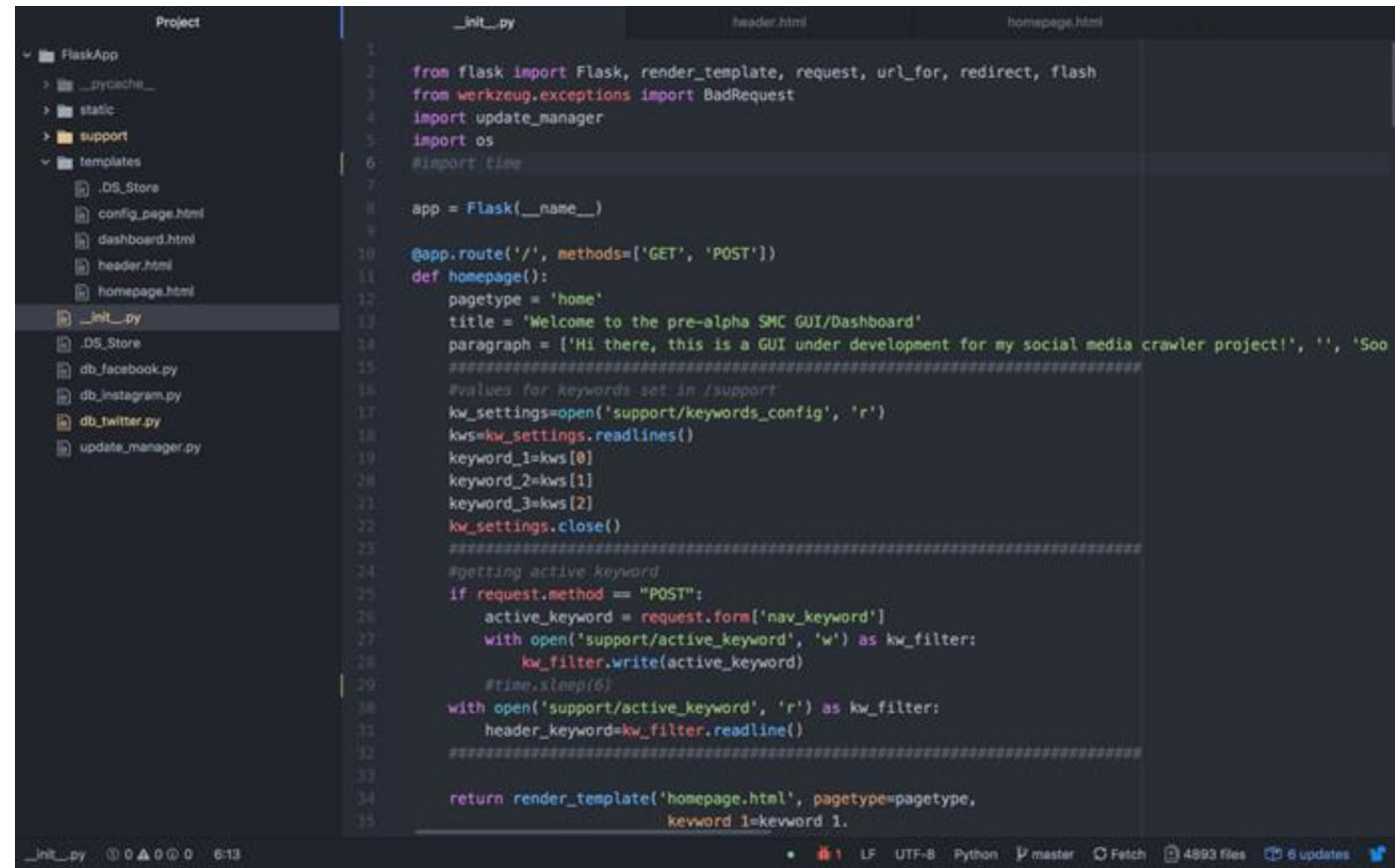
<https://jupyter.org/>

We will be using Jupyter notebooks for our practical sessions.



Text Editors

- Another method to write python scripts is using text editors
- Some popular text editors:
 - Vim (Linux terminal text editor)
 - Atom (popular open source editor)
 - Sublime Text (popular proprietary text editor)
 - Notepad ++ (Windows only)
- Usually highly customizable



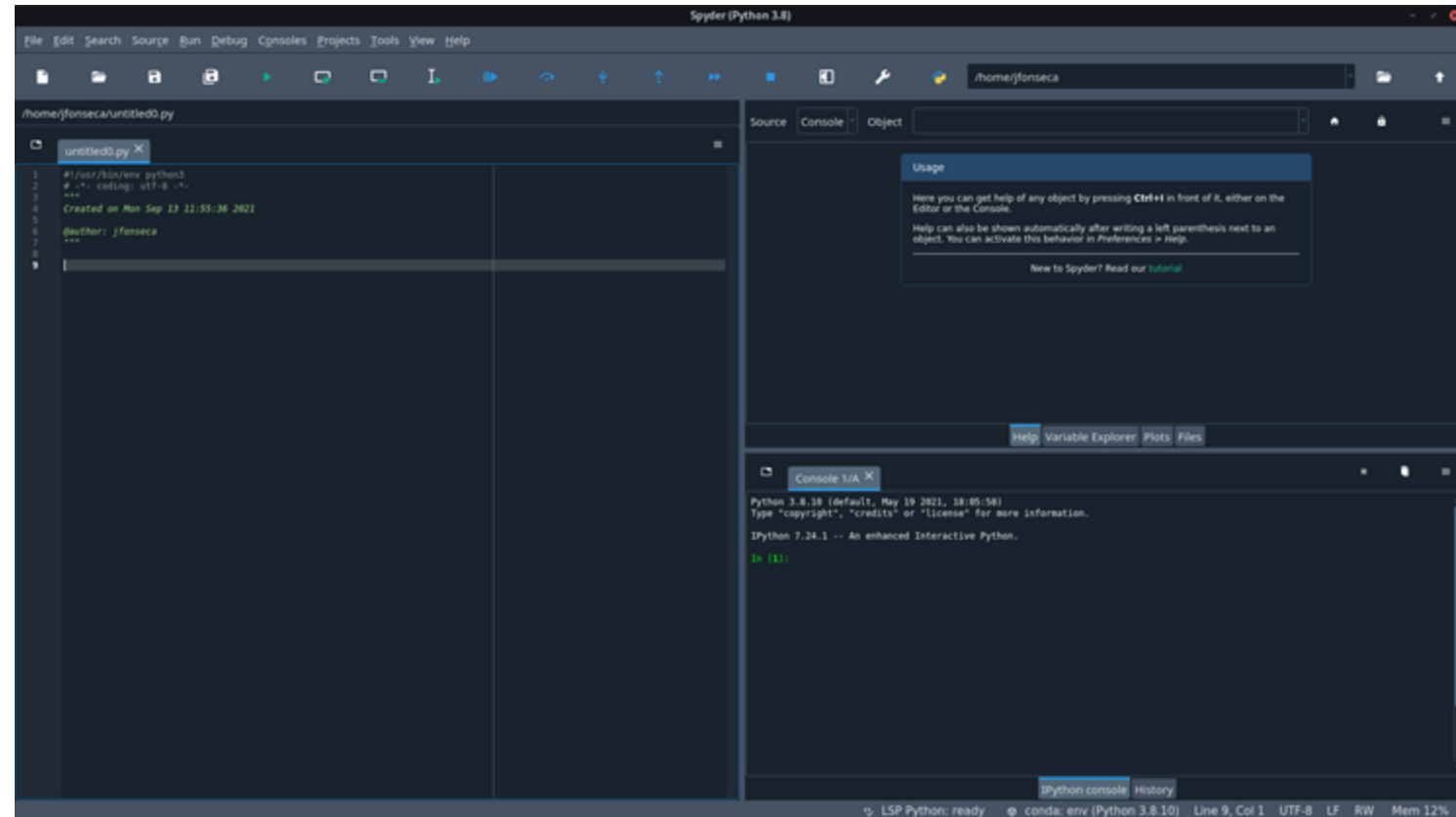
```
1 from flask import Flask, render_template, request, url_for, redirect, flash
2 from werkzeug.exceptions import BadRequest
3 import update_manager
4 import os
5 #import time
6
7
8 app = Flask(__name__)
9
10 @app.route('/', methods=['GET', 'POST'])
11 def homepage():
12     pagetype = 'home'
13     title = 'Welcome to the pre-alpha SMC GUI/Dashboard'
14     paragraph = ['Hi there, this is a GUI under development for my social media crawler project!', ' ', 'Soo
15     =====
16     #values for keywords set in /support
17     kw_settings=open('support/keywords_config', 'r')
18     kws=kw_settings.readlines()
19     keyword_1=kws[0]
20     keyword_2=kws[1]
21     keyword_3=kws[2]
22     kw_settings.close()
23     =====
24     #getting active keyword
25     if request.method == "POST":
26         active_keyword = request.form['nav_keyword']
27         with open('support/active_keyword', 'w') as kw_filter:
28             kw_filter.write(active_keyword)
29         #time.sleep(6)
30     with open('support/active_keyword', 'r') as kw_filter:
31         header_keyword=kw_filter.readline()
32     =====
33
34     return render_template('homepage.html', pagetype=pagetype,
35                             keyword_1=keyword_1,
```

