



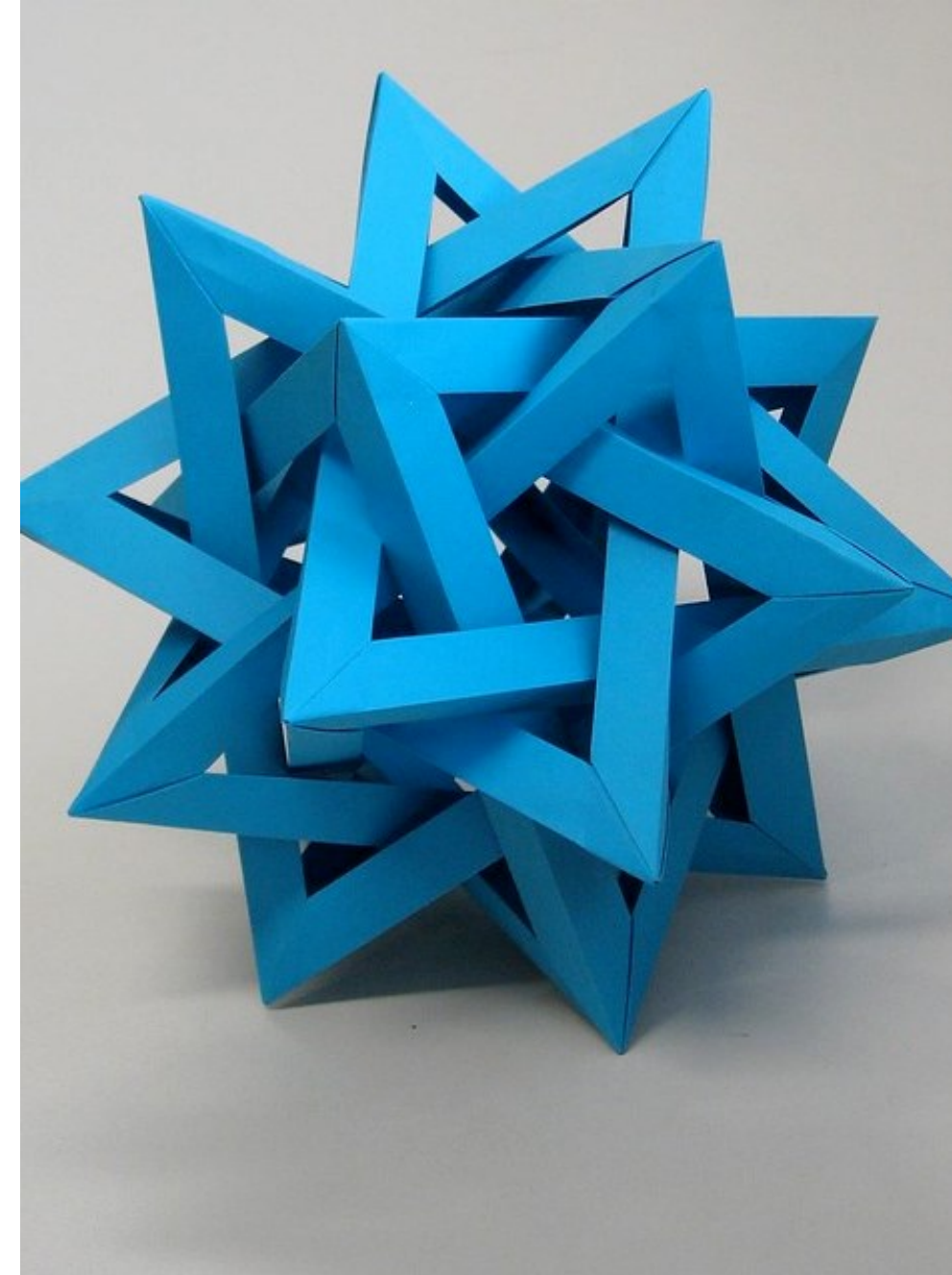
Politecnico
di Torino

Dipartimento
di Automatica e Informatica

Ambienti di calcolo e simulazione per la ricerca sperimentale

DOTTORATO DI RICERCA IN TECNOLOGIE E METODI PER LA
FORMAZIONE UNIVERSITARIA

PROF. FULVIO CORNO

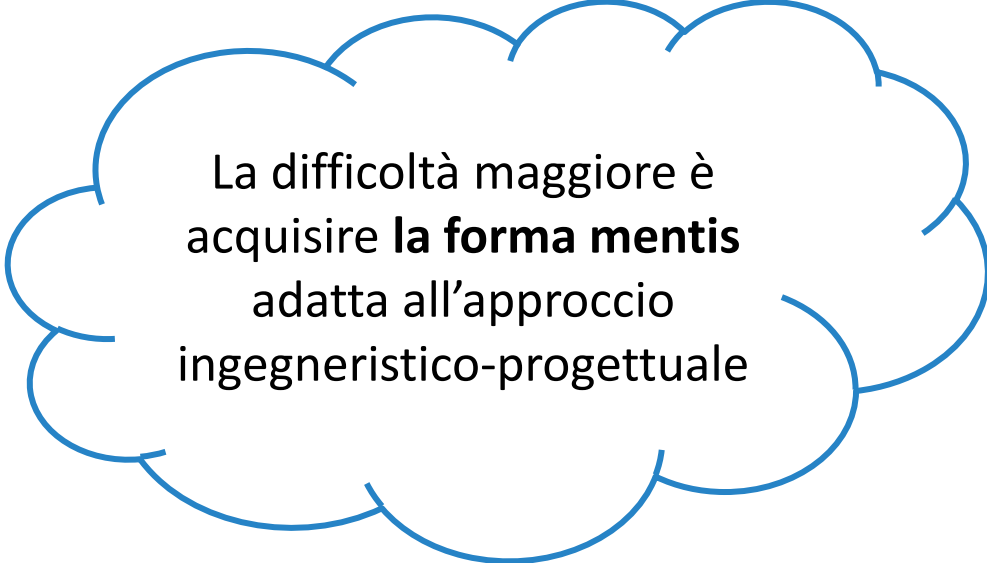


[This Photo](#) by Unknown Author is licensed under [CC BY-SA](#)

