



## *il linguaggio*

- linguaggio di programmazione (Guido van Rossum **1989**)
- *insieme a R è il più utilizzato nel **machine learning***
- *l'indentazione è **obbligatoria*** (leggibilità del codice)
- **espandibile** facilmente con librerie esterne
- librerie per big data e machine learning
  - **NumPy**
    - matrici e array multidimensionali
  - **Pandas**
    - caricamento dati e funzioni tabellari
  - **Matplotlib**
    - visualizzazione grafica dei risultati

## *diffusione del linguaggio*

MAJOR COMPANIES  
THAT USE



python

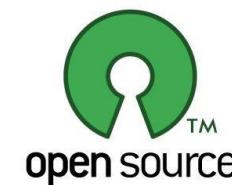
Google NETFLIX facebook. Instagram

amazon Quora # slack intel NASA

Dropbox ebay Spotify Capital One

## *caratteristiche*

- ***free***
  - è possibile utilizzarlo e distribuirlo senza restrizioni di copyright (***open-source***)
- ***facile*** da usare
  - linguaggio di alto livello (semplice e potente)
  - sintassi facile da imparare
- ***portabile***
  - linguaggio interpretato, il codice può essere eseguito su qualsiasi piattaforma purché abbia l'interprete Python installato (Unix, Linux, Windows, macOS, Android, iOS ...)
- ***multi-paradigma***
  - supporta sia la programmazione procedurale, la programmazione ad oggetti e diversi elementi della programmazione funzionale





*Numerical computing tools Python*

# *NumPy*

- *libreria open source* Python (2005 Travis Oliphant)
- aggiunge funzionalità per operare con grandi *matrici e array multidimensionali* con funzioni matematiche di *alto livello*
- il *core* è *ottimizzato* (linguaggio C)
- sito di riferimento <https://numpy.org/>
- documentazione <https://numpy.org/doc/>
- installazione 

```
pip3 install numpy
```

## *perché NumPy*

- *funzioni e metodi* agiscono *globalmente* su vettori e matrici
  - si evitano cicli espliciti (poco efficienti) sui singoli dati
- *algoritmi testati e ottimizzati*
  - gestione della memoria più efficiente che in Python
- più *semplice input/output* su dati di grandi dimensioni



NumPy



Is Numpy  
Faster Than Python?

