



Módulo 2 – Obtención y Preparación de Datos



Librerías para Ciencia de Datos

Ciencia de Datos

Objetivos



- Reconocer librerías en Python para la ciencia de datos.

Librerías para la ciencia de datos

Una librería o biblioteca es un conjunto de funciones implementadas por otro programador que nos facilitan realizar tareas, principalmente porque no debemos volver a programar este código. En esta clase será vital el uso de librerías para poder analizar archivos con información. Algunas librerías utilizadas en el análisis y ciencia de datos:

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Scikit-learn
- Scipy
- Statsmodels

<https://blogvisionarios.com/e-learning/articulos-ia/10-librerias-python-para-data-science-y-machine-learning/>

Pandas

 pandas

About us ▾ Getting started Documentation Community ▾ Contribute

pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language.

[Install pandas now!](#)

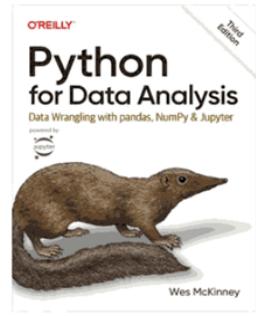
Latest version: 1.5.3

- What's new in 1.5.3
- Release date:
Jan 19, 2023
- Documentation (web)
- Download source code

Follow us

Get the book



Getting started

- Install pandas
- Getting started

Documentation

- User guide
- API reference
- Contributing to pandas
- Release notes

Community

- About pandas
- Ask a question
- Ecosystem

With the support of:



Previous versions

- 1.5.2 (Nov 22, 2022)

<https://pandas.pydata.org/>

Matplotlib

The screenshot shows the official Matplotlib website. At the top left is the Matplotlib logo. The top navigation bar includes links for Plot types, Examples, Tutorials, Reference, User guide, Develop, and Release notes. To the right of the navigation are icons for search, user profile, and social media (GitHub, LinkedIn, Twitter). The main title "Matplotlib: Visualization with Python" is centered above a brief description: "Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible." Below the description is a bulleted list of features:

- Create publication quality plots.
- Make interactive figures that can zoom, pan, update.
- Customize visual style and layout.
- Export to many file formats.
- Embed in JupyterLab and Graphical User Interfaces.
- Use a rich array of third-party packages built on Matplotlib.

A purple button labeled "Try Matplotlib (on Binder)" with a right-pointing arrow is positioned below the feature list. At the bottom of the page are five navigation links with icons: "Getting Started" (monitor), "Examples" (folder), "Reference" (book), "Cheat Sheets" (document), and "Documentation" (color wheel).

<https://matplotlib.org/>

Seaborn

The screenshot shows the official Seaborn website. At the top, there's a navigation bar with links for 'Installing', 'Gallery', 'Tutorial', 'API', 'Releases', 'Citing', and 'FAQ'. To the right of the navigation are icons for search, GitHub, RSS feed, and Twitter. Below the navigation is the title 'seaborn: statistical data visualization' in a large, dark blue font. Underneath the title are six small square thumbnails, each displaying a different type of plot generated by Seaborn, such as density plots, scatter plots, and box plots. Below these thumbnails is a brief introduction to the library: 'Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.' Further down, there's more descriptive text about the library's purpose and how to get started. On the right side of the page, there are two boxes: 'Contents' which lists links to 'Installing', 'Gallery', 'Tutorial', 'API', 'Releases', 'Citing', and 'FAQ'; and 'Features' which lists several new features with links to their API and tutorial pages.

seaborn: statistical data visualization

Seaborn is a Python data visualization library based on matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

For a brief introduction to the ideas behind the library, you can read the introductory notes or the paper. Visit the installation page to see how you can download the package and get started with it. You can browse the example gallery to see some of the things that you can do with seaborn, and then check out the tutorials or API reference to find out how.

To see the code or report a bug, please visit the GitHub repository. General support questions are most at home on stackoverflow, which has a dedicated channel for seaborn.

