Python プログラミング実践－データ分析－

手順は　データの入力 → 統計量（平均、標準偏差など）の算出 →
　　グラフの作成（棒グラフ、箱ひげ図）　→ 検定（F検定、T検定）

課題

P組とQ組が数学のテストを受けたところ、
P組のひとの点数は74, 65, 70, 72, 85, 67, 92, 71, 21, 99
Q組のひとの点数は51, 52, 32, 47, 41, 50, 25, 70, 17, 87
であった。
P組、Q組の平均点に差があるか分析しましょう。

1. 必要なモジュールのインポート

複雑な計算処理をするためには、モジュールと呼ばれる予め作られたプログラムをインポートする。今回は、

・numpy 様々な計算に対応するためのもの
・matplotlib　作図をするためのもの
・pandas　計算しやすいようにデータを整理するためのもの

「import numpy as np」 は、numpyをインポートし、今後はnpと表記して使用する」という意味である。

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

ライブラリは日本語表記ができないので、表記できるようにする。ただし、日本語は全角で入力するので、プログラミングをするときには全角・半角の変更に時間がとられる。そのため、アルファベットを使用するのがベストである。

```
!pip install japanize-matplotlib
import japanize_matplotlib
```

2. データの入力（データセットの作成）

処理させるデータセットを作成する。Dataframeでデータを表形式にしている。printでどのように形式になっているか確認する。pはP組のデータ、qはQ組のデータである。

```
data = dict(p = [74, 65, 70, 72, 85, 67, 92, 71, 21, 99],
            q = [51, 52, 32, 47, 41, 50, 25, 70, 17, 87])
df = pd.DataFrame(data = data)

print(df)
```

```
  P組  Q組
0  74  51
1  65  52
2  70  32
3  72  47
4  85  41
5  67  50
6  92  25
7  71  70
8  21  17
9  99  87
```

3. 統計量（平均、標準偏差など）の算出

基本統計量を一気に計算する。

　count サンプル数、mean平均、std標準偏差、min最小値、max最大値、
25%は1/4分位数、50%は1/2分位数、75%は3/4分位数
分位数については、数学の教科書やこちらのサイトで学ぼう
＜ https://bellcurve.jp/statistics/course/19277.html ＞

```
df.describe()
```

```
              P組         Q組
count  10.000000   10.000000
mean   71.600000   47.200000
std    21.030137   20.611755
min    21.000000   17.000000
25%    67.750000   34.250000
50%    71.500000   48.500000
75%    82.250000   51.750000
max    99.000000   87.000000
```

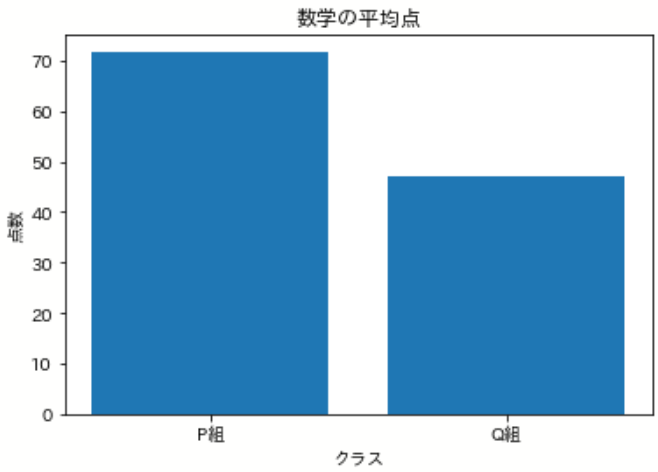
4. 棒グラフの作成

labelはその名のとおりラベル（各群の名称）、
np.arange()で点数を数列のように配列する、
lenはデータの大きさ、
plt.barで棒グラフを作図、alignはグラフの位置、centerで中央寄り

```
label = ["P組",  "Q組"]
x = np.arange(len(label))
y = np.array(df.mean())

plt.bar(x, y, tick_label=label, align="center")
plt.title("数学の平均点")
plt.xlabel("クラス")
plt.ylabel("点数")

plt.show()
```



