Dati sintetici con supporto dati Istat

simulazione deterministica con rumore gaussiano

```
import pandas as pd
import numpy as np
import random
import os
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import MinMaxScaler
import uuid
import pyarrow as pa
import pyarrow.parquet as pq


# Impostazioni
np.random.seed(42)
N = 1000

classi = ['Micro', 'Piccola', 'Media', 'Grande']
settori = ['Industria', 'Servizi']
mercati = ['Interno', 'Estero']

# Quote per ciascuna classe (25% ciascuna)
quote_classe = [0.25, 0.25, 0.25, 0.25]

# Generazione dati di base
data = []

for cls, quota in zip(classi, quote_classe):
    n_cls = int(N * quota)
```

```
    for _ in range(n_cls):
        settore = random.choice(settori)
        mercato = 'Interno' if cls in ['Micro', 'Piccola'] else ('Estero' if cls == 'Grande'

        fatt_2024 = np.random.uniform(100_000, 10_000_000)
        volume_2024 = np.random.uniform(1_000, 500_000)
        prezzo_2024 = fatt_2024 / volume_2024

        # Parametri medi per variazioni
        param = {
            'Micro':   {'vol': 0.5, 'prez': 0.5, 'fatt': 1.0},
            'Piccola': {'vol': 0.7, 'prez': 0.8, 'fatt': 1.5},
            'Media':   {'vol': 0.9, 'prez': 1.0, 'fatt': 2.0},
            'Grande':  {'vol': 1.2, 'prez': 1.5, 'fatt': 2.7},
        }
        noise = lambda m: np.random.normal(loc=m, scale=1.0)

        delta_vol = noise(param[cls]['vol'])
        delta_prez = noise(param[cls]['prez'])
        delta_fatt = noise(param[cls]['fatt'])

        if settore == 'Servizi':
            delta_prez += 0.6
            delta_fatt += 0.6

        volume_2025 = volume_2024 * (1 + delta_vol / 100)
        prezzo_2025 = prezzo_2024 * (1 + delta_prez / 100)
        fatt_2025 = volume_2025 * prezzo_2025  # coerente

        data.append({
            'ID': str(uuid.uuid4()),
            'Classe': cls,
            'Settore': settore,
            'Mercato': mercato,
            'Fatturato_2024': round(fatt_2024, 2),
            'Volume_2024': round(volume_2024, 2),
            'Prezzo_2024': round(prezzo_2024, 4),
            'Fatturato_2025': round(fatt_2025, 2),
            'Volume_2025': round(volume_2025, 2),
            'Prezzo_2025': round(prezzo_2025, 4),
        })
```

```
df = pd.DataFrame(data)
```

```
df
```

	ID	Classe	Settore	Mercato	Fatturato_2024	Volume_202
0	9996bcca-9845-46c8-a1d6-3dcf2cecd875	Micro	Industria	Interno	3807947.18	475406.44
1	b647f4a1-dd35-4673-b3f1-e8905788e22e	Micro	Industria	Interno	675027.76	433221.90
2	8d3a1bec-1f76-4772-b944-12117ec0075c	Micro	Servizi	Interno	303786.49	484985.02
3	89fd6804-1232-476f-ab51-074da3765b62	Micro	Servizi	Interno	3111998.21	262853.46
4	3bfa84f5-212e-4b4e-82f4-0c6c0502def9	Micro	Servizi	Interno	6157343.66	70607.44
...	...	...	...	...	...	...
995	c6aaa211-509e-4014-931b-7ac25d910856	Grande	Servizi	Estero	9657941.48	457991.36
996	394d0ea8-3165-4619-a2f3-3dd29678ab68	Grande	Industria	Estero	1443883.95	488741.65
997	1af87865-0268-4085-a6bd-8815764652a5	Grande	Servizi	Estero	3845324.51	45110.09
998	2ccc859b-ee4d-4f40-b751-23438e93c03c	Grande	Servizi	Estero	1748731.28	493102.22
999	74d3309d-1ab3-4cb4-87b9-9f76e70b045f	Grande	Industria	Estero	9294757.87	248874.97

Salvataggio CSV e Parquet

csv_path = “/mnt/data/dataset_sintetico_fatturato.csv” parquet_path = “/mnt/data/dataset_sintetico_fatt
df.to_csv(csv_path, index=False) df.to_parquet(parquet_path, index=False)

Output tabellare

tools.display_dataframe_to_user(name=“Dataset Sintetico Fatturato 2024-2025”, dataframe=df)

(csv_path, parquet_path)