

THỰC HÀNH LAB 02:

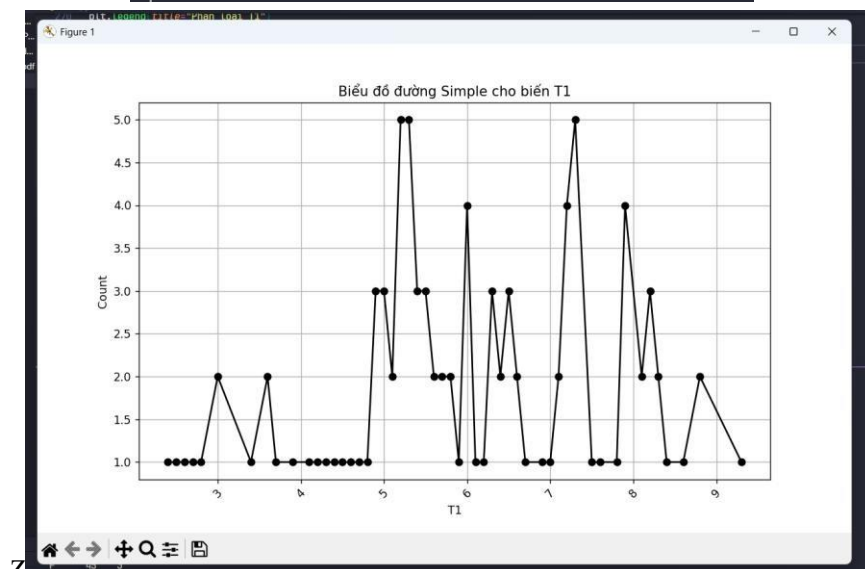
BÀI THỰC HÀNH TRÌNH BÀY DỮ LIỆU

Phần 4: Trực quan hóa dữ liệu nâng cao

```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import scipy.stats as stats
import numpy as np
```

1. Vẽ biểu đồ đường Simple cho biến T1

```
# Phần 4 - Trực quan hóa dữ liệu nâng cao
# Câu 1: Vẽ biểu đồ đường Simple cho biến T1
plt.figure(figsize=(10, 6))
t1_counts = df["T1"].value_counts().sort_index()
plt.plot(t1_counts.index, t1_counts.values,
         marker="o", linestyle="-", color="black")
plt.title("Biểu đồ đường Simple cho biến T1")
plt.xlabel("T1")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.grid()
plt.show()
```



2. Hãy tạo biến phân loại (phanlopt1) cho môn toán (T1) như sau:

- Từ 0 đến dưới 5 = kém (ký hiệu "k")
- Từ 5 đến dưới 7 = trung bình (ký hiệu "tb")

c. Từ 7 đến dưới 8 = khá (ký hiệu “k”)

d. Từ 8 trở lên = giỏi (ký hiệu “g”)

```
# Câu 2: Tạo biến phân loại (phanlopt1) cho môn toán (T1)
def classify_t1(score):
    if score < 5:
        return "k"
    elif 5 <= score < 7:
        return "tb"
    elif 7 <= score < 8:
        return "kha"
    else:
        return "g"

df["phanlopt1"] = df["T1"].apply(classify_t1)
print("\nDữ liệu sau khi phân loại biến T1:")
print(df[["T1", "phanlopt1"]].head())
```

```
Dữ liệu sau khi phân loại biến T1:
  T1 phanlopt1
0  7.2      kha
1  5.4      tb
2  5.6      tb
3  6.6      tb
4  6.0      tb
```

3. Lập bảng tần số cho biến phanloait1

```
# Câu 3: Lập bảng tần số cho biến phanloait1
phanlopt1_counts = df["phanlopt1"].value_counts()
phanlopt1_frequencies = df["phanlopt1"].value_counts(normalize=True) * 100

phanlopt1_table = pd.DataFrame(
    {"Tần số": phanlopt1_counts, "Tần suất (%)": phanlopt1_frequencies})
print("\nBảng tần số và tần suất của phanlopt1:")
print(phanlopt1_table)
```

Bảng tần số và tần suất của phanlopt1:

Tần số Tần suất (%)

phanlopt1

tb 46 46.0

k 23 23.0

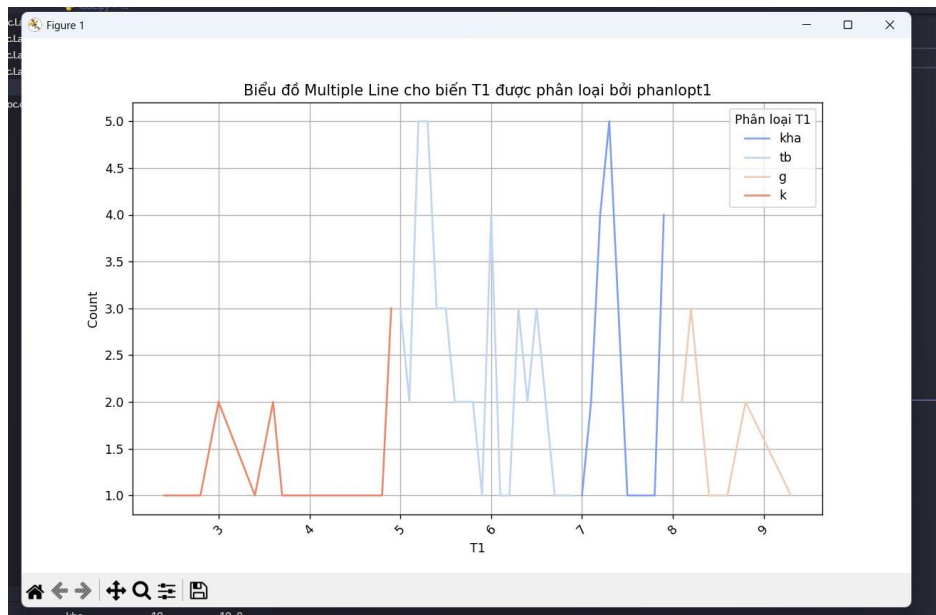
kha 19 19.0

g 12 12.0

PS D:\VLU\N3 (2024-2025)\HK2\Học Sâu\THỰC HÀNH\Lab02> |

4. Vẽ biểu đồ đường Multiple Line cho biến T1 được phân loại bởi biến phanlopt1

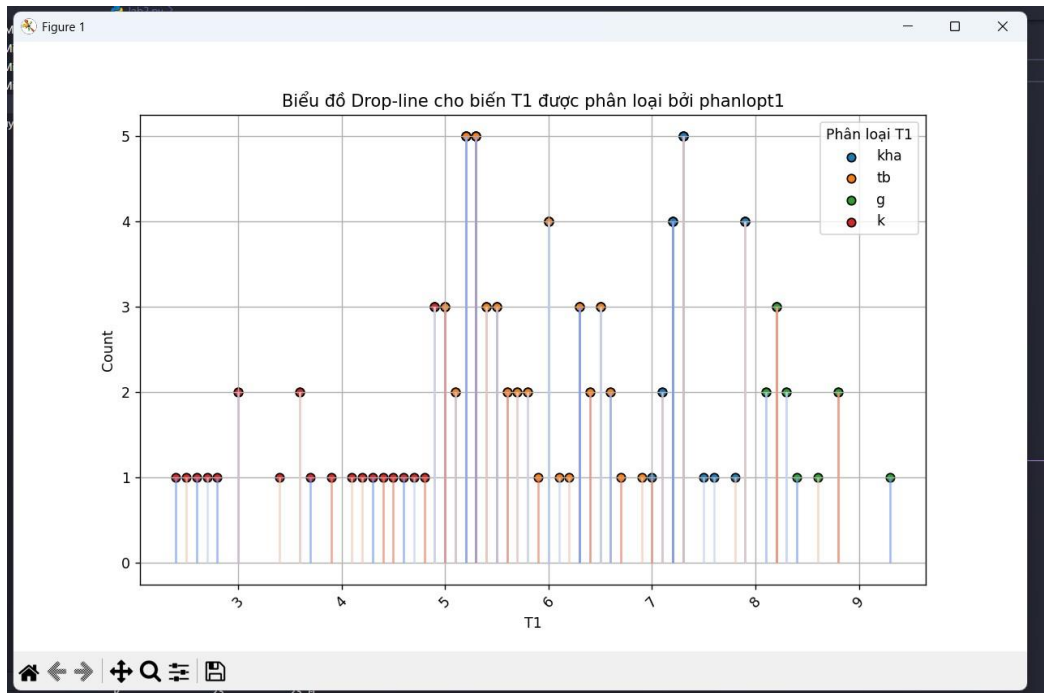
```
# Câu 4: Vẽ biểu đồ đường Multiple Line cho biến T1 được phân loại bởi biến phanlopt1
plt.figure(figsize=(10, 6))
sns.lineplot(data=df, x="T1", y=df["T1"].map(
    df["T1"].value_counts()), hue="phanlopt1", palette="coolwarm")
plt.title("Biểu đồ Multiple Line cho biến T1 được phân loại bởi phanlopt1")
plt.xlabel("T1")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.grid()
plt.legend(title="Phân loại T1")
plt.show()
```