Un assaggio di... **Ingegneria**

■ Ingegneria =

- Saper progettare
- Risolvere problemi
- Trovare soluzioni
- Soddisfare le specifiche
- Nel rispetto dei vincoli
- Con gli strumenti disponibili



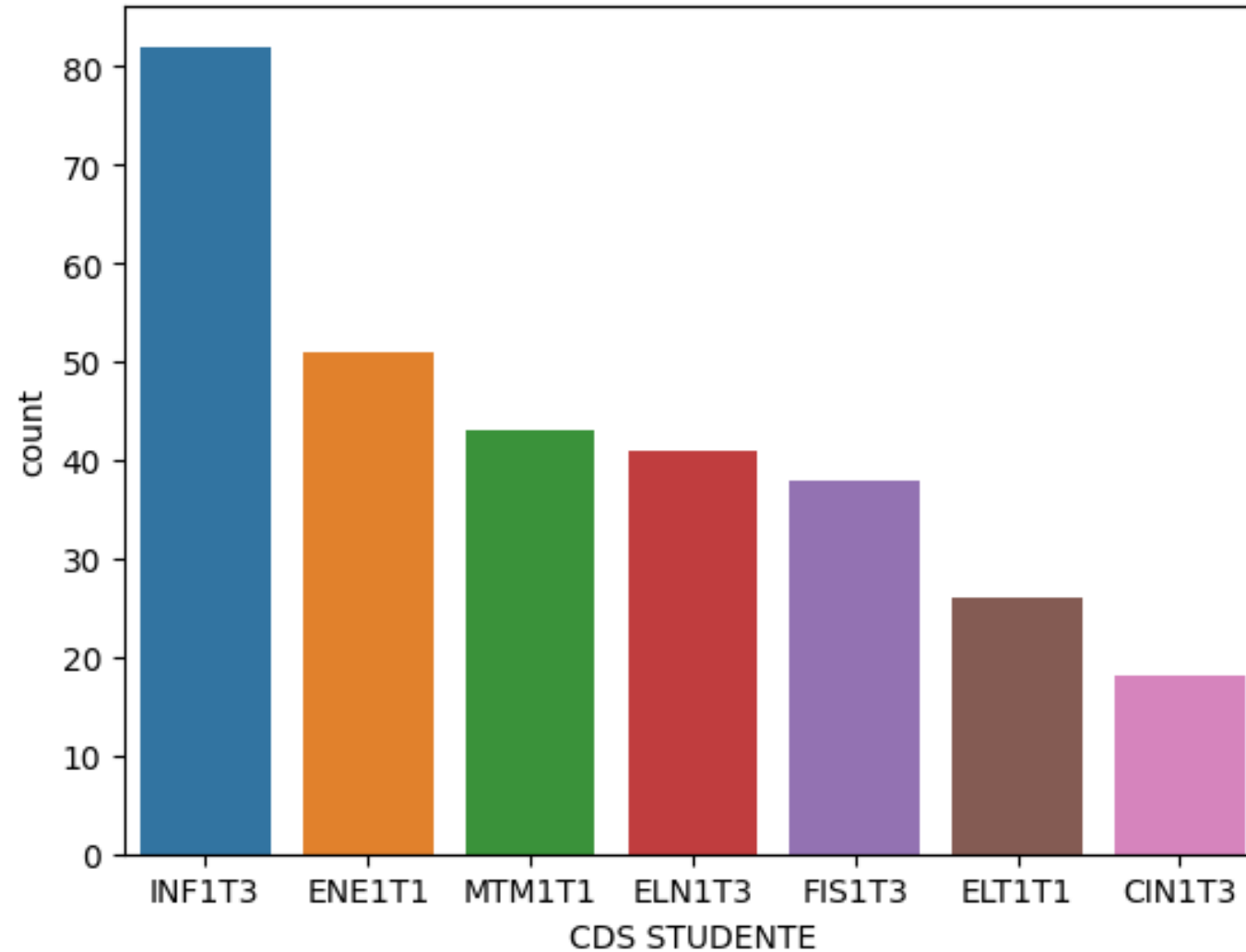
La difficoltà maggiore è
acquisire **la forma mentis**
adatta all'approccio
ingegneristico-progettuale

■ Ingegneria informatica =

- Problemi di ogni genere (calcolo, gestione dati, interazione, ...)
- Lo strumento è il calcolatore

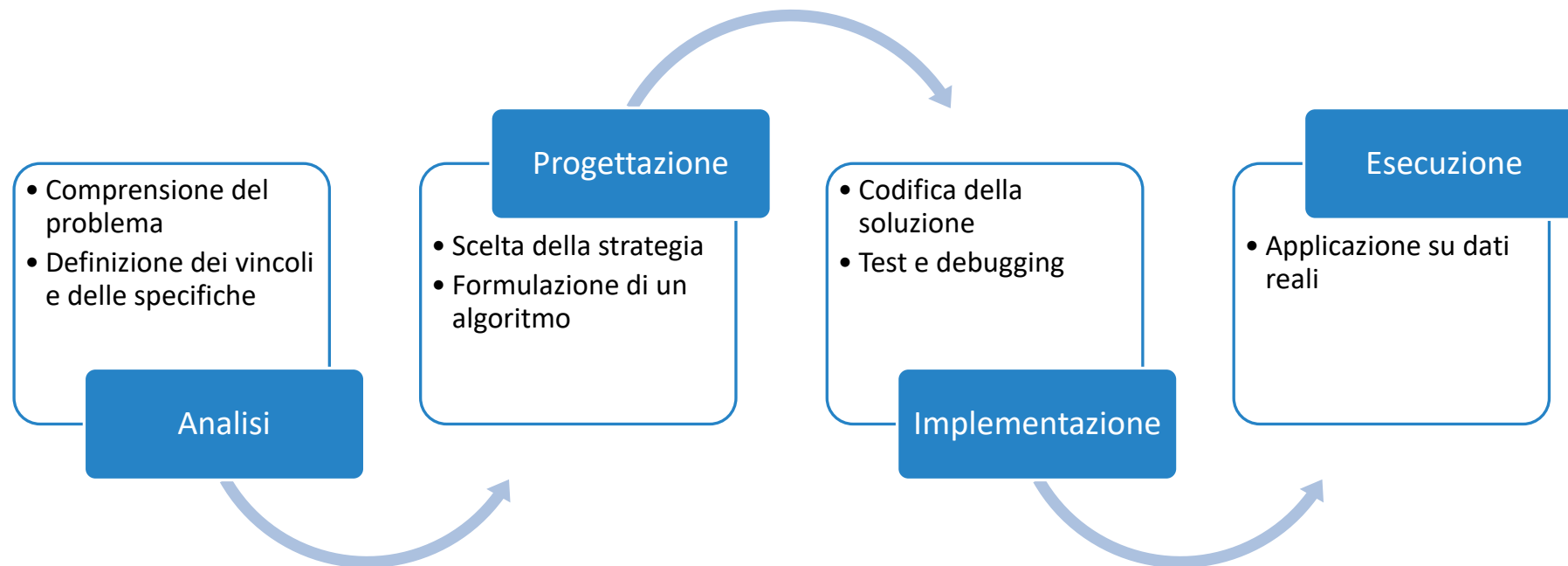
Statistiche studenti

```
import pandas as pd
import seaborn as sns
studenti = pd.read_csv('14BHDOA_2024.csv')
sns.countplot(data=studenti, x='CDS STUDENTE')
```



