Funzioni

🔗 Merge & Join

- `pd.merge(df1, df2, on="col", how="left")` → unisce DataFrame su chiave comune.
- `df.join()` → join basata sugli indici (meno usata qui).

🧹 Pulizia dati

- `df.drop_duplicates(subset=["col"])` → elimina duplicati.
- `pd.to_datetime(df["col"], errors="coerce")` → conversione in datetime.
- `df.dropna(subset=["col"])` → elimina righe con NaN in una colonna.
- `df["col"].fillna(0)` → sostituisce NaN con un valore.
- `df.isna().sum()` → conta i valori mancanti per colonna.

📊 Aggregazioni e raggruppamenti

- `df.groupby(["col1","col2"])` → raggruppa per più colonne.
- `.agg({...})` → applica più funzioni di aggregazione.
- `.mean()`, `.median()`, `.std()`, `.count()` → statistiche di base.
- `.sort_values(by="col")` → ordina un DataFrame per valori.

🚨 Outlier detection (IQR)

- `df["col"].quantile(0.25)` → primo quartile (Q1).
- `df["col"].quantile(0.75)` → terzo quartile (Q3).
- `groupby().transform(lambda s: ...)` → calcola Q1/Q3 per ogni gruppo.

🏆 Ranking & Normalizzazione

- Operazioni vettoriali:
  - `(s - s.min()) / (s.max() - s.min())` → normalizzazione min-max.
- `groupby().transform("min")` e `"max"` → min/max per gruppo.
- `df.sort_values("col", ascending=False)` → ordinamento per punteggio.

- `.head(10)` → primi 10 elementi.

📤 Esportazione

- `with pd.ExcelWriter("file.xlsx") as writer:` → scrive più fogli Excel.
- `df.to_excel(writer, sheet_name="Nome", index=False)` → esporta in Excel.
- `df.to_csv("file.csv", index=False)` → esporta in CSV.

ℹ️ Metodi utili per debug

- `df.head()` → prime 5 righe.
- `df.info()` → tipi e valori nulli.
- `df.describe()` → statistiche descrittive.

🔢 Funzioni NumPy

- `np.nan` → valore mancante.
- `np.issubdtype(df["col"].dtype, np.datetime64)` → check tipo datetime.
- Operazioni matematiche di base (sottrazioni, divisioni) usate per normalizzazione e calcolo IQR.

🔗 Merge & Join

```
import pandas as pd

df1 = pd.DataFrame({"id":[1,2], "nome":["Anna","Luca"]})
df2 = pd.DataFrame({"id":[1,2], "stipendio":[30000,40000]})

df = pd.merge(df1, df2, on="id", how="left")
print(df)
# id  nome   stipendio
# 1   Anna   30000
# 2   Luca   40000
```

🧹 Pulizia dati

```
import pandas as pd, numpy as np

df = pd.DataFrame({
    "id":[1,1,2],
    "data":["2020-01-01","2020-01-01","2021-05-10"],
    "bonus":[np.nan, 1000, 2000]
})

# Rimuovi duplicati
print(df.drop_duplicates(subset=["id"]))

# Converti in datetime
df["data"] = pd.to_datetime(df["data"], errors="coerce")

# Rimuovi righe con NaN in 'bonus'
print(df.dropna(subset=["bonus"]))

# Sostituisci NaN con 0
df["bonus"] = df["bonus"].fillna(0)

# Conta NaN
print(df.isna().sum())
```

📊 Aggregazioni e raggruppamenti

```
df = pd.DataFrame({
    "dept":["HR","HR","Eng","Eng"],
    "role":["Analyst","Manager","Dev","Dev"],
    "salary":[30000,40000,50000,60000]
})

agg = df.groupby(["dept","role"]).agg(
    emp_count=("salary","count"),
    mean_salary=("salary","mean"),
    std_salary=("salary","std")
```

```
)
print(agg)
```

🚨 Outlier detection (IQR)

```
df = pd.DataFrame({
    "dept":["HR","HR","HR","HR"],
    "total_comp":[30000,32000,31000,80000]
})

q1 = df.groupby("dept")["total_comp"].transform(lambda s: s.quantile(0.25))
q3 = df.groupby("dept")["total_comp"].transform(lambda s: s.quantile(0.75))
iqr = q3 - q1
lower = q1 - 1.5*iqr
upper = q3 + 1.5*iqr

df["is_outlier"] = (df["total_comp"] < lower) | (df["total_comp"] > upper)
print(df)
```

🏆 Ranking & Normalizzazione

```
df = pd.DataFrame({
    "dept":["HR","HR","HR"],
    "rating":[3.5,4.0,5.0],
    "goals":[6,7,10]
})

# Normalizzazione min-max PER reparto
df["rating_norm"] = (df["rating"] - df.groupby("dept")["rating"].transform("min")) / (
    df.groupby("dept")["rating"].transform("max") - df.groupby("dept")["rating"].transform("min")
)
```

```
df["goals_norm"] = (df["goals"] - df.groupby("dept")["goals"].transform("min")) / (
    df.groupby("dept")["goals"].transform("max") - df.groupby("dept")["goals"].transform("min")
)

df["perf_score"] = 0.7*df["rating_norm"] + 0.3*df["goals_norm"]

# Ranking globale
print(df.sort_values("perf_score", ascending=False).head(2))
```

📤 Esportazione

```
df = pd.DataFrame({"id":[1,2], "score":[95,88]})

# Esporta Excel con più fogli
with pd.ExcelWriter("report.xlsx") as writer:
    df.to_excel(writer, sheet_name="Foglio1", index=False)
    df.describe().to_excel(writer, sheet_name="Statistiche")

# Esporta CSV
df.to_csv("outliers.csv", index=False)
```

ℹ️ Metodi utili per debug

```
df = pd.DataFrame({"A":[1,2,3,4,5], "B":[10,20,30,40,50]})

print(df.head())     # prime 5 righe
print(df.info())     # tipi e null
print(df.describe()) # statistiche
```

🔢 NumPy base

```
import numpy as np
```

```
# NaN
val = np.nan

# Controllo tipo datetime in Pandas
import pandas as pd
df = pd.DataFrame({"date":["2020-01-01"]})
df["date"] = pd.to_datetime(df["date"])
print(np.issubdtype(df["date"].dtype, np.datetime64))  # True
```