Atom Text Editor

Integrated Development Environment (IDE)

- Popular IDE's:
 - Spyder
 - PyCharm
 - **VSCode**
 - Rodeo
- Anaconda comes with Spyder and VSCode

Usage of IDE and/or Text editor
(and which ones to use) comes
down to personal preference



Everything is on Moodle

202526 - Data Mining - Turma TP1 | TP2 - S1

Disciplina Configurações Participantes Pauta Atividades Mais ▾


✓ **Geral**

[Contrair tudo](#)

 [Anúncios](#)

 [Course Syllabus](#)

 [Project Guidelines](#)

 [Teams Session](#)

 Oculto para os alunos

 [Practical Sessions](#)

 Oculto para os alunos

You should be able to see this now :)

Please read the guidelines



Project Guidelines

Pasta

Configurações

Mais ▾

✦ Funcionalidades da IA ▾

Editar



data



[DM MAA Report Template.docx](#)



[DM2526 ProjectGuidelines slides.pdf](#)



[DM2526 ProjectGuidelines v3.pdf](#)

Amazing International Airlines Inc.

Data Mining Project Guidelines

Fall Semester 2025-2026

Last Updated: 25 August 2025

1 Introduction

Amazing International Airlines Inc. (AIAI) is facing the challenge of designing personalized services and marketing strategies for its diverse customer base. In today's highly competitive airline industry, leveraging data-driven approaches to understand customer segments is crucial for improving satisfaction, increasing retention, and maximizing revenue potential.

Customer segmentation [4, 1] enables AIAI to identify distinct groups within their loyalty program. For instance, some customers may prioritize premium services and convenience, while others may be more cost-conscious and focused on basic travel needs. Additionally, certain groups may display seasonal travel patterns or specific route preferences. By uncovering these patterns, AIAI can tailor services, loyalty rewards, and marketing communications to meet the unique needs and behaviors of each segment.

Project Part 1: EDA

4 Deliverable 1: Exploratory Data Analysis (30 points)

This deliverable lays the groundwork for segmentation by examining the airline loyalty dataset in detail. The focus is on uncovering meaningful patterns, identifying limitations, and generating initial hypotheses about customer groups.

Key Tasks:

- Conduct descriptive statistics and visualizations to highlight distributions, trends, and anomalies, while noting which variables appear most relevant for segmentation.
- Assess data quality issues and evaluate how these may affect clustering reliability.
- Identify preliminary behavioral signals that suggest distinct types of customers.
- Develop and justify engineered features. Show how these derived variables capture richer aspects of customer behavior and explain their potential contribution to clustering models.

In presenting results, consider addressing the following:

1. Which findings were most unexpected or insightful, and what do they reveal about likely customer clusters?
2. What data limitations pose the greatest risks for clustering, and how might they be mitigated?
3. Which patterns in customer activity, including those revealed by engineered features, suggest natural groupings, and what cluster characteristics do you anticipate?
4. How would you communicate these insights to non-technical stakeholders? Include a clear explanation of the expected number of clusters, the most important differentiating features, and any anticipated challenges.

Submission Deadline



November 4th

Project Part 1: EDA

Academic paper example:

M. Maphosa, W. Doorsamy and B. S. Paul, "Student Performance Patterns in Engineering at the University of Johannesburg: An Exploratory Data Analysis," in *IEEE Access*, vol. 11, pp. 48977-48987, 2023, doi: 10.1109/ACCESS.2023.3277225.

<https://ieeexplore.ieee.org/abstract/document/10128127>

Project Part 2: Final Report

5 Deliverable 2: Clustering Analysis (60 points)

This phase applies clustering techniques to the airline loyalty dataset in order to generate meaningful customer segments. The emphasis is on experimenting with multiple perspectives, validating results, and merging insights into a comprehensive solution that supports business objectives.

Key Tasks:

- Prepare the dataset for modeling.
- Analyze available features to determine which segmentation perspectives can be meaningfully applied.
- Perform segmentation using the identified perspectives.
- Apply at least two clustering approaches within each perspective and compare results.
- Propose a final merged segmentation solution that integrates the most important insights across perspectives into a coherent framework.

When presenting results, consider addressing the following:

1. Which clustering method(s) produced the most interpretable and stable results for each perspective?
2. How many clusters best represent the customer base overall, and what evidence supports this decision?
3. What differentiating features emerged as most important within each perspective, and how do they complement one another in the merged solution?
4. How would you describe the final set of customer segments to a business audience? Summarize defining traits, potential marketing opportunities, and any challenges encountered when integrating perspectives.

Submission Deadline



January 3rd

Project Part 3: Discussion

Deliverable 3: Discussion	10	Date TBA
Individual Assessment	10	In Person

Project Bonus (**Optional**)

7 Optional Bonus Components

Each deliverable can include an optional bonus component up to 20% of that deliverable's points. Bonus components are designed to reward exceptional work while maintaining core assignment focus.

Check Guidelines to learn about the options ;)

Let's get started!

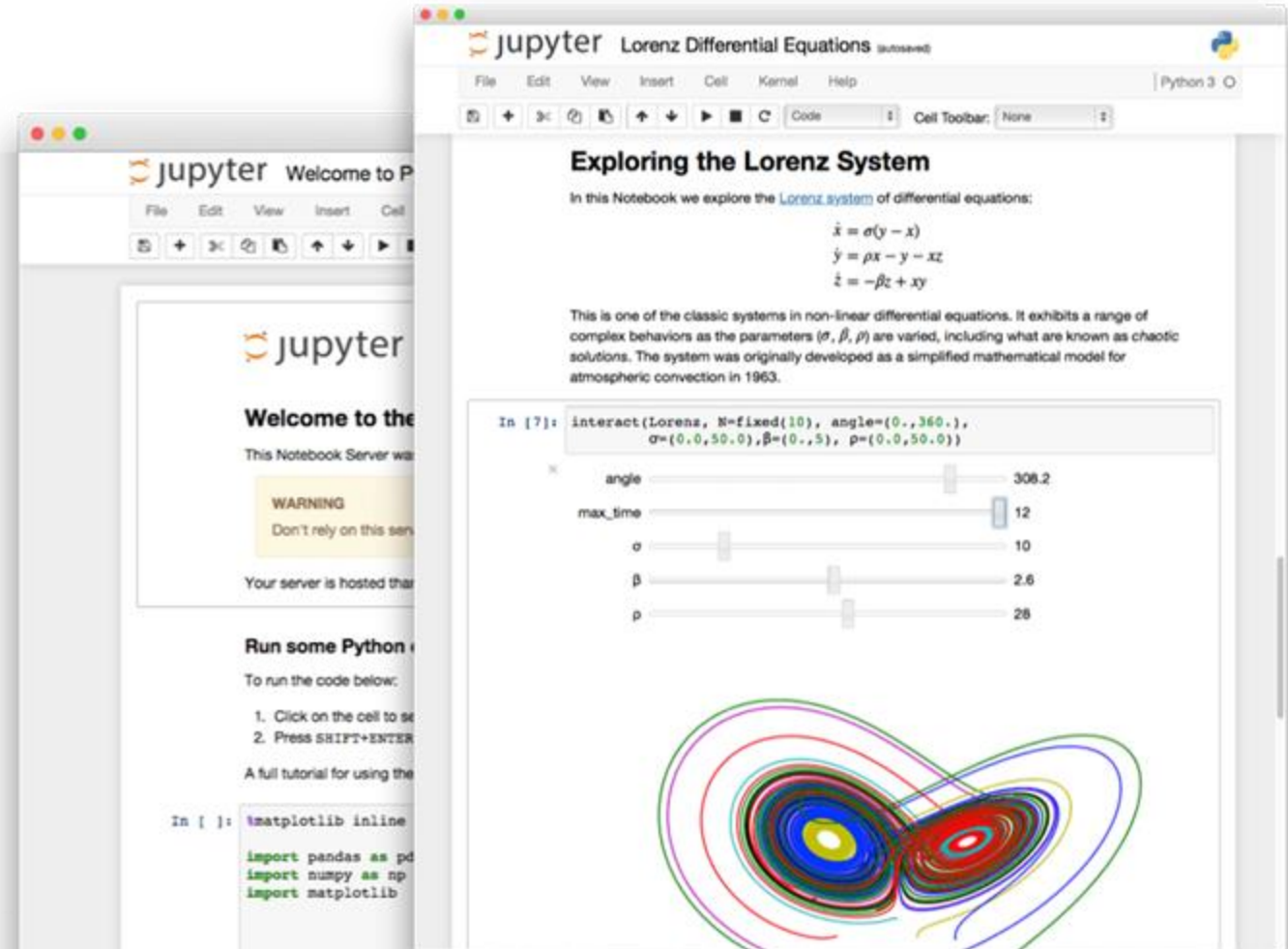
Next: Setting up our tools

The Jupyter Notebook

<http://jupyter.org/>

Let's try it out!