...e cioè cosa impariamo a fare?

- Quali sono i nomi più frequenti in quest'aula?



Una possibile soluzione... in Python

```
import csv
from matplotlib import pyplot

# Leggi l'elenco degli studenti e salvalo in un'array
def leggi(nome_file):
    file = open(nome_file, 'r')
    reader = csv.reader(file)
    prima = True
    studenti = []
    for line in reader:
        if prima: # skip first line (headers)
            prima = False
        else:
            studenti.append(line)
    file.close()
    return studenti

# estrai i nomi di battesimo da un elenco di studenti
def estrai_nomi(elenco):
    lista_nomi = []
    for riga in elenco:
        lista_nomi.append(riga[2])
    return lista_nomi

# Calcola le frequenze dei vari nomi presenti in un array
def frequenze(tokens):
    freq = {}
    for token in tokens:
        if token in freq:
            freq[token] = freq[token] + 1
        else:
            freq[token] = 1
    return freq
```

```
# calcola il massimo valore presente nelle frequenze
def max_frequenza(freq):
    return max(freq.values())

def nomi_piu_frequenti(freq, max):
    return [nome for (nome, frequenza) in freq.items() if frequenza == max]

FILENAME = '01TXYOYV_2020.csv'
def main():
    stud = leggi(FILENAME)
    nomi = estrai_nomi(stud)
    print(f"Nella classe ci sono {len(stud)} studenti")
    freq = frequenze(nomi)
    max_freq = max_frequenza(freq)
    print(f"il nome più frequente compare {max_freq} volte")
    nomi_max = nomi_piu_frequenti(freq, max_freq)
    print(f"Si tratta di : {nomi_max}")
    # estrai solo i nomi che compaiono almeno 3 volte
    freq2 = {k: v for (k, v) in freq.items() if v >= 3}
    print(
        f"I nomi che compaiono più volte sono {', '.join(sorted(list(freq2.keys())))}."
    )

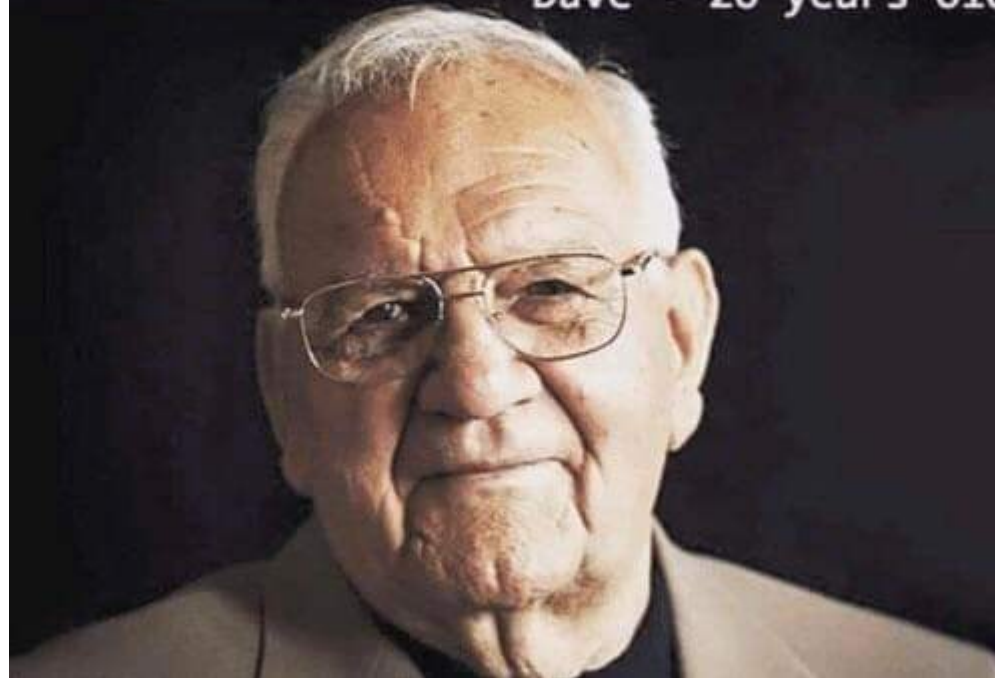
    pyplot.barh(list(freq2.keys()), freq2.values())
    pyplot.show()

main()
```

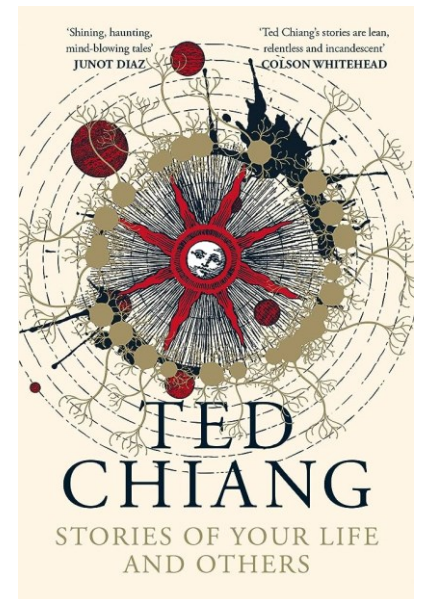
<https://replit.com/@fulcorno/NomiFrequentiStudenti#main.py>

“Programming is
not stressful at all”

Dave - 26 years old



A cosa serve imparare a programmare?



