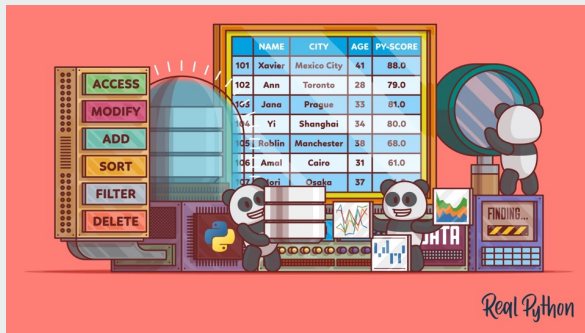


30 minuti con... Pandas



Corso di Quality Outsourcing Management
Lezione 3 di 3

Roberto Nai (Dipartimento di Informatica – UNITO)



Agenda

- Introduzione
- Correlazione tra dati
- Grafici
- Massimi e minimi
- Valori distinti
- Conclusioni



Materiale della lezione: https://github.com/roberto-nai/SUISS_22-23

Pandas, dove eravamo rimasti?

- Nella lezione precedente si è visto come *pulire* un **dataset** (insieme di dati) in formato CSV (Comma-Separated Value) tramite vari metodi della libreria Pandas.
- La pulizia dei dati (data cleaning) è un'attività molto importante, perché determina la qualità dei dati che verranno analizzati.



<https://pandas.pydata.org>

Pandas, trova le relazioni

- Un aspetto importante della libreria Pandas è il metodo `corr()`.
- Il metodo `corr()` calcola la relazione tra ciascuna colonna del dataset.
- Il metodo `corr()` ignora le colonne *non numeriche*.



Maggiore è il numero di dati (righe nel dataset), maggiore sarà la precisione del metodo `corr()`.

Pandas, trova le relazioni

```
df.corr()
```

	durata	pulsazione	pulsazione_max	calorie
durata	1.000000	-0.162098	0.003578	0.923053
pulsazione	-0.162098	1.000000	0.787035	0.015301
pulsazione_max	0.003578	0.787035	1.000000	0.195309
calorie	0.923053	0.015301	0.195309	1.000000

Pandas, trova le relazioni

- Il risultato del metodo `corr()` è una tabella con molti numeri che rappresentano la relazione tra due colonne.
- Il valore della relazione varia da -1 a 1.
- 1 significa che c'è una relazione 1 a 1 (una correlazione perfetta) e per questo dataset, ogni volta che un valore aumenta nella prima colonna, anche nell'altra aumenta.
 - Anche 0,9 è una buona relazione e se si aumenta un valore, probabilmente aumenterà anche l'altro.
- Il valore -0,9 è una relazione altrettanto buona ed indica che, se si aumenta un valore, probabilmente l'altro scenderà.
- Il valore 0,2 non è una buona relazione, nel senso che se un valore sale non significa che lo farà anche l'altro.

Pandas, trova le relazioni



Che cos'è una **buona correlazione**? Dipende, ma si può affermare che un valore di almeno 0,6 (o -0,6) indica una buona correlazione.



Gli **esperti del dominio** (medici, avvocati, professionisti, ecc.) possono *validare* le correlazioni automatiche trovate da Pandas.

Pandas, trova le relazioni

- Osservando le correlazioni, la "durata" e le "calorie" hanno ottenuto un'alta correlazione di 0,92; si può prevedere che più a lungo ci si allena, più calorie si bruciano, e viceversa: se si bruciano molte calorie, probabilmente l'allenamento è stato duraturo.

```
df.corr()
```

	durata	pulsazione	pulsazione_max	calorie
durata				0.923053
pulsazione	-0.162098	1.000000	0.787035	0.015301
pulsazione_max	0.003578	0.787035	1.000000	0.195309
calorie	0.923053	0.015301	0.195309	1.000000

Pandas, disegna

- Utilizzando la libreria `Pyplot`, un sottomodulo della libreria `Matplotlib`, è possibile creare e visualizzare grafici sullo schermo legati ai dati contenuti nel dataset.
- Per prima cosa, importare la libreria:

```
import matplotlib.pyplot as plt
```

```
# Importa la libreria pyplot contenuta in matplotlib e le associa l'alias plt  
import matplotlib.pyplot as plt
```