- Install and Open Anaconda Navigator
- Start Jupyter Notebook



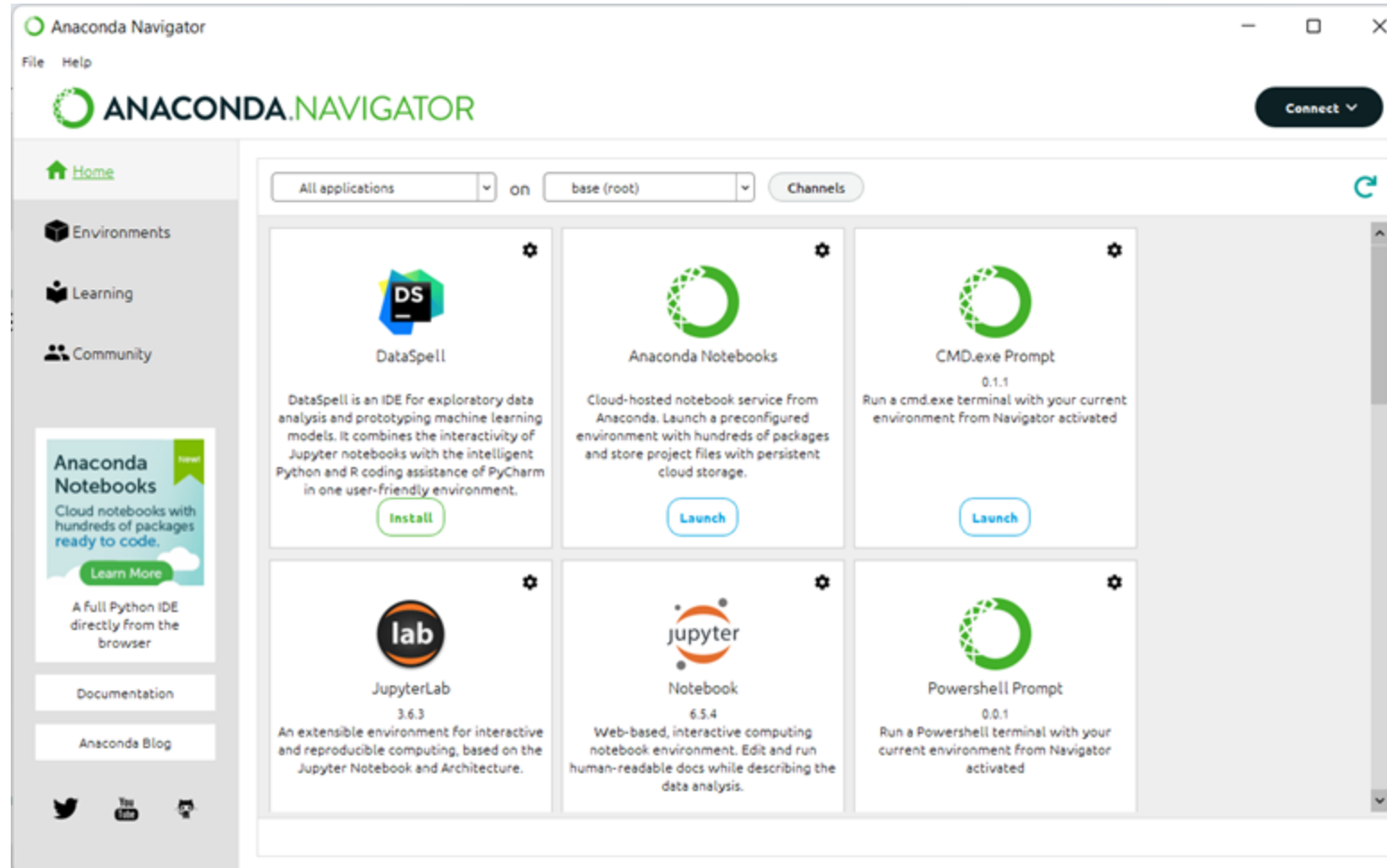
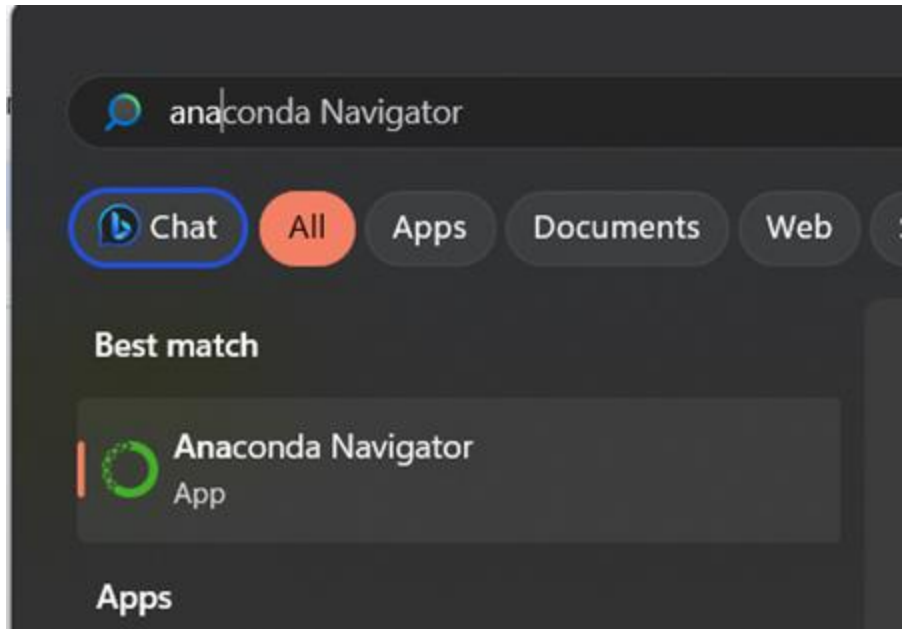
Setting up our tools

Two options:

1. **With GUI (Anaconda Navigator)**
1. Command line (miniconda) (skip to slide 62)