# *Pandas* *Python for data analysis*



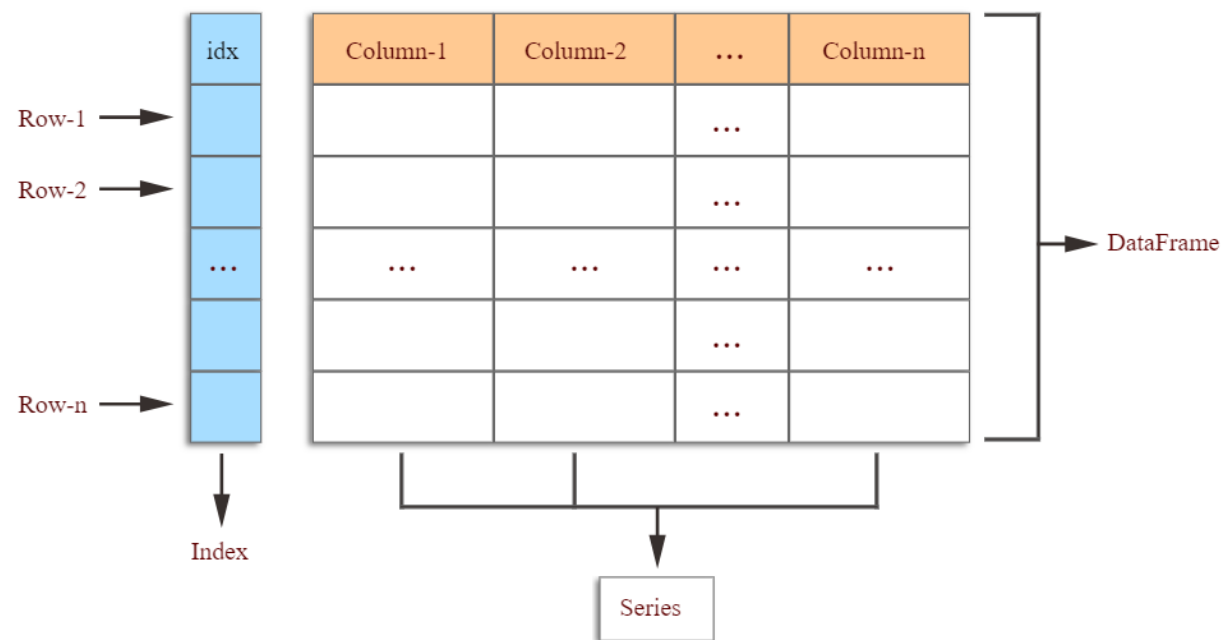


## *Pandas*

- libreria Python che fornisce **strutture** dati di **alto livello** progettate per facilitare e ottimizzare le operazioni sui dati
- le strutture dati sono in formato sequenziale (**Series**) o tabellare (**DataFrame**)
- caratteristiche principali:
  - caricamento e salvataggio di formati standard per dati tabellari
    - CSV (Comma-separated Values), TSV (Tab-separated Values) ...
  - operazioni di indicizzazione e aggregazione di dati semplici e potenti
  - funzioni numeriche e statistiche

## *strutture dati*

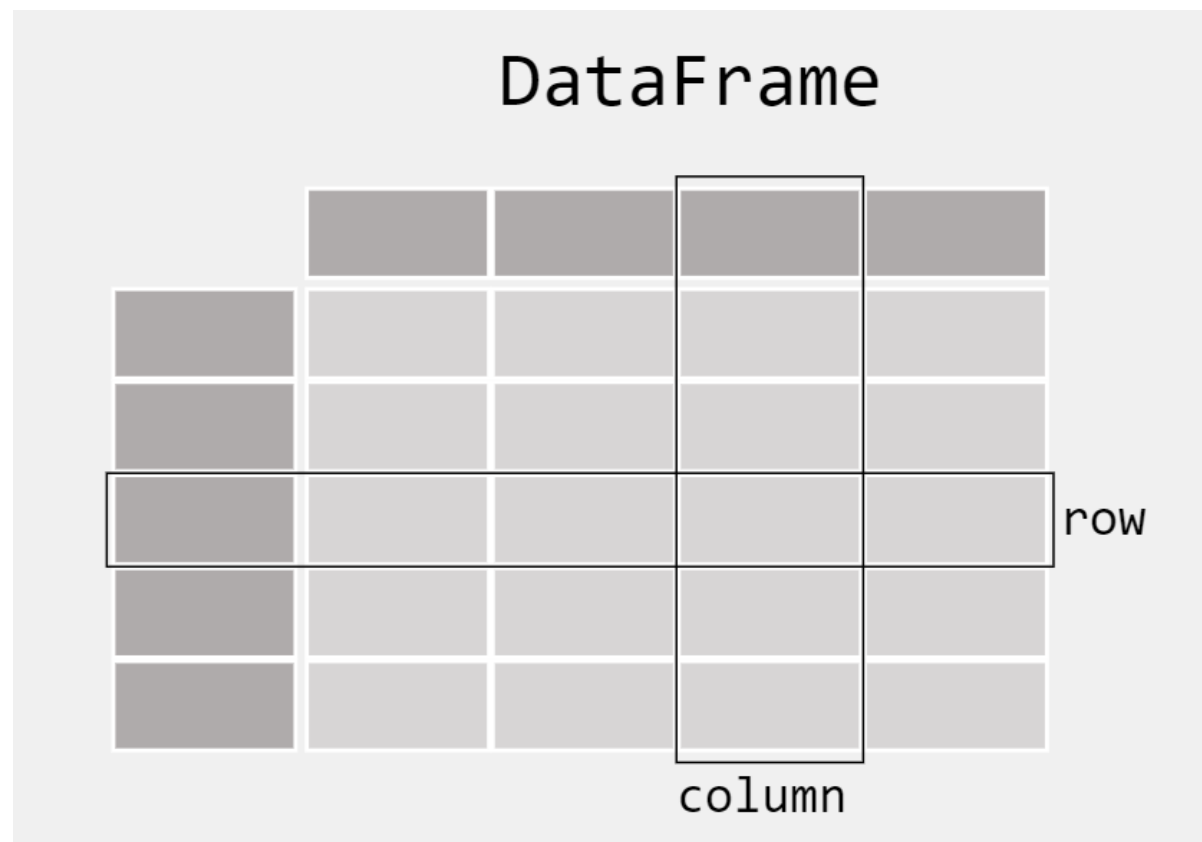
- *strutture dati* fondamentali
  - *Series (1d)*
  - *DataFrame (2d)*
    - ogni colonna di un DataFrame è una Series