Uno sguardo a Python

VISIONE GENERALE DELL'ECOSISTEMA PYTHON

Il linguaggio Python



- Linguaggio gratuito ed open-source
- Disponibile per tutti i sistemi operativi
 - Windows, Mac OS X, Linux
 - Sistemi embedded, Raspberry PI, Android
- Progettato negli anni '90 da Guido Van Rossum
 - Sintassi semplice, pulita, regolare
 - Approccio «batterie incluse»
 - Ampia libreria di funzioni standard
 - Basso gradino d'accesso
 - Linguaggio interpretato
- Sterminata documentazione on-line



<https://www.python.org/>

The screenshot shows the Python.org homepage with a dark blue header and navigation menu. The main content area features a featured article titled "Functions Defined" with a code snippet for a Fibonacci function. Below the article is a promotional message: "Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)".

```
# Python 3: Fibonacci series up to n
>>> def fib(n):
>>>     a, b = 0, 1
>>>     while a < n:
>>>         print(a, end=' ')
>>>         a, b = b, a+b
>>>     print()
>>> fib(1000)
0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987
```

Functions Defined

The core of extensible programming is defining functions. Python allows mandatory and optional arguments, keyword arguments, and even arbitrary argument lists. [More about defining functions in Python 3](#)

1 2 3 4 5

Python is a programming language that lets you work quickly and integrate systems more effectively. >>> [Learn More](#)

Get Started

Whether you're new to programming or an experienced developer, it's easy to learn and use Python.

[Start with our Beginner's Guide](#)

Download

Python source code and installers are available for download for all versions!

Latest: [Python 3.8.5](#)

Docs

Documentation for Python's standard library, along with tutorials and guides, are available online.

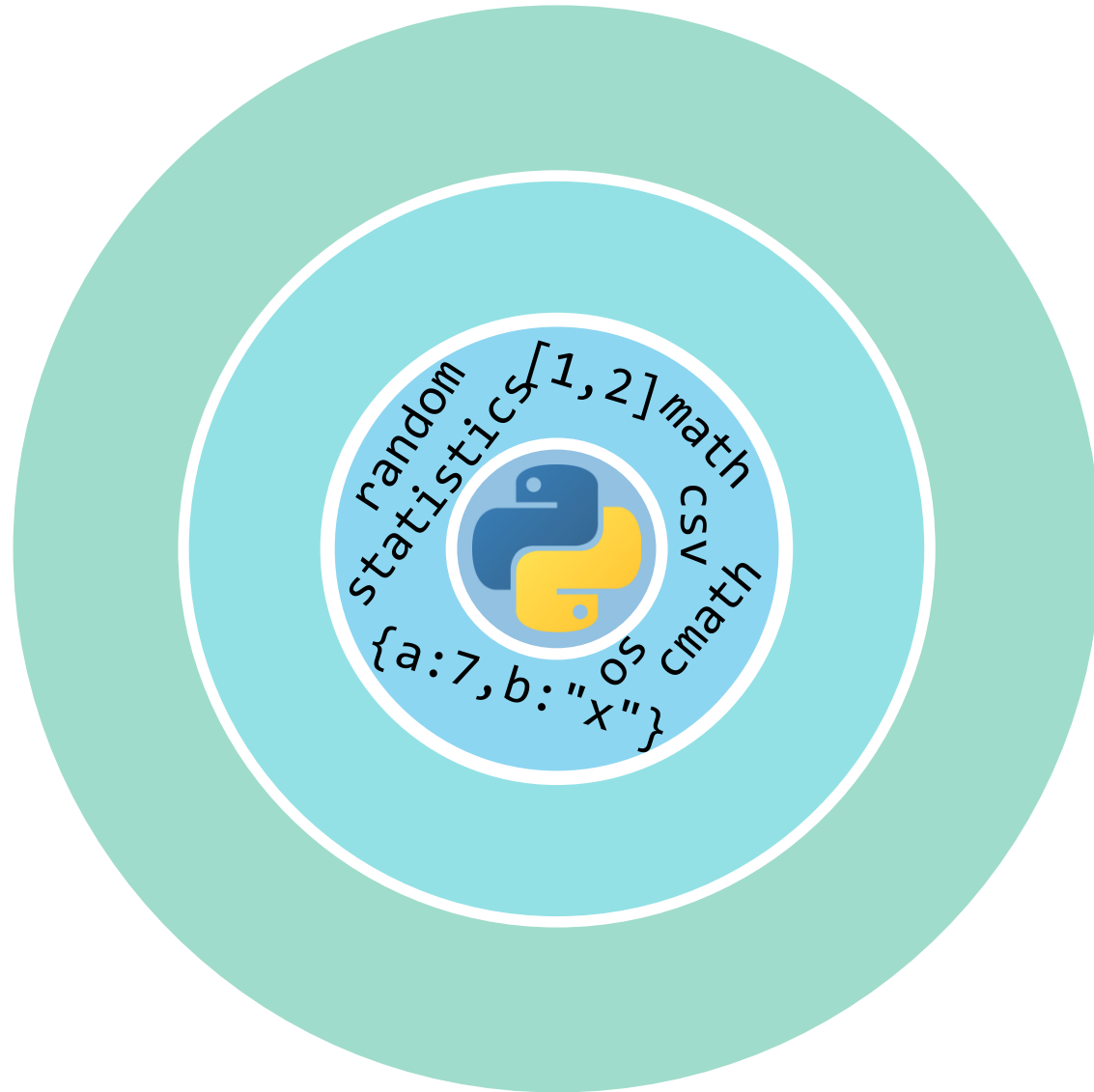
docs.python.org

Jobs

Looking for work or have a Python related position that you're trying to hire for? Our **relaunched community-run job board** is the place to go.

jobs.python.org

Batterie incluse



- **Tipi di dato fondamentali**
 - boolean, int, float, complex, string, regexp
- **Strutture dati fondamentali**
 - liste/array/matrici, tuple, insiemi, dizionari/mappe/hash, file, ...
- **Orientato agli oggetti**
 - Utilizzo semplice e diretto di oggetti predefiniti
 - Possibilità di creare classi ed oggetti personalizzati (avanzato)
- **200+ Moduli nella libreria standard**