Setting up our tools

Load Anaconda Navigator



aext-share-
notebook

The aext-share-notebook component of anaconda-toolbox

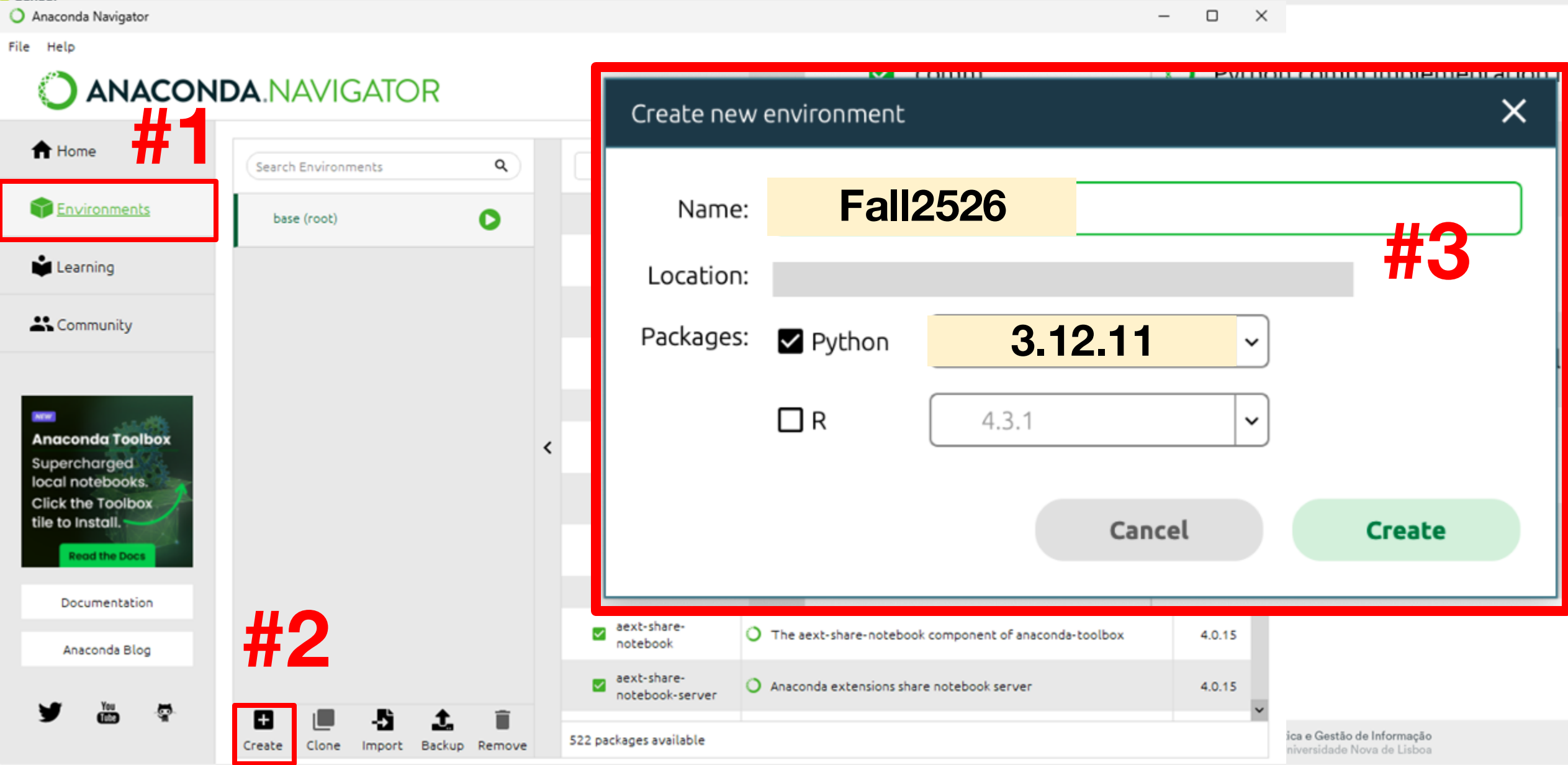
4.0.15

aext-share-
notebook-server

Anaconda extensions share notebook server

4.0.15

Create a new environment: Fall2526



Create a new environment: Fall2526

ANACONDA NAVIGATOR

Connected to Cloud

Connect ▾

Home

Environments

Learning

Community

Search Environments



base (root)

.conda

Fall2526



mcbook

Installed ▾

Channels

Update index...

Search Packages



Name

▼

T

Description

Version



bzip2



High-quality data compressor

1.0.8



ca-certificates



Certificates for use with other packages.

2025.7.15



expat



Expat xml parser library in c

2.7.1



libcxx



Llvm c++ standard library

20.1.8



libffi



A portable foreign function interface library

3.4.4



ncurses



Library for text-based user interfaces

6.5



openssl



Openssl is an open-source implementation of the ssl and tls protocols

3.0.17



pip



Pypa recommended tool for installing python packages

25.2



python



General purpose programming language

[3.12.11](#)

readline



Library for editing command lines as they are typed in

8.3



setuptools



Download, build, install, upgrade, and uninstall python packages

78.1.1



sqlite



Implements a self-contained, zero-configuration, sql database engine

3.50.2



tk



A dynamic programming language with gui support. bundles tcl and tk.

8.6.15



tzdata



The time zone database (called tz, tzdb or zoneinfo)

2025b



wheel



A built-package format for python.

0.45.1



xz



Data compression software with high compression ratio

5.6.4



zlib



A massively spiffy yet delicately unobtrusive compression library

[1.2.13](#)

17 packages available

**Anaconda
Quick Start
Environments**Jump into pre-configured
environments by
project or industry.
Clean dependencies,
faster development

Launch Your Environment

Documentation

Anaconda Blog



Create



Clone



Import



Backup



Remove

Add conda channel: conda-forge

base (root)

DM2425

Installed

Channels

Update index...

Name	T	Description
✓ python	○	General purpose programming language
✓ setuptools	○	Download, build, install, upgrade, and uninstall python
✓ sqlite	○	Implements a self-contained, zero-configuration, sql c engine
✓ tk	○	A dynamic programming language with gui support. b and tk

Add conda channel: conda-forge

All

Channels

Update index...

pandas X

Name

☐ ydata-profiling

☐ tabula-py

☐ streamz

☐ spatialpandas

☐ sklearn-pandas

☐ qpd

Manage channels you want Navigator to include.

Add...

<div>Generate profile report for pandas dataframe defaults</div>	4.8.3 <div></div>
<div>Simple wrapper of tabula-java: extract table from pdf into pandas dataframe</div>	2.6.0
<div>Manage streaming data, optionally with dask and pandas</div>	0.6.3
<div>Pandas extension arrays for spatial/geometric operations</div>	0.4.10
<div>Pandas integration with sklearn</div>	2.2.0
<div>Query pandas using sql</div>	0.4.4

Cancel

Update channels

Add conda channel: conda-forge

Manage channels you want Navigator to include.

Add...

☐ Generate profile report for pandas dataframe 4.8.3

defaults



☐ Simple wrapper of tabula-java: extract table from pdf into pandas dataframe 2.6.0

conda-forge



☐ Manage streaming data, optionally with dask and pandas 0.6.3

☐ Pandas extension arrays for spatial/geometric operations 0.4.10

☐ Pandas integration with sklearn 2.2.0

☐ Query pandas using sql 0.4.4

Cancel

Update channels

Add conda channel: conda-forge

base (root)

DM2425

Installed

Channels

Update index...

Name	T	Description
✓ python	○	General purpose programming language
✓ setuptools	○	Download, build, install, upgrade, and uninstall python
✓ sqlite	○	Implements a self-contained, zero-configuration, sql c engine
✓ tk	○	A dynamic programming language with gui support. b and tk

Install libraries

File Help

ANACONDA.NAVIGATOR

Connect

Home

Environments

Learning

Community

Anaconda Toolbox
Supercharged local notebooks.
Click the Toolbox tile to Install.
Read the Docs

Documentation

Anaconda Blog

Twitter YouTube GitHub

Create Clone Import Backup Remove

Search Environments

base (root)

DM2425

make sure to show "All"

Installed

- Installed
- Not installed
- Updatable
- Selected
- All

Channels Update index...

Search Packages

Description	Version
General purpose programming language	3.12.4
Download, build, install, upgrade, and uninstall python packages	72.1.0
Implements a self-contained, zero-configuration, sql database engine	3.45.3
tk	8.6.14
tzdata	2024a
vc	14.40
vs2015_runtime	14.40.33
wheel	0.43.0
xz	5.4.6
zlib	1.2.13

16 packages available

Install libraries

search for the libraries we need

Anaconda Navigator

[Upgrade Now](#) [Connect](#)

All Channels Update index...

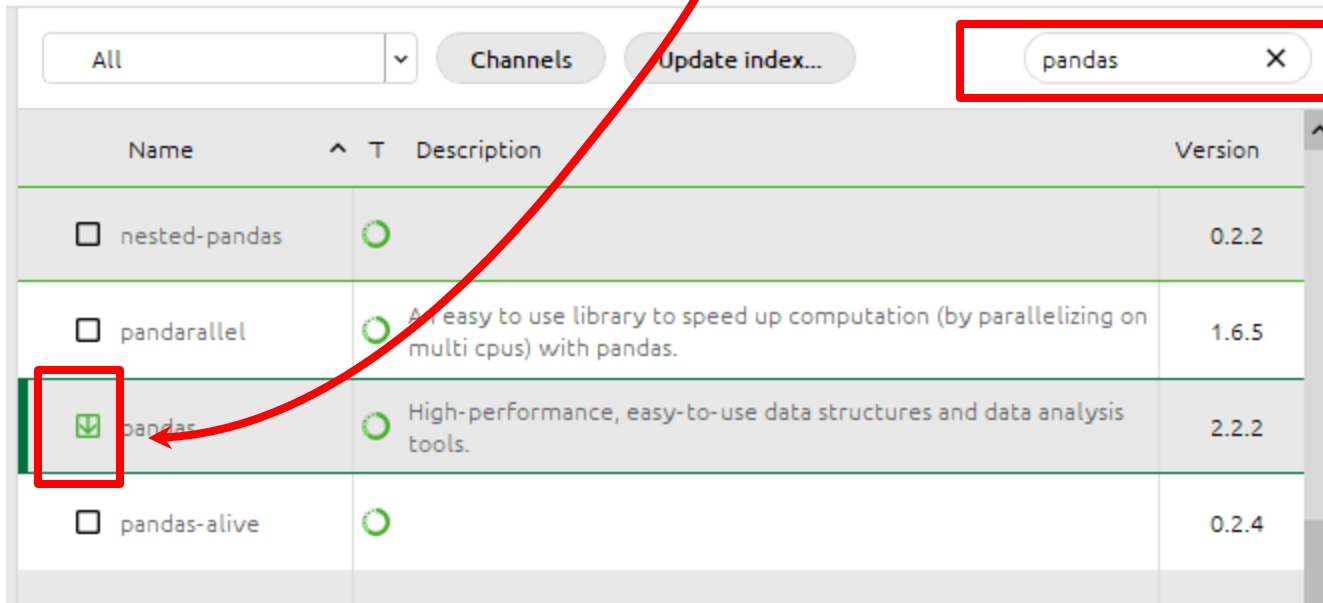
jupyter

Name	T	Description	Version
<input type="checkbox"/> ipywidgets		Jupyter interactive widgets	8.1.0
<input type="checkbox"/> jupyter		Jupyter metapackage. install all the jupyter components in one go.	1.0.0
<input type="checkbox"/> jupyter-archive			3.4.0
<input type="checkbox"/> jupyter-black			0.3.4
<input type="checkbox"/> jupyter-book			0.8.2
<input type="checkbox"/> jupyter-cache			0.6.1