5. 箱ひげ図の作成

詳しくは数学の宿題として勉強すること。基本統計量として1/4分位数、1/2分位数、3/4分位数を算出した意義がわかるはず。箱ひげ図から外れた位置にある”〇”は外れ値である。

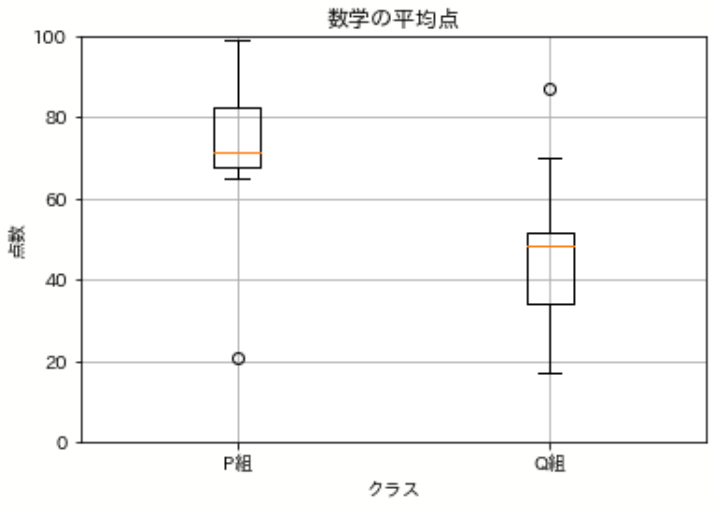
```
# データセットを改めて作成
p = [74, 65, 70, 72, 85, 67, 92, 71, 21, 99]
q = [51, 52, 32, 47, 41, 50, 25, 70, 17, 87]
```

```
fig, ax = plt.subplots()
ax.boxplot((p,q))

ax.set_xticklabels(['P組', 'Q組'])

plt.title('数学の平均点')
plt.xlabel('クラス')
plt.ylabel('点数')
# Y軸の目盛りの範囲
plt.ylim([0,100])
plt.grid()

plt.show()
```



6. F検定（分散の検定）

複雑な統計処理ができるライブラリをインポートする。

```
from scipy import stats
```

データの大きさ(n1, n2としてlenで算出)、分散(var1, var2としてstats.tvarで算出)を求めてF値を算出する。

```
n1 = len(p)
n2 = len(q)
var1 = stats.tvar(p)
var2 = stats.tvar(q)

bigger_var = np.max([var1, var2])
smaller_var = np.min([var1, var2])
f = bigger_var / smaller_var

print(f)
```

```
1.0410084736897163
```

F検定を行い、p値を算出する。0.05以下であれば異分散である。

```
#p値の算出、0.05以下であれば異分散
#自由度を算出
p_df = n1 - 1
q_df = n2 - 1

pval1 = stats.f.cdf(f, p_df, q_df)  # 片側検定のp値 1
pval2 = stats.f.sf(f, p_df, q_df)   # 片側検定のp値 2
pval = min(pval1, pval2) * 2  # 両側検定のp値

print(pval)
```

```
0.9532430348447408
```

7. T検定

T検定には

①同集団（対応のある集団）の検定

②異集団で等分散の検定・・・StudentのT検定（今回の検定はこれに当たる）

③異集団で異分散の検定・・・WelchのT検定

異集団で等分散であることが分かったので、StudentのT検定を行う。pvalueの値が0.05以下で帰無仮説が棄却される。statisticは統計量のこと。

```
stats.ttest_ind(p, q, equal_var=True)
```

```
Ttest_indResult(statistic=2.6203086510868387, pvalue=0.017342066886450555)
```

＜参考＞

異集団で異分散の検定・・・WelchのT検定

```
#equal_varをFalseにするだけ
stats.ttest_ind(p, q, equal_var=False)
```

同集団の検定

```
stats.ttest_rel(p, q)
```

＜補足＞

統計処理に特化したプログラミング言語にRというものがあります。Rを使用すると、もっとシンプルなコードで同等、もしくはそれ以上の処理ができます。google clabでは、言語をPythonからRに変更して使用することもできるので、興味がある人は検索してトライしてみましょう。