200 Moduli della libreria standard

abc	chunk	decimal	getpass	keyword	optparse	queue	sndhdr	telnetlib	unittest
aifc	cmath	difflib	gettext	linecache	os	quopri	socket	tempfile	urllib
argparse	cmd	dis	glob	locale	ossaudiodev (Linux, FreeBSD)	random	socketserver	termios (Unix)	uu
array	codecs	distutils	graphlib	logging	parser	re	spwd (Unix)	test	uuid
ast	codeop	doctest	grp (Unix)	lzma	pathlib	readline (Unix)	sqlite3	textwrap	venv
asynchat	collections	email	gzip	mailbox	pdb	replib	ssl	threading	warnings
asyncio	colorsys	encodings	hashlib	mailcap	pickle	resource (Unix)	stat	time	wave
asyncore	compileall	ensurepip	heapq	marshal	pickletools	rlcompleter	statistics	timeit	weakref
atexit	configparser	enum	hmac	math	pipes (Unix)	runpy	string	tkinter	webbrowser
audioop	contextlib	errno	html	mimetypes	pkgutil	sched	stringprep	token	winreg (Win)
base64	contextvars	faulthandler	http	mmap	platform	secrets	struct	tokenize	winsound (Win)
bdb	copy	fcntl (Unix)	imaplib	modulefinder	plistlib	select	subprocess	trace	wsgiref
binascii	copyreg	filecmp	imghdr	msilib (Windows)	poplib	selectors	sunau	traceback	xdrlib
binhex	crypt (Unix)	fileinput	imp	msvcrt (Windows)	pprint	shelve	symbol	tracemalloc	xml
bisect	csv	fnmatch	importlib	multiprocessing	profile	shlex	symtable	tty (Unix)	xmlrpc
builtins	ctypes	fractions	inspect	netrc	pstats	shutil	sys	turtle	zipapp
bz2	curses (Unix)	ftplib	io	nis (Unix)	pty (Linux)	signal	sysconfig	turtledemo	zipfile
calendar	dataclasses	functools	ipaddress	nntplib	pwd (Unix)	site	syslog (Unix)	types	zipimport
cgi	datetime	gc	itertools	numbers	pyclbr	smtpd	tabnanny	typing	zlib
cgitb	dbm	getopt	json	operator	pydoc	smtplib	tarfile	unicodedata	zoneinfo

Gli ambienti di lavoro



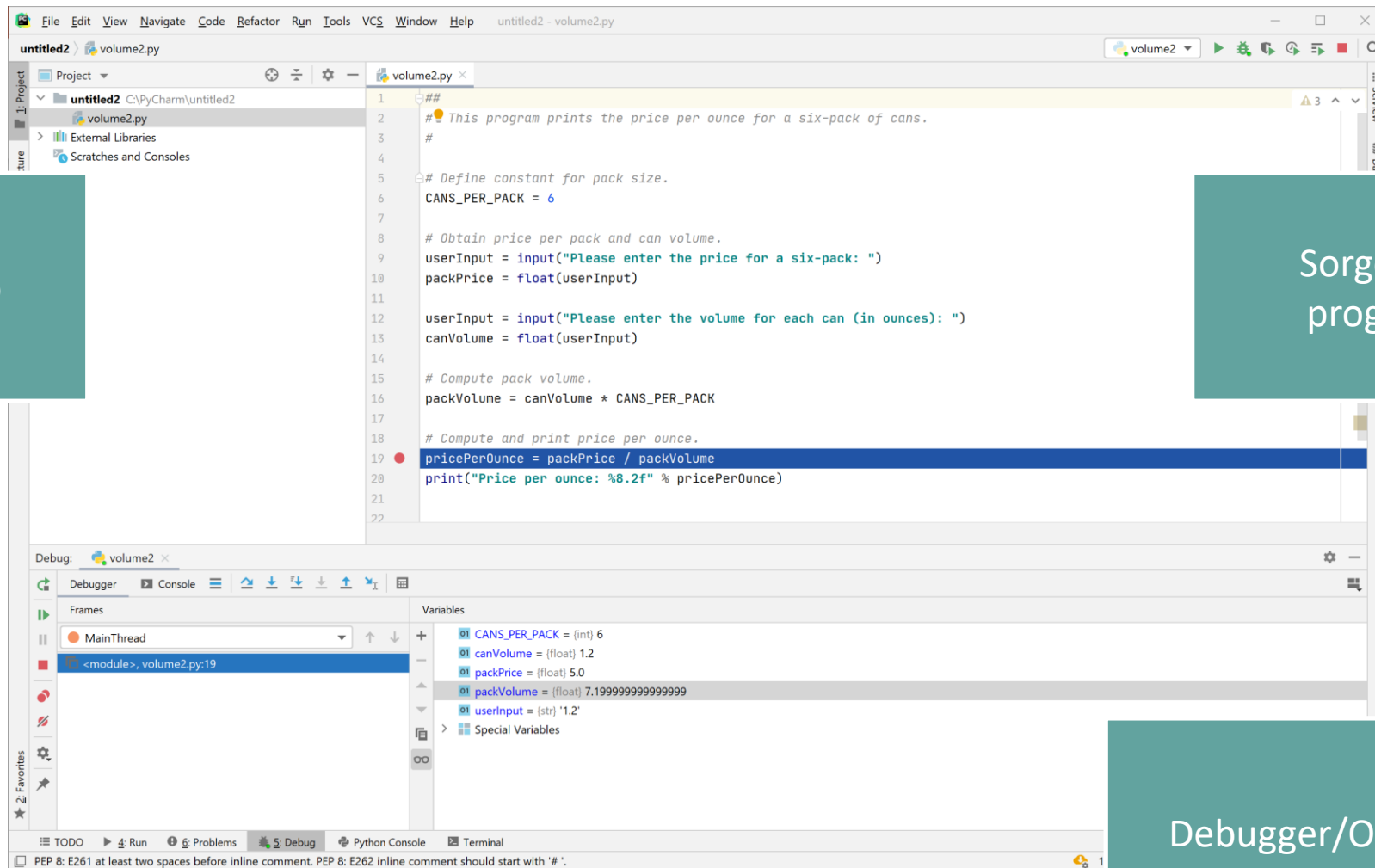
- Ambienti di sviluppo **tradizionali** (IDE)
 - IDLE, PyCharm, Visual Studio Code, Eclipse PyDev, ...
- Ambienti di sviluppo **on-line**
 - Repl.it, PythonAnywhere, Python Tutor
- Ambienti per il calcolo **interattivo**
 - Spyder, IPython
- **Notebook** Computazionali
 - Jupyter, JupyterLab, Google Colab
- Ambienti per **l'apprendimento**
 - Mu, Thonny, Wing