Install libraries

search for the libraries we need

- search package name, then select the check box on the left of the name



- after selecting, you can search the next library name, then select that, and so on

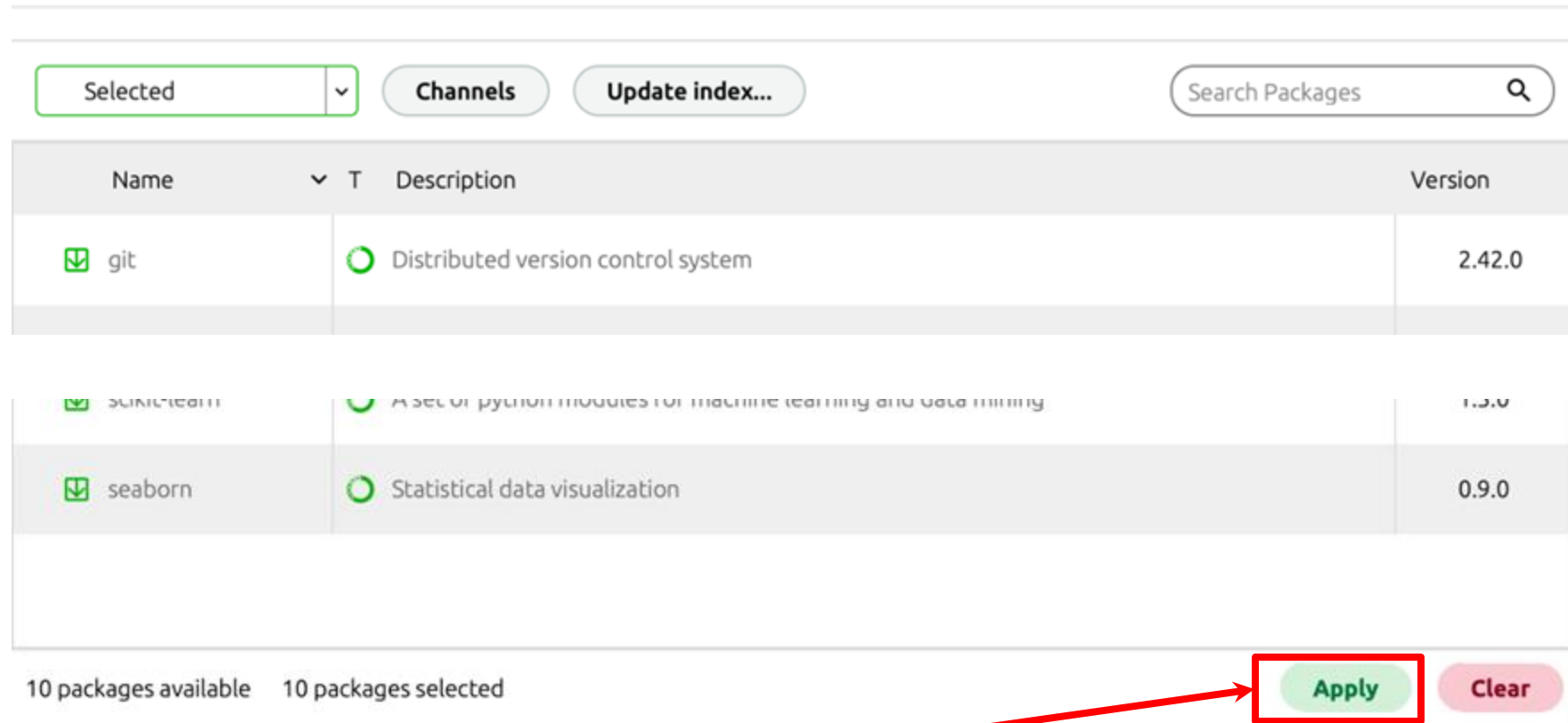
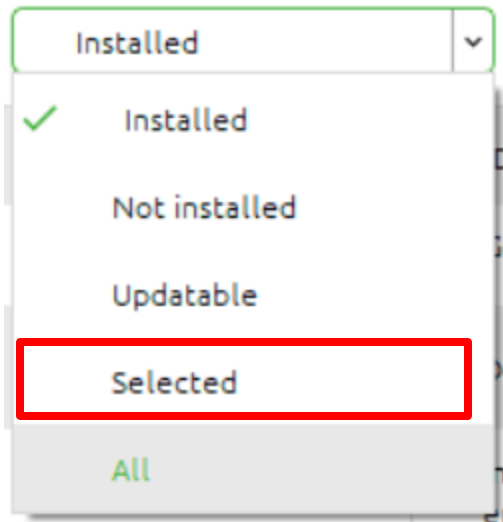
1. jupyter
2. scikit-learn
3. scikit-image
4. numpy
5. pandas
6. ipywidgets
7. matplotlib
8. seaborn
9. minisom
10. ydata-profiling

Install libraries

after searching and selecting everything on the list,

you can clear the search bar by clicking the 'x'

then filter the "Selected" option to see if you got everything



then click Apply

Install libraries

Install Packages

The following packages will be modified:

Solving package specifications

Cancel
Apply

Install Packages

15 packages will be installed

	Name	Unlink	Link	Channel	Action
1	*threadpoolctl	-	3.5.0	pkgs/main	Install
2	*tbb	-	2021.8.0	pkgs/main	Install
3	*scipy	-	1.13.1	pkgs/main	Install
4	*pybind11-abi	-	5	pkgs/main	Install
5	*numpy-base	-	1.26.4	pkgs/main	Install
6	*numpy	-	1.26.4	pkgs/main	Install

* indicates the package is a dependency of a selected package

Cancel

Apply

Go back to home


Anaconda Navigator

File Help

ANACONDA.NAVIGATOR Connect ▾

Home Environments Learning Community


Installed applications or DM2425 Channels



Anaconda Cloud Notebooks

Cloud-hosted notebook service from Anaconda. Launch a preconfigured environment with hundreds of packages and store project files with persistent cloud storage.

[Launch](#)




JupyterLab

4.2.5

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

[Launch](#)



Notebook

7.2.2

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

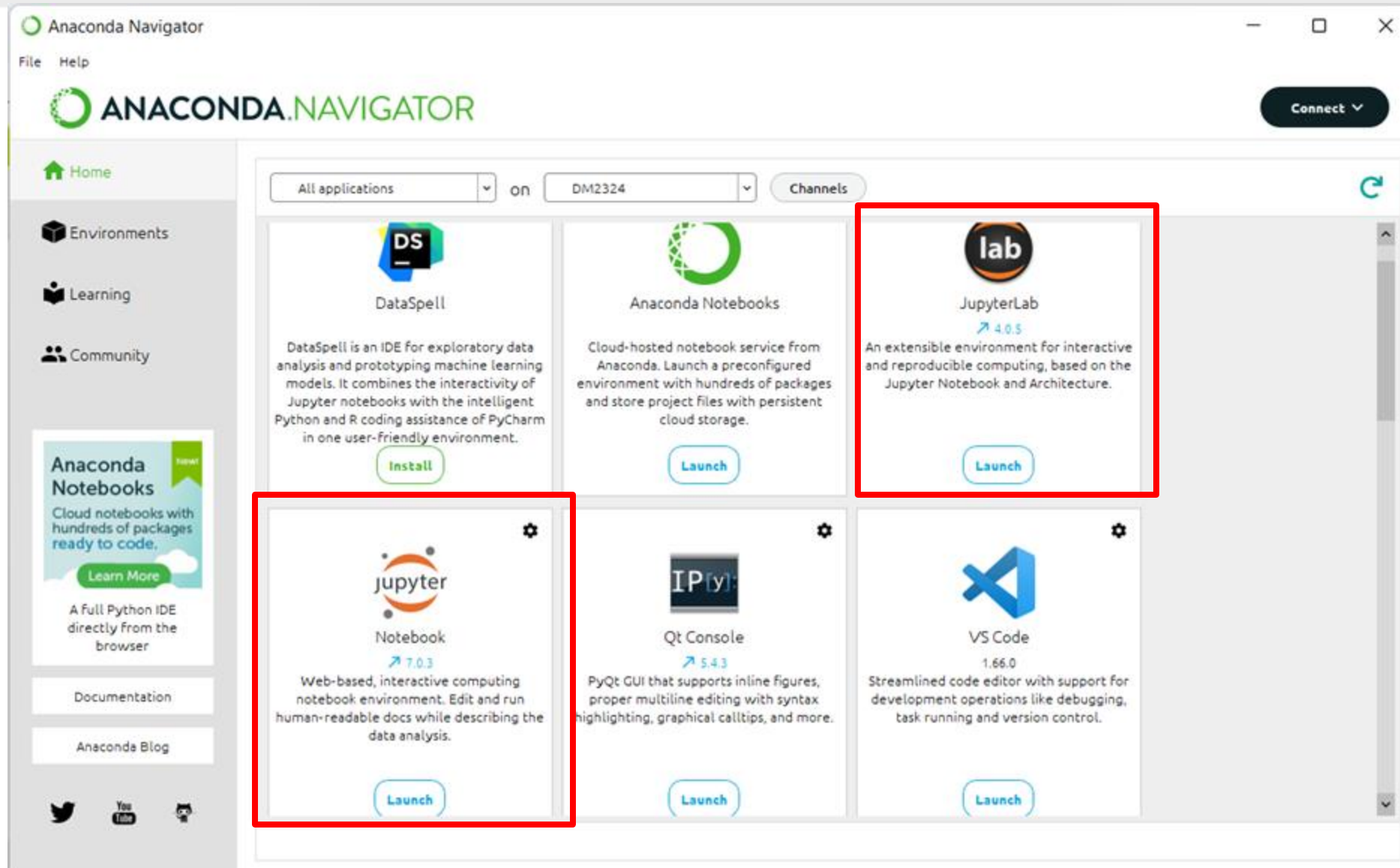
[Launch](#)