5. Vẽ biểu đồ Drop-line cho biến T1 được phân loại bởi biến phanlopt1

```
# Câu 5: Vẽ biểu đồ Drop-line cho biến T1 được phân loại bởi biến phanlopt1
plt.figure(figsize=(10, 6))
for category in df["phanlopt1"].unique():
    subset = df[df["phanlopt1"] == category]
    plt.vlines(subset["T1"], ymin=0, ymax=subset["T1"].map(df["T1"].value_counts(
    )), colors=sns.color_palette("coolwarm", len(df["phanlopt1"].unique())), alpha=0.7)
    plt.scatter(subset["T1"], subset["T1"].map(
        df["T1"].value_counts()), label=category, edgecolors="black")

plt.title("Biểu đồ Drop-line cho biến T1 được phân loại bởi phanlopt1")
plt.xlabel("T1")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.grid()
plt.legend(title="Phân loại T1")
plt.show()
```



Phần 5: Mô tả dữ liệu và khảo sát dạng phân phối

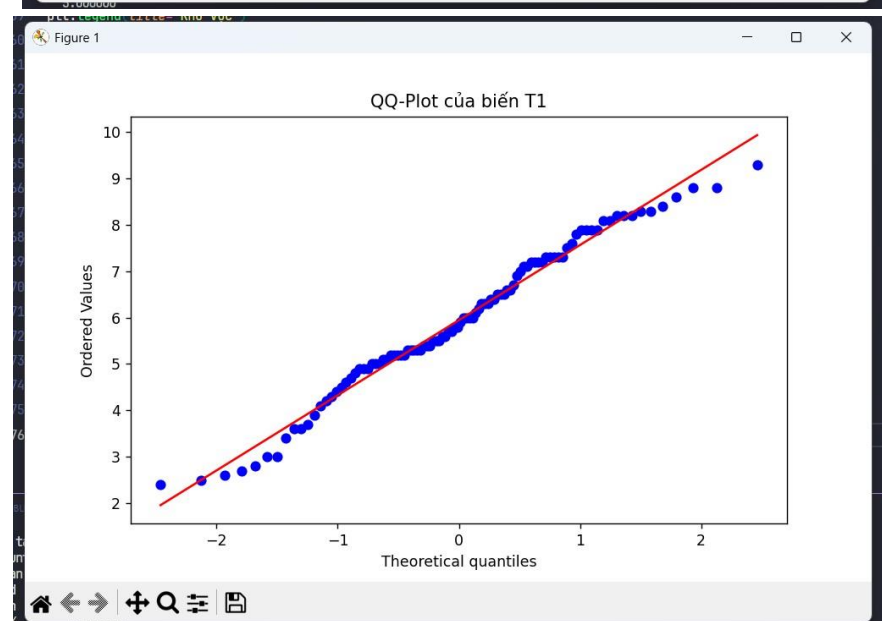