L'IDE di Visual Studio Code

The image displays the Visual Studio Code IDE interface with several components highlighted by green boxes:

- File di progetto:** The Explorer sidebar on the left shows a project structure for 'SETTIMANE' with subfolders for weeks 08 through 14, and files like 'database.txt' and 'dati_aeroporto_torino.csv'.
- Sorgente del programma:** The main editor window shows the source code of 'soluzione.py' with a yellow highlight on line 18: `with open(nome_file, 'r', encoding='utf-8') as f:`.
- Debugger:** The 'VARIABLES' pane on the left shows local variables: `special variables`, `corso: []`, `fermate: []`, and `nome_file: 'dati_aeroporto_torino.csv'`.
- Errori:** The 'PROBLEMS' pane at the bottom shows a `FileNotFoundError` exception: `[Errno 2] No such file or directory: 'dati_aeroporto_torino.csv'`.
- Output:** The 'TERMINAL' pane at the bottom shows the execution output, including PowerShell commands and the error message.

L'IDE di PyCharm



File di progetto

Sorgente del programma

Debugger/Output

IDE On-line : <https://replit.com/>

The screenshot displays the Replit online IDE interface. At the top, the user profile is '@anonymous / GentleGiftedGeneric' and there is a 'Sign up' button. The main workspace is divided into three sections: a file explorer on the left, a code editor in the center, and a terminal on the right. The file explorer shows a file named 'main.py'. The code editor contains the following Python code:

```
1 # My first Python program
2 print("Hello, world")
```

The terminal on the right shows the output of the program: 'Hello, world'. Three blue callout boxes with arrows point to specific elements: 'File di progetto' points to the 'main.py' file in the explorer; 'Sorgente del programma' points to the code in the editor; and 'Output del programma e console interattiva' points to the terminal output. A green rounded rectangle at the bottom contains the following text:

Opzione fantastica per esempi rapidi, per testare frammenti di programma, per evitare di creare un intero Progetto per una piccola parte di codice,, ...

Ambienti scientifici interattivi

SPYDER

Editor - /tmp/interpolation.py

```
4 From the SciPy Cookbook
5 """
6
7 from numpy import arange, cos, linspace, pi, sin, random
8 from scipy.interpolate import splprep, splev
9
10 # make ascending spiral in 3-space
11 t=linspace(0,1.75*2*pi,100)
12
13 x = sin(t)
14 y = cos(t)
15 z = t
16
17 #%% add noise
18 x+= random.normal(scale=0.1, size=x.shape)
19 y+= random.normal(scale=0.1, size=y.shape)
20 z+= random.normal(scale=0.1, size=z.shape)
21
22 #%% spline parameters
23 s=3.0 # smoothness parameter
24 k=2 # spline order
25 nest=-1 # estimate of number of knots needed (-1 = maximal)
26
27 #%% find the knot points
28 tckp,u = splprep([x,y,z],s=s,k=k,nest=-1)
29
30 #%% evaluate spline, including interpolated points
31 xnew,ynew,znew = splev(linspace(0,1,400),tckp)
32
33 import pylab
```

Object Inspector: mean

Definition: mean(a, axis=None, dtype=None, out=None, keepdims=False)

Type: Function of numpy.core.fromnumeric module

Compute the arithmetic mean along the specified axis.

Returns the average of the array elements. The average is

IPython console: Python 3.4.0 on linux -- IPython 4.0.0

In [1]: runfile('/tmp/interpolation.py', wdir='/tmp')

JUPYTERLAB (ANCHE ON-LINE), GOOGLE COLAB

Files: notebooks

Name	Last Modified
Data.ipynb	an hour ago
Fasta.ipynb	a day ago
Julia.ipynb	a day ago
Lorenz.ipynb	seconds ago
R.ipynb	a day ago
iris.csv	a day ago
lightning.json	9 days ago
lorenz.py	3 minutes ago

Code

In this Notebook we explore the Lorenz system of differential equations:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy \end{aligned}$$

Let's call the function once to view the solutions. For this set of parameters, we see the trajectories swirling around two points, called attractors.

In [4]: `from lorenz import solve_lorenz`
`t, x_t = solve_lorenz(N=10)`

Output View

sigma: 10.00
beta: 2.67
rho: 28.00

lorenz.py

```
9 def solve_lorenz(N=10, max_time=4.0, sigma=10.0, beta=8./3, rho=28.0):
10     """Plot a solution to the Lorenz differential equations."""
11     fig = plt.figure()
12     ax = fig.add_axes([0, 0, 1, 1], projection='3d')
13     ax.axis('off')
14
15     # prepare the axes limits
16     ax.set_xlim((-25, 25))
17     ax.set_ylim((-35, 35))
18     ax.set_zlim((5, 55))
19
20     def lorenz_deriv(x,y,z, t0, sigma=sigma, beta=beta, rho=rho):
21         """Compute the time-derivative of a Lorenz system."""
22         x, y, z = x,y,z
23         return [sigma * (y - x), x * (rho - z) - y, x * y - beta * z]
24
25     # Choose random starting points, uniformly distributed from -15 to 15
26     np.random.seed(1)
27     x0 = -15 + 30 * np.random.random((N, 3))
28
```