Test Jupyter Notebook

Test Jupyter notebook

Close all Terminal windows / Anaconda Navigator / Anaconda Prompt

Test loading Jupyter notebook

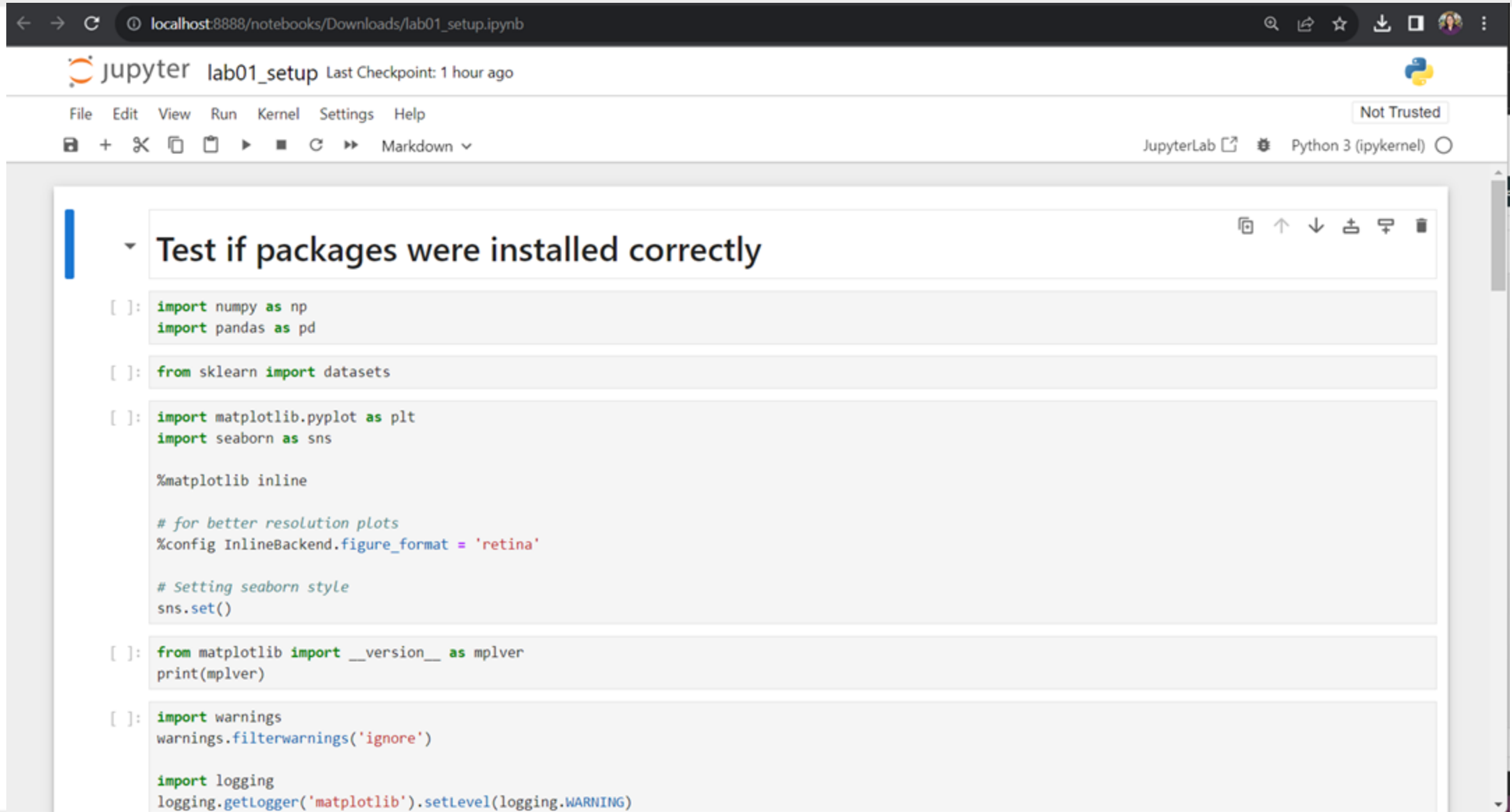


Test loading Jupyter notebook

The screenshot shows the JupyterLab web interface in a browser window at localhost:8888/tree. The interface includes a top navigation bar with 'File', 'View', 'Settings', and 'Help' menus. Below this is a tabbed interface with 'Files' and 'Running' tabs. The 'Files' tab is active, displaying a file browser. At the top of the file browser, there is a prompt 'Select items to perform actions on them.' and buttons for 'New', 'Upload', and a refresh icon. The main area shows a list of files and folders. A red text overlay 'Look for the lab files you downloaded' is positioned over the file list.

Name	Last Modified	File Size
anaconda3	1 hour ago	
Contacts	last year	
Desktop	3 months ago	
Documents	11 months ago	
Downloads	35 minutes ago	
Favorites	last year	
Links	last year	
Music	last year	
OneDrive	last year	
Pictures	12 months ago	
Saved Games	last year	
scikit_learn_data	3 months ago	
Searches	last year	
Videos	last year	

Load the lab01_setup.ipynb notebook file



The screenshot shows a JupyterLab interface with a notebook titled "lab01_setup". The notebook is open in a web browser at the URL "localhost:8888/notebooks/Downloads/lab01_setup.ipynb". The interface includes a menu bar with options like File, Edit, View, Run, Kernel, Settings, and Help. A toolbar with various icons for file operations and execution is visible. The notebook content is displayed in a central area with a sidebar on the left. The sidebar shows a section titled "Test if packages were installed correctly" with a blue bar next to it. The notebook contains several code cells with Python code for importing libraries and setting up the environment.

```
[ ]: import numpy as np
import pandas as pd

[ ]: from sklearn import datasets

[ ]: import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

# for better resolution plots
%config InlineBackend.figure_format = 'retina'

# Setting seaborn style
sns.set()

[ ]: from matplotlib import __version__ as mplver
print(mplver)

[ ]: import warnings
warnings.filterwarnings('ignore')

import logging
logging.getLogger('matplotlib').setLevel(logging.WARNING)
```