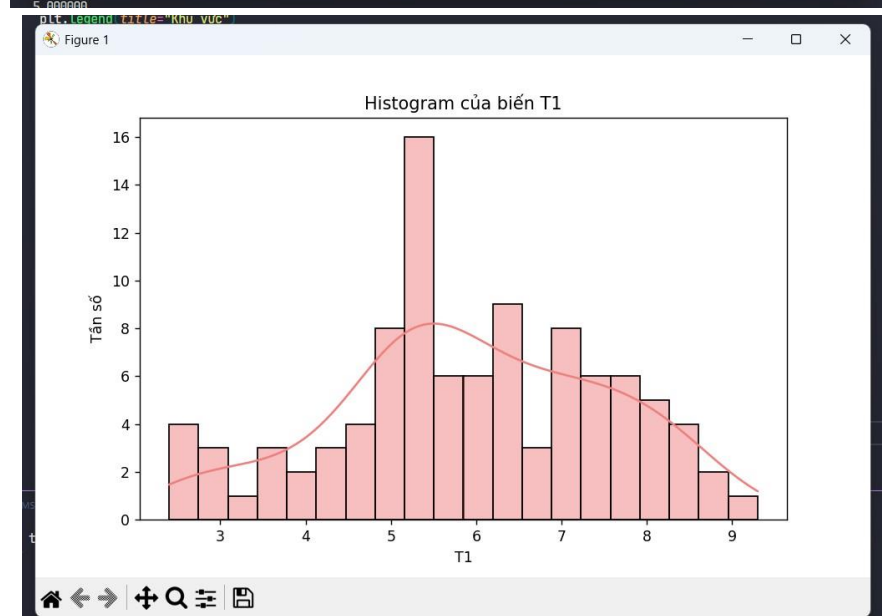
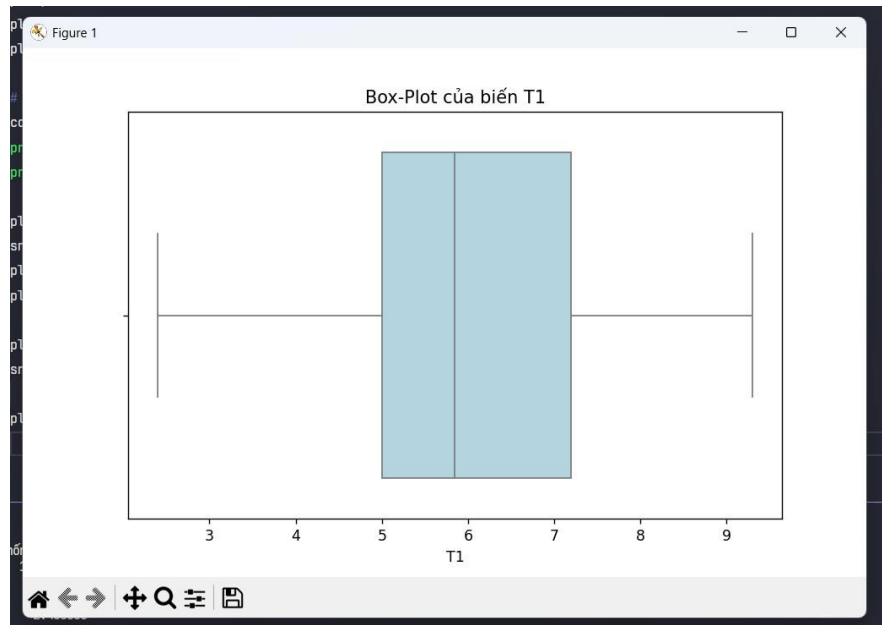
1. Hãy mô tả và khảo sát phân phối cho biến T1

```
# Phần 5 - Mô tả dữ liệu và khảo sát dạng phân phối
# Câu 1: Mô tả và khảo sát phân phối cho biến T1
print("\nMô tả thống kê của biến T1:")
print(df["T1"].describe())

# Vẽ biểu đồ Box-Plot
plt.figure(figsize=(8, 5))
sns.boxplot(x=df["T1"], color="lightblue")
plt.title("Box-Plot của biến T1")
plt.xlabel("T1")
plt.show()

# Vẽ biểu đồ Histogram
plt.figure(figsize=(8, 5))
sns.histplot(df["T1"], bins=20, kde=True, color="lightcoral")
plt.title("Histogram của biến T1")
plt.xlabel("T1")
plt.ylabel("Tần số")
plt.show()

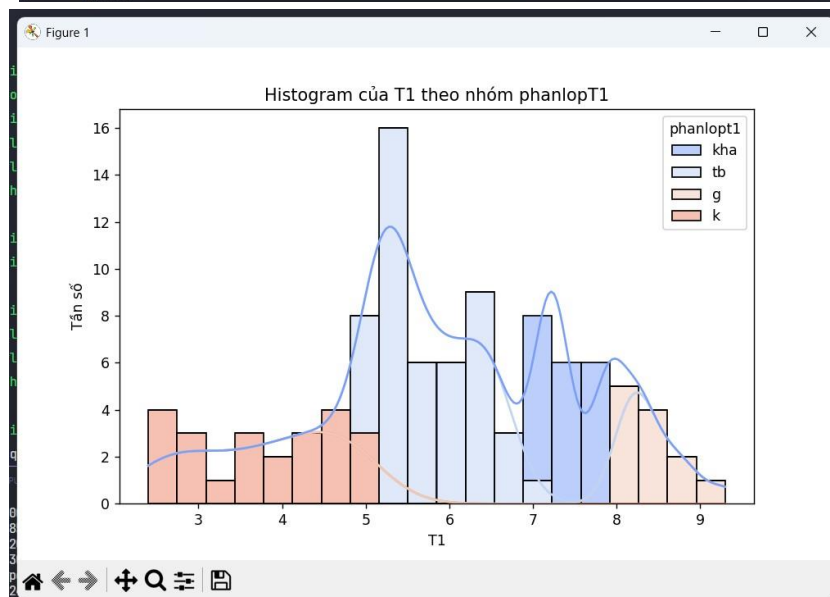
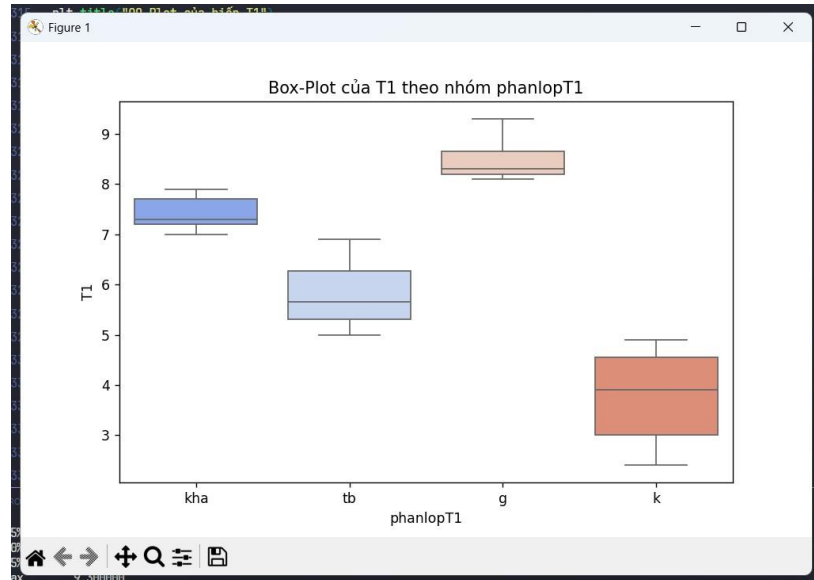
# Vẽ biểu đồ QQ-Plot
plt.figure(figsize=(8, 5))
stats.probplot(df["T1"], dist="norm", plot=plt)
plt.title("QQ-Plot của biến T1")
plt.show()
```