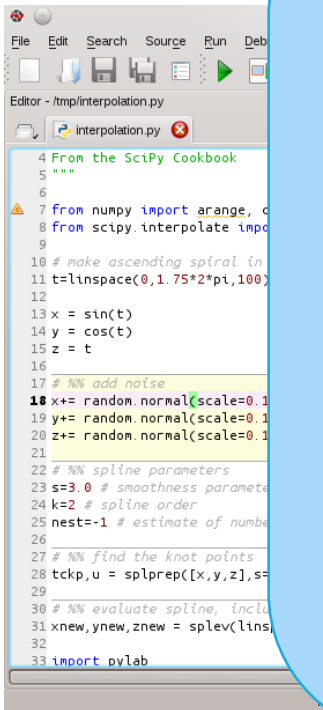
Ambienti scientifici interattivi

SPYDER

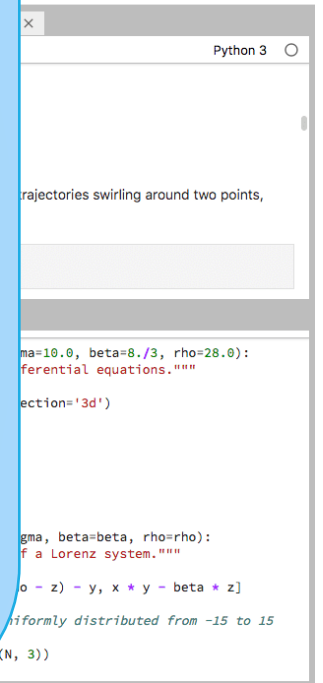
JUPYTERLAB (ANCHE ON-LINE), GOOGLE COLAB

Nuove possibilità...

- Pubblicare on-line esercizi «interattivi» sotto forma di notebook
- Redigere le prime versioni di un articolo, inframmezzando il testo alle formule, con il [ri-calcolo automatico di risultati e grafici



```
4 From the SciPy Cookbook
5 """
6
7 from numpy import arange, cos, sin, linspace
8 from scipy.interpolate import splprep, splev
9
10 # make ascending spiral in the xy plane
11 t=linspace(0,1.75*2*pi,100)
12
13 x = sin(t)
14 y = cos(t)
15 z = t
16
17 #%% add noise
18 x+= random.normal(scale=0.1, size=x.size)
19 y+= random.normal(scale=0.1, size=y.size)
20 z+= random.normal(scale=0.1, size=z.size)
21
22 #%% spline parameters
23 s=3.0 # smoothness parameter
24 k=2 # spline order
25 nest=-1 # estimate of number of knots
26
27 #%% find the knot points
28 tckp,u = splprep([x,y,z],s=s,k=k,nest=nest)
29
30 #%% evaluate spline, including noise
31 xnew,ynew,znew = splev(linspace(0,2*pi,100),tckp,u)
32
33 import pylab
```



```
gamma, beta=beta, rho=rho):
    f a Lorenz system."""
    rho - z) - y, x * y - beta * z]
    uniformly distributed from -15 to 15
    ((N, 3))
```

Librerie per ambiti applicativi



open source
304,797 projects



- Scientific computation
 - NumPy, SciPy, SymPy
- Data Analysis, Algoritmi, Grafi
 - Pandas, networkx, GeoPandas
- Image Processing
 - Pillow, scikit-image, OpenCV
- Visualization
 - Pyviz, matplotlib, plotly, seaborn, altair
- Machine Learning
 - Scikit-learn, tensorflow, pytorch, keras
- Fintech
 - f.fn, zipline, pyalgotrade
- Biology and Genome
 - Biopython
- Fluid Dynamics
 - Fluidity
- Finite Elements
 - Sfepy
- Control systems

Use Python for... >>> More

Web Development: Django, Pyramid, Bottle, Tornado, Flask, web2py

GUI Development: tkinter, PyGObject, PyQt, PySide, Kivy, wxPython

Scientific and Numeric: SciPy, Pandas, IPython

Software Development: Buildbot, Trac, Roundup

System Administration: Ansible, Salt, OpenStack



Singoli moduli



Toolkit completo
per data science

Calcolo scientifico



- NumPy
 - Array, vettori, algebra lineare



- SciPy
 - Package specializzati su diversi ambiti scientifici



- SymPy
 - Calcolo simbolico



- Pandas
 - Analisi e manipolazione dati

Subpackage

`cluster`

`constants`

`fftpack`

`integrate`

`interpolate`

`io`

`linalg`

`ndimage`

`odr`

`optimize`

`signal`

`sparse`

`spatial`

`special`

`stats`

Description

Clustering algorithms

Physical and mathematical constants

Fast Fourier Transform routines

Integration and ordinary differential equation solvers

Interpolation and smoothing splines

Input and Output

Linear algebra

N-dimensional image processing

Orthogonal distance regression

Optimization and root-finding routines

Signal processing

Sparse matrices and associated routines

Spatial data structures and algorithms

Special functions

Statistical distributions and functions

Calcolo scientifico



- NumPy
 - Array, vettori, algebra lineare



- SciPy
 - Package specializzati su diversi ambiti scientifici



- SymPy
 - Calcolo simbolico

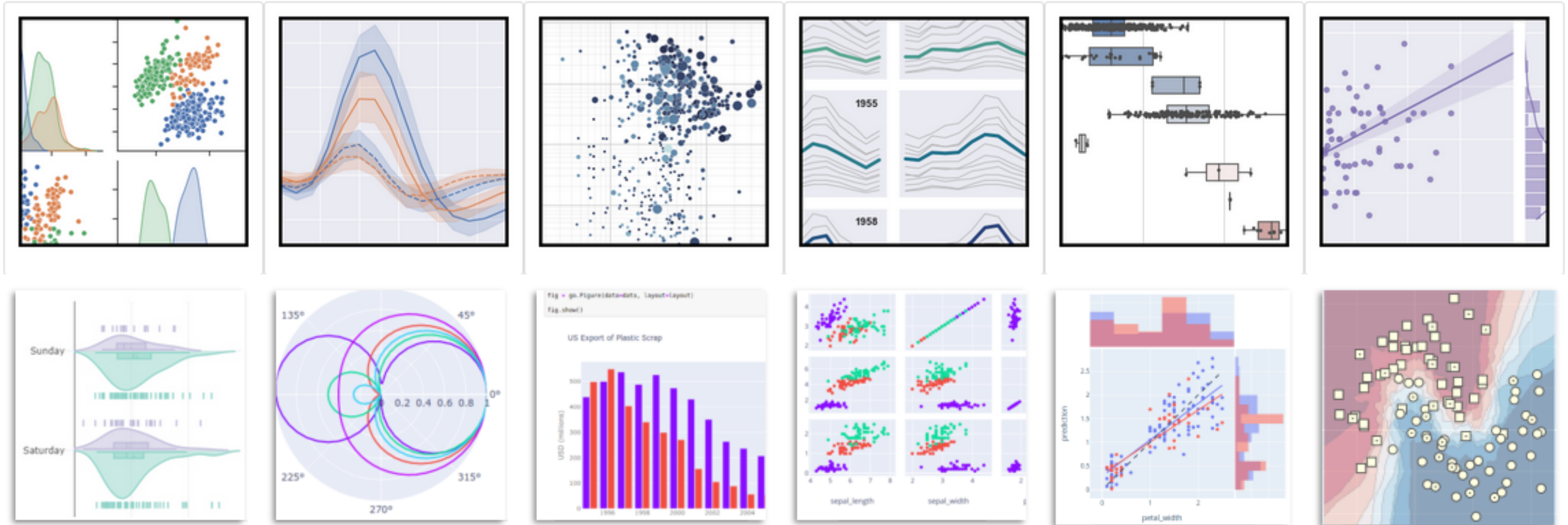
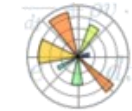