Pandas, disegna

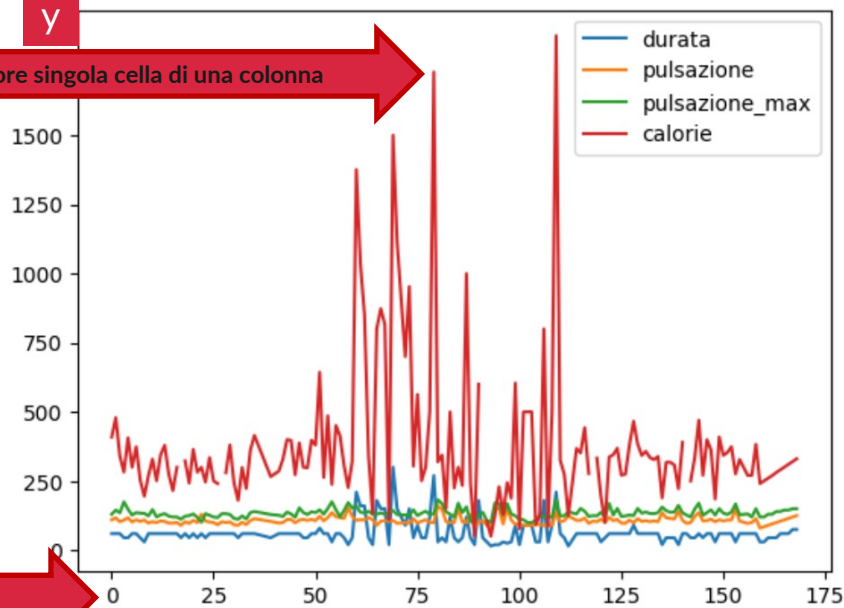
- Pandas utilizza il metodo `plot()` per creare diagrammi.
- Il metodo `plot` distribuisce sull'asse `y` i valori delle celle di ogni singola colonna, e sulla `x` i valori dei numeri di riga (1, 2, 3...).

Pandas, disegna

```
# Crea il grafico dei valori di ogni colonna e lo visualizza con il metodo show()  
df.plot()  
plt.show()
```

y

Valore singola cella di una colonna



Indice di riga

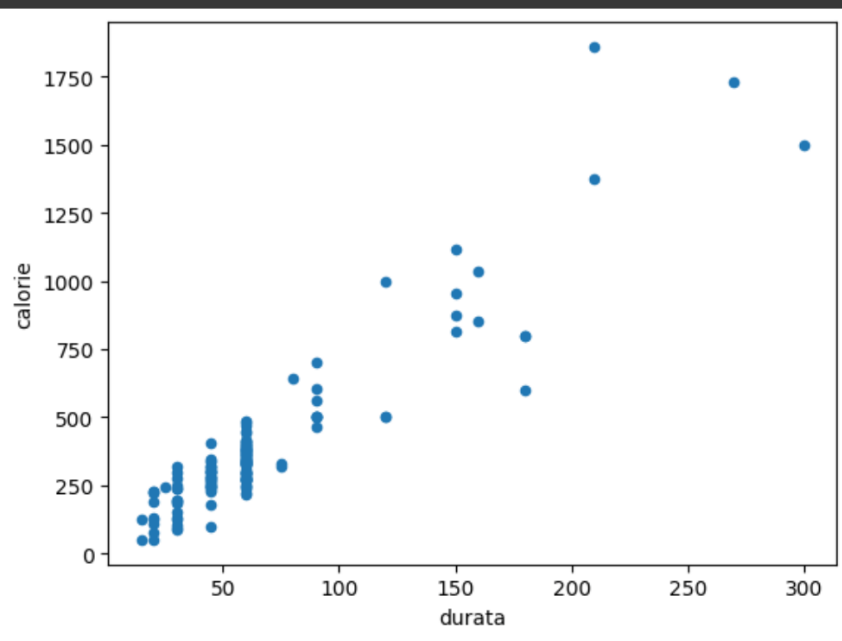
x

Pandas, disegna

- Un secondo tipo di grafico è la *dispersione* (*scatter plot* o *nuvola di punti*) in cui *due* variabili di un dataset sono riportate su uno spazio cartesiano.
- Ad esempio, si può visualizzare il consumo di calorie in base al tempo di corsa.
 - `df.plot(kind='scatter', x='durata', y='calorie')`

Pandas, disegna

```
# Crea il grafico a dispersione (scatter) delle calorie (y) rispetto alla durata  
df.plot(kind = 'scatter', x = 'durata', y = 'calorie')  
plt.show()
```

