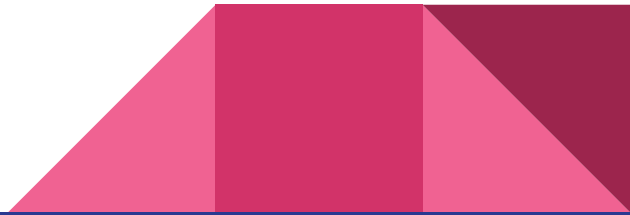
08. Pandas

Corso di Python per il Calcolo Scientifico

Angelo Cardellicchio
angelo.cardellicchio@stiima.cnr.it

Outline

- Dataset e dataframe
- Lettura ed accesso agli elementi
- Aggiunta di feature e dati
- Descrivere e visualizzare i dati



Dataset e dataframe

- Pandas utilizza delle strutture chiamate **dataframe** per rappresentare i dati presenti all'interno di un **dataset**
- I dataset contenggono una serie di **campioni** descritti da una o più **feature**

Feature

Campioni

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Lettura ed accesso agli elementi (1)

- Per **leggere un dataset** contenuto in un file CSV:

```
titanic = pd.read_csv('titanic.csv')
```

- Possiamo anche usare altri formati, come ad esempio Excel o JSON:

```
titanic_xlsx = pd.read_excel('titanic.xlsx')
titanic_json = pd.read_json('titanic.json')
```

- Per **salvare un dataframe** in un file di un dato formato (ad esempio CSV) :

```
titanic = df.to_csv('titanic.csv')
```

Lettura ed accesso agli elementi (2)

- Impostiamo una feature come l'**indice** utilizzato per il dataframe:

```
titanic.set_index('Ticket', inplace=True)
```

- Accediamo ad un campione in base al suo indice numerico:

```
titanic.iloc[i, :]
```

- Possiamo accedere mediante indice e nome di colonna:

```
titanic.loc['STON/O2. 3101282', 'Name']
```

- Proviamo ad usare una maschera booleana:

```
men = df[(df['Age'] > 18) & (df['Sex'] == 'male')]
```

Aggiunta di feature e dati

- Per aggiungere una feature ad un dataframe già esistente:

```
df = pd.DataFrame([1,2,3,4,5], columns=['one’])
df['two'] = df['one'] * 2
```

- Per concatenare due dataframe per righe:

```
df_add = pd.DataFrame([[6,7]], columns=['one', 'two'])
df = pd.concat([df, df_add])
```

- Agendo sul parametro **`axis`**, si concatena lungo quel determinato asse:

```
pd.concat([df, df_add], axis=1)
```

Descrivere e visualizzare i dati

- Per visualizzare una o più feature, possiamo usare la funzione **`plot()`**:

```
df['Age'].plot()
```

- È anche possibile plottare l'intero dataframe:

```
df.plot()
```

- Ancora, possiamo plottare un istogramma:

```
df['Age'].plot().hist()
```

- Usando la funzione **`describe()`** abbiamo una descrizione statistica del dataframe:

```
df.describe()
```

Domande?

42