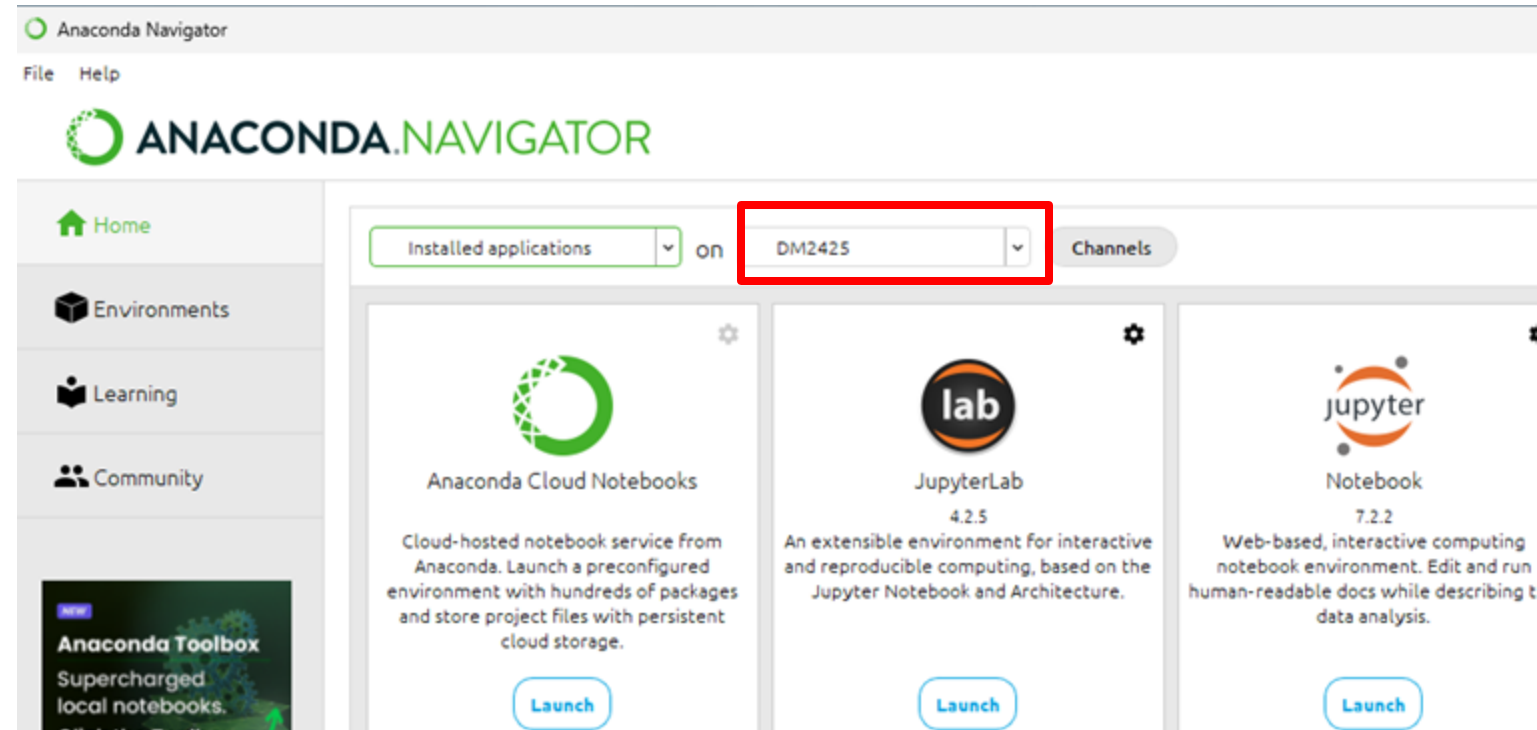
Don't worry about
understanding the code at this point

Right now we just want to make sure that all the
packages we need are installed and work correctly