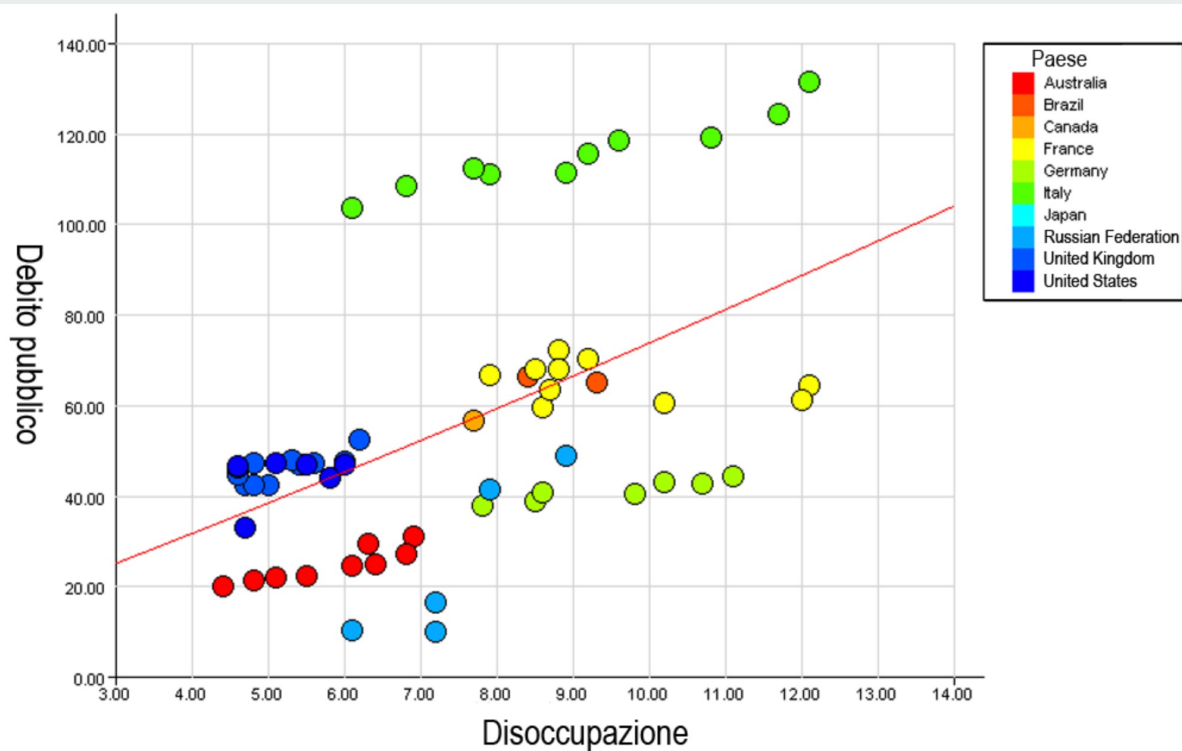


Pandas, disegna



- Un possibile esempio dell'uso del grafico a dispersione è l'analisi dell'andamento delle due seguenti variabili: il debito pubblico e la percentuale di disoccupazione di un paese.
- Avendo due variabili, è necessario decidere quale rappresentare sull'asse delle ascisse (o x) e quale sull'asse delle ordinate (y).
- Non vi è una soluzione corretta o sbagliata, solitamente la variabile più importante è sull'asse delle y , quindi se fosse necessario mostrare quanto varia il debito pubblico in relazione alla disoccupazione si porrà quest'ultima sull'asse x , viceversa ponendo la disoccupazione sull'asse y verrà evidenziato come essa varia in relazione al debito pubblico.