## *Series*

- una *Series* è un *vettore mono-dimensionale* i cui elementi sono etichettati con un *index*
- similitudine con liste Python e array di Numpy
  - possibile accedere in sequenza agli elementi
- similitudine con i dizionari Python
  - accesso agli elementi tramite indice

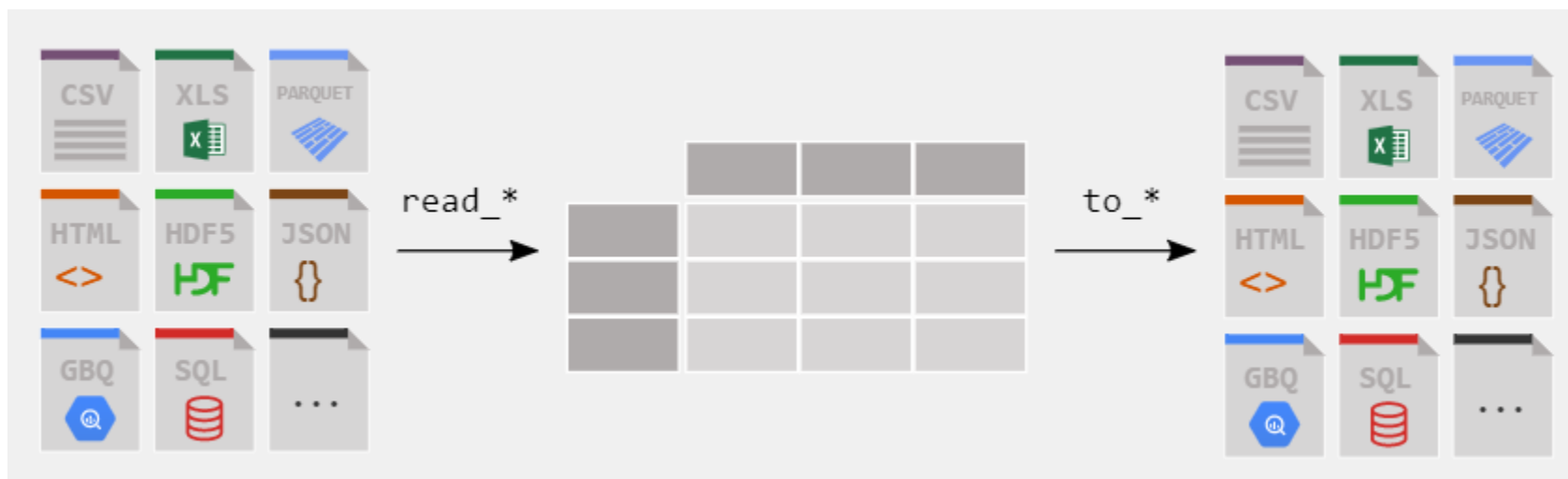
# *DataFrame*



## *DataFrame*

- ***tabella*** di oggetti eterogenei
  - equivalente bidimensionale di una Series
- ***indici*** sia per le righe che per le colonne
  - ***index*** rappresenta le etichette delle righe
  - ***columns*** rappresenta le etichette delle colonne
- l'attributo ***shape*** descrive le dimensioni della tabella
- ogni colonna di un DataFrame è una Series
- tutte le operazioni sulle Series possono essere applicate a colonne estratte da un DataFrame