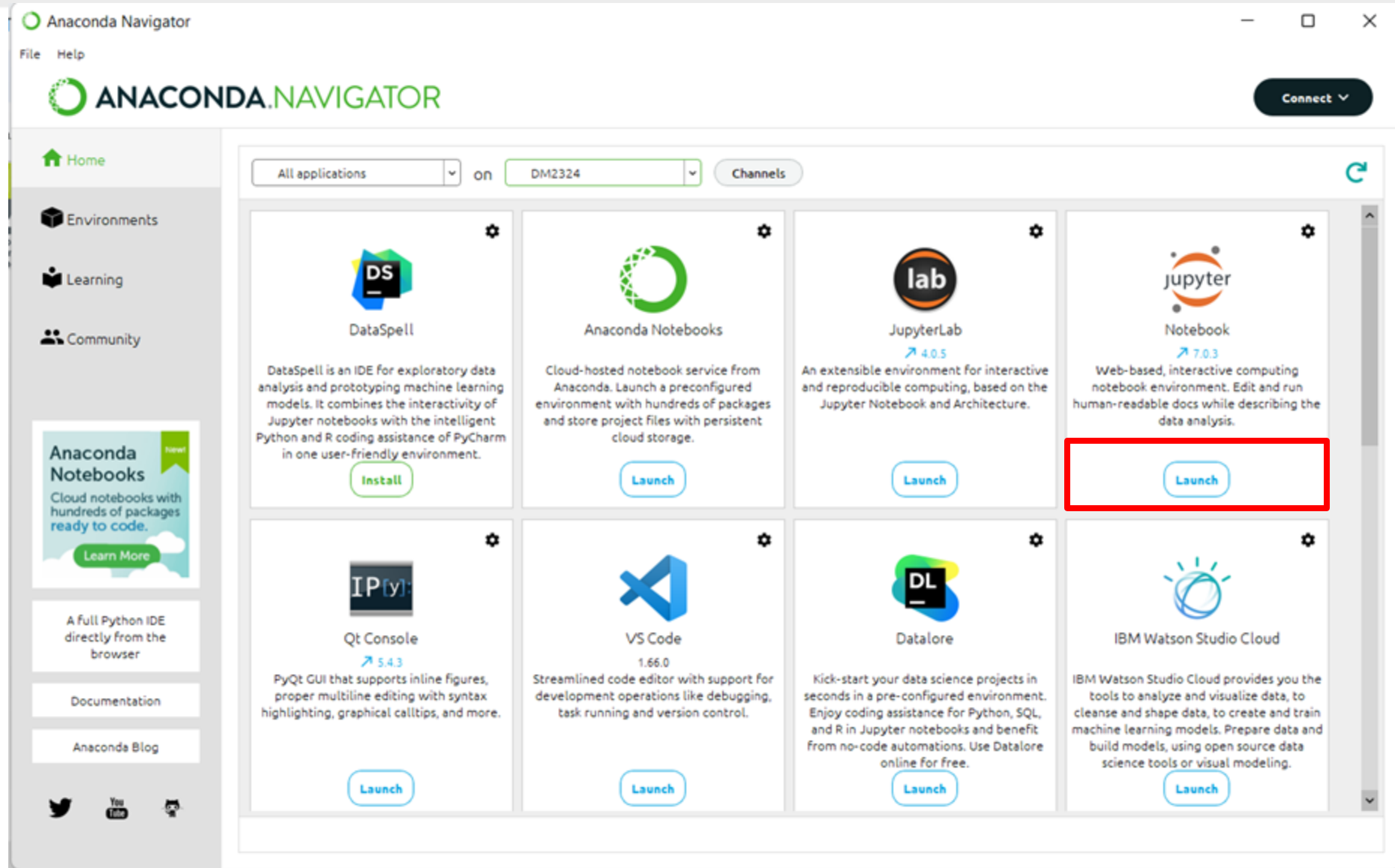
Test Jupyter notebook

Open Anaconda Navigator

Make sure you select the
environment we created



Test Jupyter notebook



Test Jupyter notebook

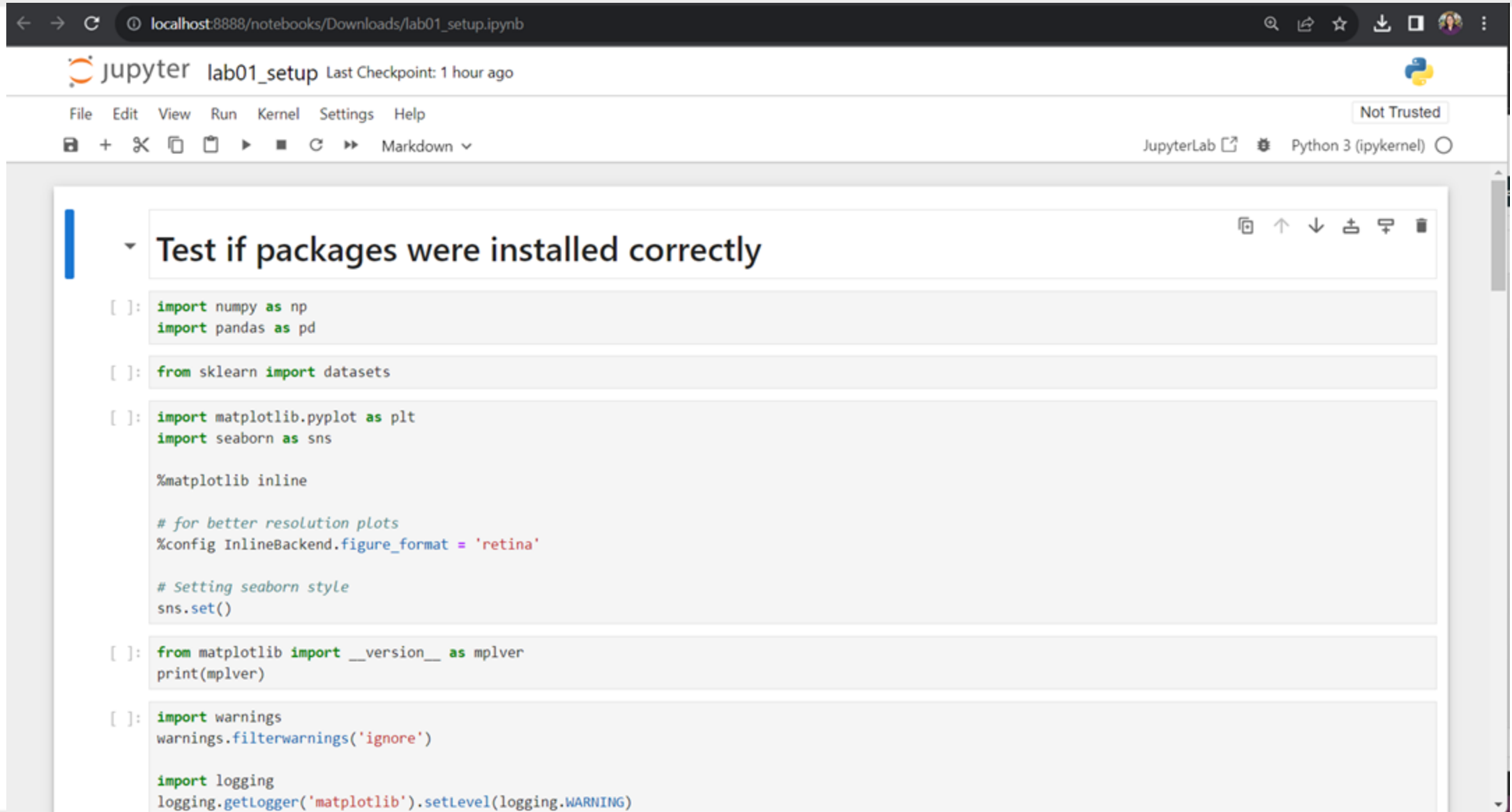
The screenshot shows the JupyterLab interface in a web browser at localhost:8888/tree. The 'Files' tab is active, displaying a file browser. A red text overlay reads 'Look for the lab files you downloaded'. The file list includes folders like anaconda3, Downloads, and scikit_learn_data.

Name	Last Modified	File Size
anaconda3	1 hour ago	
Contacts	last year	
Desktop	3 months ago	
Documents	11 months ago	
Downloads	35 minutes ago	
Favorites	last year	
Links	last year	
Music	last year	
OneDrive	last year	
Pictures	12 months ago	
Saved Games	last year	
scikit_learn_data	3 months ago	
Searches	last year	
Videos	last year	

Don't worry about
understanding the code at this point

Right now we just want to make sure that all the
packages we need are installed and work correctly

Load the notebook file



The screenshot shows a JupyterLab interface in a web browser. The address bar indicates the notebook is located at `localhost:8888/notebooks/Downloads/lab01_setup.ipynb`. The JupyterLab header shows the notebook name `lab01_setup` and the last checkpoint time `Last Checkpoint: 1 hour ago`. The interface includes a menu bar with `File`, `Edit`, `View`, `Run`, `Kernel`, `Settings`, and `Help`. Below the menu is a toolbar with icons for saving, opening, and running code. The main area displays a notebook with a section titled `Test if packages were installed correctly`. The code in the notebook is as follows:

```
[ ]: import numpy as np
import pandas as pd

[ ]: from sklearn import datasets

[ ]: import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

# for better resolution plots
%config InlineBackend.figure_format = 'retina'

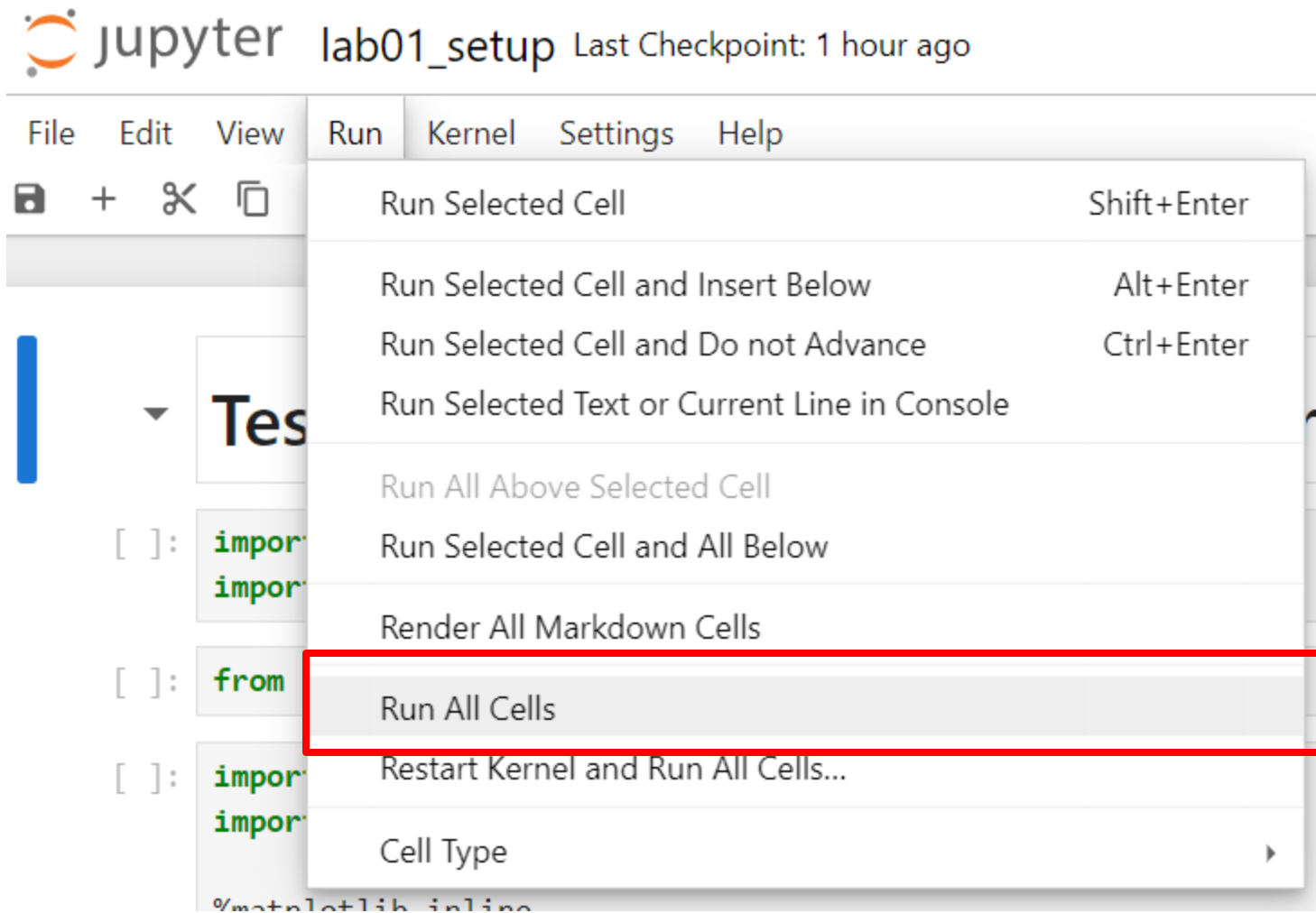
# Setting seaborn style
sns.set()

[ ]: from matplotlib import __version__ as mplver
print(mplver)

[ ]: import warnings
warnings.filterwarnings('ignore')

import logging
logging.getLogger('matplotlib').setLevel(logging.WARNING)
```

Run all cells



Does anyone still have any errors in the notebook?

Installing using command line

You can also use miniconda
instead of Anaconda Navigator

<https://docs.anaconda.com/free/anaconda/getting-started/distro-or-miniconda/>

Installing using the command line

Install miniconda

<https://docs.conda.io/projects/miniconda/en/latest/#quick-command-line-install>

Create environment using the command line

Windows: Open Anaconda Prompt (miniconda3)

Linux/Mac: Open Terminal

```
cd Downloads
```

```
conda env create -f fall2526_env.yml
```

```
conda activate Fall2526
```

Test Jupyter notebook + install other packages

Windows: Open Anaconda Prompt (miniconda3)

Linux/Mac: Open Terminal

```
conda activate Fall2526  
jupyter notebook
```

Follow the instructions in previous slides for testing Jupyter notebook and installing additional packages

Let's get started! (for real)

Next:

Jupyter notebook

Distance Matrix

Questions?

Morada: Campus de Campolide, 1070-312 Lisboa, Portugal

Tel: +351 213 828 610 | **Fax:** +351 213 828 611

Acreditações e Certificações da NOVA IMS

Cofinanciado por



UNIGIS



Computing
Accreditation
Commission