Pandas, disegna



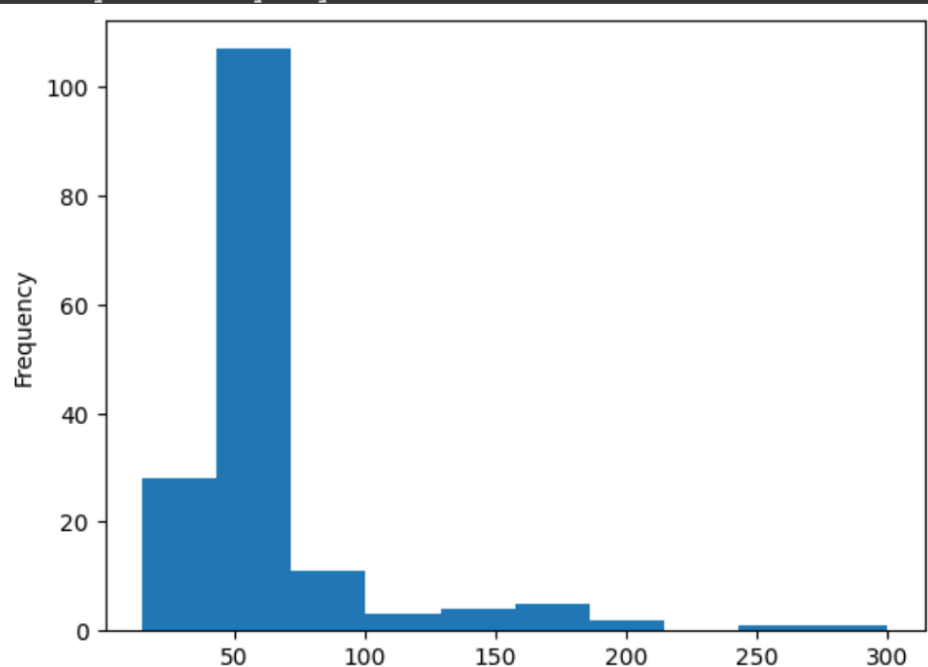
Pandas, disegna

- Un terzo tipo di grafico utile è l'istogramma, per disegnare la frequenza degli intervalli di valori di una colonna (es.: quanti allenamenti sono durati tra i 50 e i 60 minuti).
- Un istogramma viene eseguito su una sola colonna.
- `df['durata'].plot(kind = 'hist')`

Pandas, disegna

```
df['durata'].plot(kind = 'hist')
```

<Axes: ylabel='Frequency'>



Pandas, dimmi massimi e minimi

- Per ogni colonna è possibile stabilire il valore massimo e minimo attraverso, rispettivamente, i metodi `max()` e `min()`.

```
# Massimo di una colonna  
durata_massima = df['durata'].max()  
durata_massima
```

```
300
```

```
# Minimo di una colonna  
durata_minima = df['durata'].min()  
durata_minima
```

```
15
```

Pandas, dimmi i valori distinti

- Per ogni colonna è possibile ottenere la lista dei valori *distinti* (unici) e la loro quantità (es.: {A, A, A, B, B} ha 2 valori distinti A e B) attraverso i comandi `unique()` e `nunique()`.

Per la colonna
durata sono
presenti 16 valori
distinti

```
# Valori distinti (unici) di una colonna
df['durata'].unique()

array([ 60,  45,  30,  80,  20, 210, 160, 180, 150, 300,  90, 120, 270,
        15,  25,  75])

df['durata'].nunique()

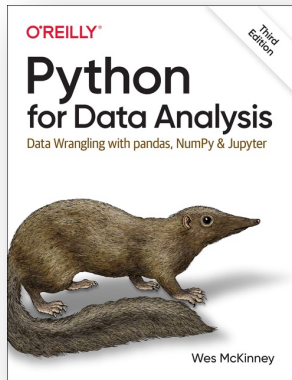
16
```

Conclusioni

- In questa lezione si è visto come è possibile mettere in correlazione tra loro i valori di un dataset.
- Oltre alla correlazione, con Pandas è possibile creare grafici di vario tipo sui dati presenti in una o più colonne.
- Esistono poi vari metodi che permettono di conoscere il valore massimo e minimo di una colonna, nonché i suoi valori distinti.

Bibliografia

- Python for Data Analysis: Data Wrangling With Pandas, Numpy, and Jupyter, 3a edizione, Wes McKinney, O'Reilly.



Bibliografia

- Pandas – metodo `corr()`
 - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.corr.html>
- Pandas – metodo `plot()`
 - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html>
- Pandas – metodo `max()`
 - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.max.html>
- Pandas – metodo `min()`
 - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.min.html>
- Pandas – metodo `unique()`
 - <https://pandas.pydata.org/docs/reference/api/pandas.unique.html>
- Pandas – metodo `nunique()`
 - <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.nunique.html>

Bibliografia

- Python
 - <https://www.python.org>
- SublimeText (programma per sviluppare codice in Python)
 - <https://www.sublimetext.com>
- Visual Studio Code (programma per sviluppare codice in Python)
 - <https://code.visualstudio.com>
- Google Colaboratory (Colab):
 - <https://colab.research.google.com>

Fine presentazione

Grazie per l'attenzione



roberto.nai@unito.it