

Codice Python Es15

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Funzione per calcolare la retta di regressione
def regressione_lineare(x, y):
    n = len(x)
    x_mean = np.mean(x)
    y_mean = np.mean(y)

    numeratore = np.sum((x - x_mean) * (y - y_mean))
    denominatore = np.sum((x - x_mean) ** 2)

    a = numeratore / denominatore
    b = y_mean - a * x_mean

    return a, b

# Lettura dati
df = pd.read_excel("Meteo_Chiooggia60.ods", sheet_name='Dati')
Tmin = df["Tmin"].values
Tmed = df["Tmed"].values
Tmax = df["Tmax"].values
Ptot = df["Ptot"].values

# Calcolo della rette di regressione
a1, b1 = regressione_lineare(Tmin, Tmed)
a2, b2 = regressione_lineare(Tmin, Ptot)

# Generazione dei grafici di dispersione con le rette di regressione
fig, axes = plt.subplots(1, 2, figsize=(12, 5))
axes[0].scatter(Tmin, Tmed, label="Dati", color="blue")
axes[0].plot(Tmin, a1 * Tmin + b1, color="red", label=f"Retta: y = {a1:.2f}x +
```

```

{b1:.2f}")
axes[0].set_xlabel("Tmin")
axes[0].set_ylabel("Tmed")
axes[0].legend()
axes[0].set_title("Regressione Tmin vs Tmed")
axes[1].scatter(Tmin, Ptot, label="Dati", color="green")
axes[1].plot(Tmin, a2 * Tmin + b2, color="red", label=f"Retta: y = {a2:.2f}x
+ {b2:.2f}")
axes[1].set_xlabel("Tmin")
axes[1].set_ylabel("Ptot")
axes[1].legend()
axes[1].set_title("Regressione Tmin vs Ptot")
plt.tight_layout()
plt.show()

# Autodecomposizione della matrice di covarianza
dati = np.vstack((Tmin, Tmed, Tmax, Ptot)).T
cov_matrix = np.cov(dati, rowvar=False)
autovalori, autovettori = np.linalg.eigh(cov_matrix)

# Selezione autovalori maggiori
idx = np.argsort(autovalori)[::-1]
autovalori = autovalori[idx]
autovettori = autovettori[:, idx]

# Matrice diagonale degli autovalori
D = np.diag(autovalori)

# Matrice ortonormale degli autovettori
U = autovettori

print("Matrice diagonale degli autovalori D:")
print(D)

print("\nMatrice ortonormale degli autovettori U:")
print(U)

# Trasformazione dei dati nel nuovo sistema di coordinate

```

```
dati_trasformati = dati @ autovettori

# Grafico delle prime due componenti principali
plt.figure(figsize=(6, 5))
plt.scatter(dati_trasformati[:, 0], dati_trasformati[:, 1], color="purple", label
="Dati trasformati")
plt.xlabel("Prima componente principale")
plt.ylabel("Seconda componente principale")
plt.title("Componenti principali dopo la trasformazione PCA")
plt.legend()
plt.grid()
plt.show()
```