- Pandas
 - Analisi e manipolazione dati

Features

- ✓ Core capabilities
- ✓ Polynomials
- ✓ Calculus
- ✓ Solving equations
- ✓ Combinatorics
- ✓ Discrete math
- ✓ Matrices
- ✓ Geometry
- ✓ Plotting
- ✓ Physics
- ✓ Statistics
- ✓ Cryptography
- ✓ Parsing
- ✓ Printing

Visualizzazione



matplotlib, plotly, seaborn, ...

Esempio: dati ufficiali Covid-19 in real-time

```
import pandas as pd
import seaborn as sns
sns.set_style("whitegrid")

# Leggi dati aggiornati
covid = pd.read_json(
    path_or_buf='https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-json/dpc-covid19-ita-andamento-nazionale.json',
    convert_dates=['data'])

covid.set_index('data', inplace=True)

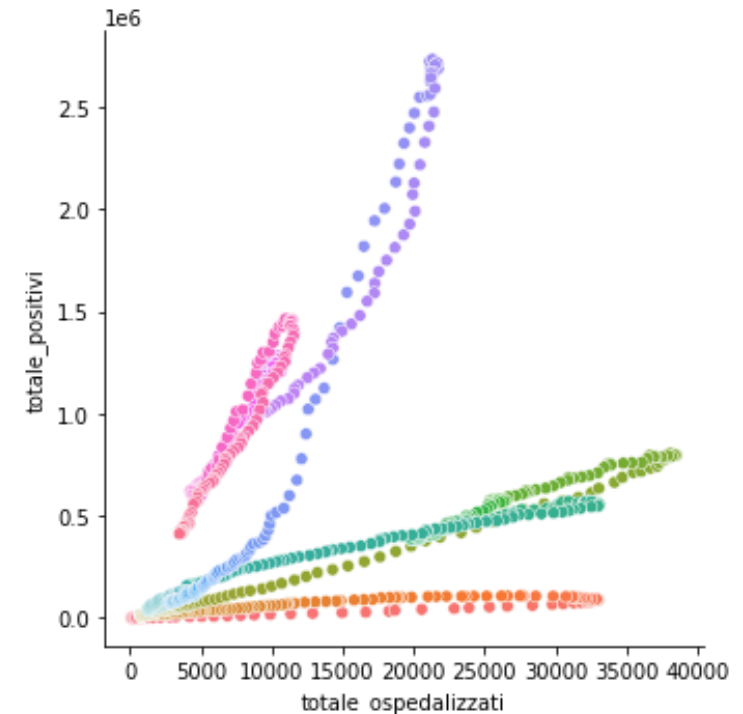
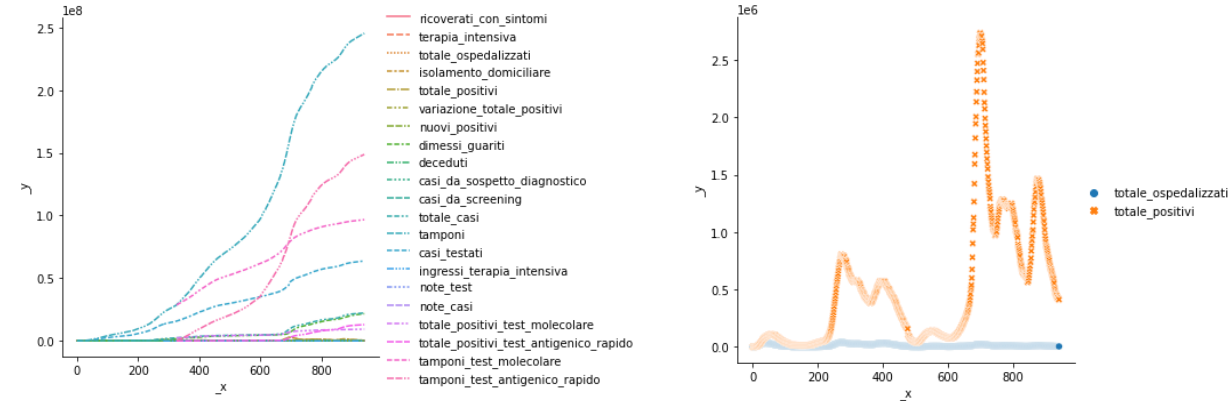
sns.relplot(data=covid, kind='line')

dati_utili = covid[['totale_ospedalizzati', 'totale_positivi']]

sns.relplot(data=dati_utili, kind='line')

sns.relplot(data=dati_utili, kind='scatter',
            x='totale_ospedalizzati', y='totale_positivi', hue='data',
            legend=False)
```

Try me on Google Colab



Homework

- Choose one (quantitative) research question related to your PhD work
 - Concerning experimental data
 - Concerning user interviews
 - Concerning theoretical models
 -
- Write down the research question
- Analyze the question and try to re-write it in the most possible explicit and non-ambiguous way
- <https://polito.padlet.org/fulcorno/ambienti-di-calcolo-e-simulazione-per-la-ricerca-sperimental-x2znwd1ibxngu0n>