Contents

- Installing
- Gallery
- Tutorial
- API
- Releases
- Citing
- FAQ

Features

- **New** Objects: [API](#) | [Tutorial](#)
- Relational plots: [API](#) | [Tutorial](#)
- Distribution plots: [API](#) | [Tutorial](#)
- Categorical plots: [API](#) | [Tutorial](#)
- Regression plots: [API](#) | [Tutorial](#)
- Multi-plot grids: [API](#) | [Tutorial](#)
- Figure theming: [API](#) | [Tutorial](#)
- Color palettes: [API](#) | [Tutorial](#)

<https://seaborn.pydata.org>

Scikit-learn

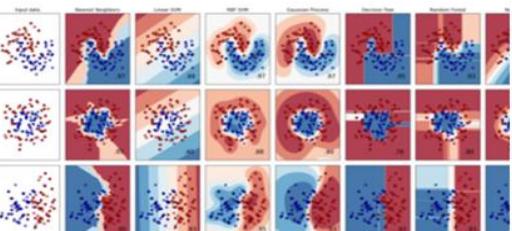
SCIKIT-LEARN

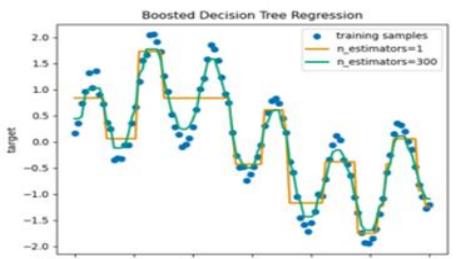
Install User Guide API Examples Community More ▾

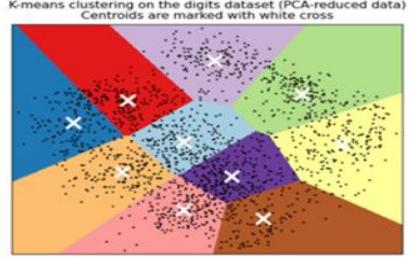
scikit-learn
Machine Learning in Python

Getting Started Release Highlights for 1.2 GitHub

Simple and efficient tools for predictive data analysis
Accessible to everybody, and reusable in various contexts
Built on NumPy, SciPy, and matplotlib
Open source, commercially usable - BSD license

Classification
Identifying which category an object belongs to.
Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...


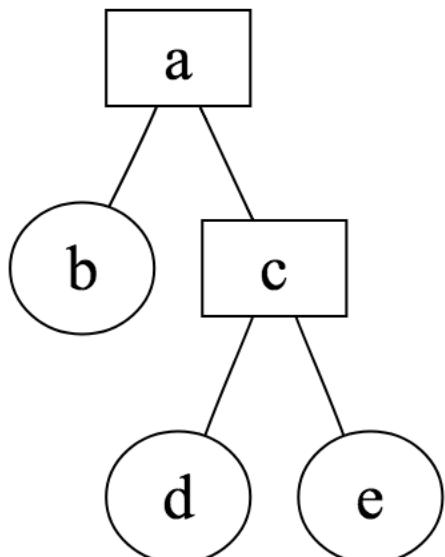
Regression
Predicting a continuous-valued attribute associated with an object.
Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...


Clustering
Automatic grouping of similar objects into sets.
Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...


<https://scikit-learn.org/>

Paquetes en Python

```
a/  
  __init__.py  
  b.py  
  c/  
    __init__.py  
    d.py  
    e.py
```



math.cos(0)
1.0

- **Módulos:** Para facilitar la gestión de programas grandes, el software se divide en trozos más pequeños. Contiene funciones.
- **Subpaquetes:** Contiene módulos agrupados por tipo.
- **Paquetes/Librerías:** Contiene cientos de módulos organizados en jerarquías.



KIBERNUM

Librerías de Python

Sintaxis:

- **import nombre_librería as abreviación estándar.**
- **from nombre_librería.sublibrería import sublibrería.**
- **from nombre_librería.sublibrería.módulo import módulo as abreviación estándar.**

```
from numpy.linalg import linalg
```

or

```
from numpy.linalg import linalg as lin
```

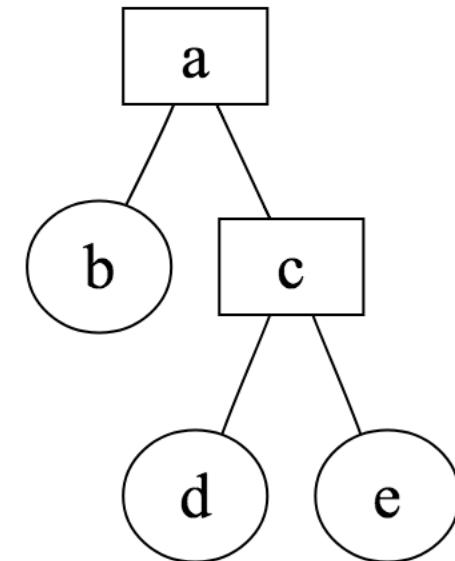
Librerías de Python

④ **from nombre_librería.sublibrería.módulo import función**

```
from numpy.linalg.linalg import det
```

Jerarquía:

- numpy package
- .
linalg subpackage
- .
linalg module
- .
det function



Dudas y consultas



KIBERNUM



Fin Presentación



KIBERNUM