2. Hãy mô tả và khảo sát phân phối cho biến T1 trên từng nhóm phân lớp (phanlopT1)

```
# Câu 2: Mô tả và khảo sát phân phối của T1 trên từng nhóm phanlopT1
plt.figure(figsize=(8, 5))
sns.boxplot(x=df["phanlopT1"], y=df["T1"],
            hue=df["phanlopT1"], palette="coolwarm")
plt.title("Box-Plot của T1 theo nhóm phanlopT1")
plt.xlabel("phanlopT1")
plt.ylabel("T1")
plt.show()

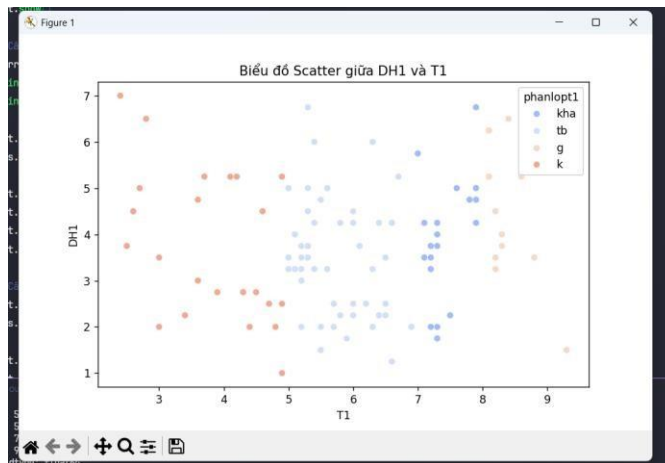
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x="T1", hue="phanlopT1", kde=True,
            multiple="stack", bins=20, palette="coolwarm")
plt.title("Histogram của T1 theo nhóm phanlopT1")
plt.xlabel("T1")
plt.ylabel("Tần số")
plt.show()
```