- molte delle operazioni definite per le Series possono essere applicate direttamente su un DataFrame

## *importazione / esportazione dati*



## *importazione dati*

- l'importazione dei dati è il primo passo in qualsiasi progetto di data science
- i file CSV (Comma Separated Value) sono uno degli standard di interscambio dati fra procedure diverse.
- la funzione `read_csv()` fornisce un modo estremamente semplice per importare i dati da un file CSV a un DataFrame

```
iris = pd.read_csv("iris.csv")
```

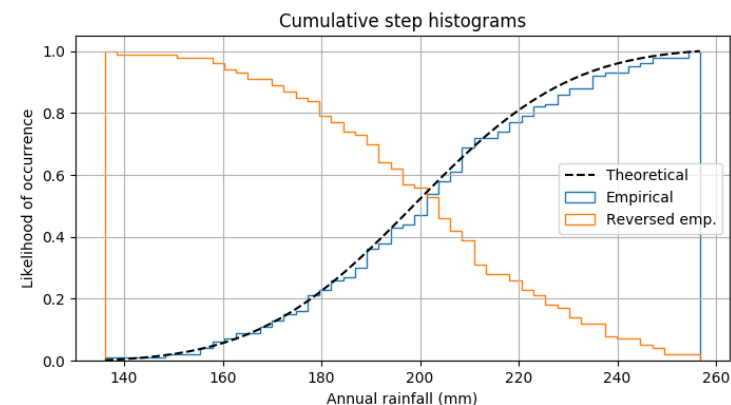
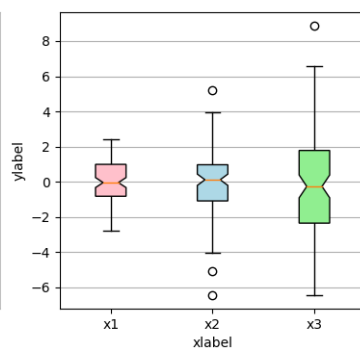
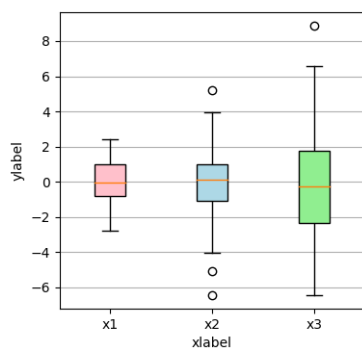
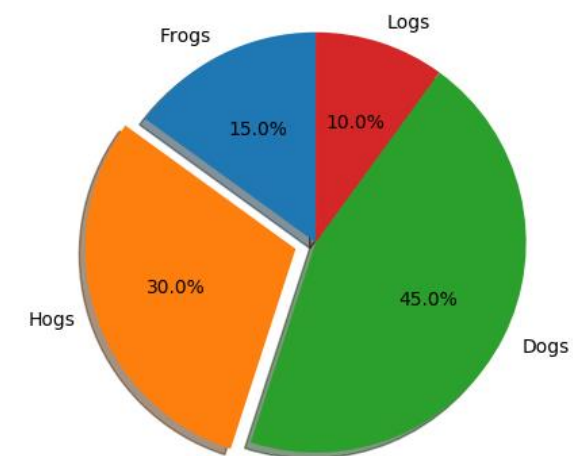
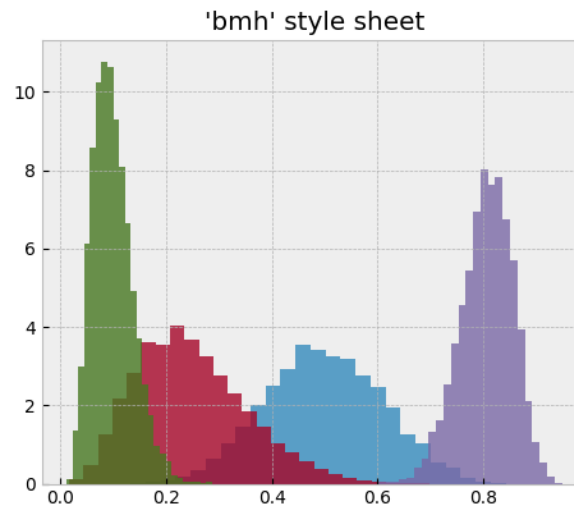
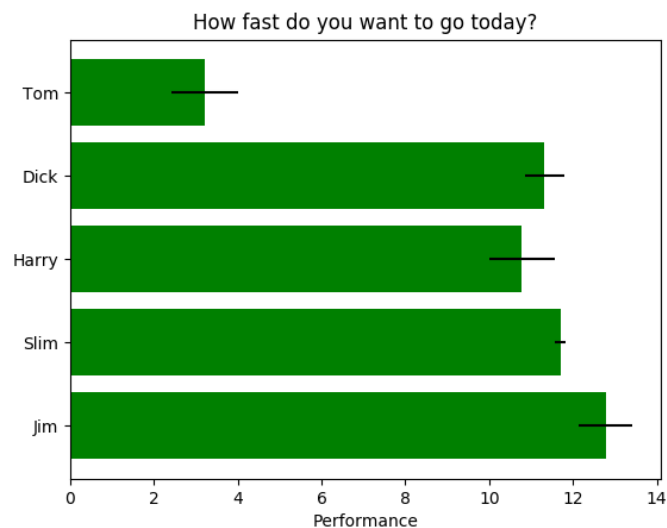
# matplotlib