3. Hãy khảo sát tương quan giữa biến DH1 theo biến T1

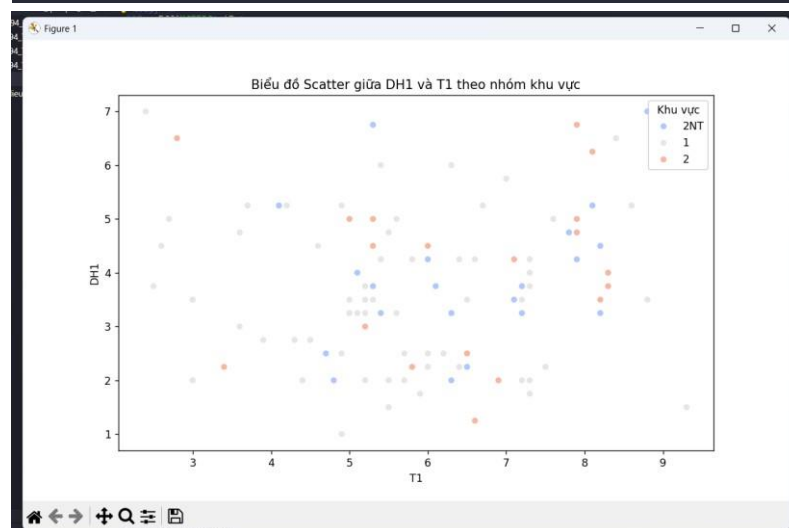
```
# Câu 3: Khảo sát tương quan giữa DH1 theo biến T1
correlation = df[["DH1", "T1"]].corr()
print("\nMa trận tương quan giữa DH1 và T1:")
print(correlation)

plt.figure(figsize=(8, 5))
sns.scatterplot(x=df["T1"], y=df["DH1"],
                hue=df["phanlopt1"], palette="coolwarm", alpha=0.7)
plt.title("Biểu đồ Scatter giữa DH1 và T1")
plt.xlabel("T1")
plt.ylabel("DH1")
plt.show()
```



4. Hãy khảo sát tương quan giữa biến DH1 theo biến T1 trên từng nhóm khu vực

```
# Câu 4: Khảo sát tương quan giữa DH1 theo biến T1 trên từng nhóm khu vực
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x="T1", y="DH1", hue="KV",
                palette="coolwarm", alpha=0.7)
plt.title("Biểu đồ Scatter giữa DH1 và T1 theo nhóm khu vực")
plt.xlabel("T1")
plt.ylabel("DH1")
plt.legend(title="Khu vực")
plt.show()
```



5. Hãy khảo sát tương quan giữa các biến DH1, DH2, DH3

```
# Câu 5: Khảo sát tương quan giữa DH1, DH2, DH3
correlation_matrix = df[["DH1", "DH2", "DH3"]].corr()
print("\nMa trận tương quan giữa DH1, DH2, DH3:")
print(correlation_matrix)

plt.figure(figsize=(8, 5))
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Ma trận tương quan giữa DH1, DH2, DH3")
plt.show()

sns.pairplot(df, vars=["DH1", "DH2", "DH3"],
              hue="phanlopt1", palette="coolwarm")
plt.show()
```