## *matplotlib*

- libreria per la gestione di *grafici 2d*
- coordinate del grafico definite
  - da un set di valori del dominio della funzione
  - e dai corrispondenti valori calcolati nel codominio
- utilizza *NumPy* per ottimizzare le prestazioni

<https://matplotlib.org/examples/>



## ***referimenti***

- <https://matplotlib.org/>
  - sito ufficiale
- [https://www.w3schools.com/python/matplotlib\\_intro.asp](https://www.w3schools.com/python/matplotlib_intro.asp)
  - esempi interattivi



## *Scikit-learn*

- libreria Python open source di *apprendimento automatico*
- contiene algoritmi di *classificazione, regressione e clustering ...*
- è progettato per operare con le librerie NumPy e SciPy
- <https://scikit-learn.org/>

## *Jupyter Notebook*

- web application open source
- permette di creare e condividere documenti che contengono
  - codice eseguibile
  - testo formattato notazione scientifica (markdown, LaTeX)
  - grafici
- documentazione  
(<https://jupyter-notebook.readthedocs.io/en/stable/>)
- esecuzione dal prompt dei comandi
  - `jupyter notebook`



## *python install*

- **Python** – (<https://www.python.org/>)
  - Python 3.x ...
  - download ... install for all users ... add Python to environment variables
- ***prompt dei comandi***
  - (Windows) **cmd** – (Linux) Applicazioni > Accessori > Terminale – (Mac) Terminale
- **NumPy** (*Numeric Python*)
  - `pip3 install numpy`
- **Pandas** (*Python Data Analysis*)
  - `pip3 install pandas`
- **Matplotlib**
  - `pip3 install matplotlib`
- **Jupyter Notebook**
  - `pip3 install notebook`
- **Scikit-learn**
  - `pip3 install scikit-learn`

