```
# いつものモジュールのインポート
import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
```

```
--NORMAL--
```

```
# 配布ファイルをインポートするために，ドライブをマウントする
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

検出力

A商店では，新製品のクッキー(A)の市場での評価を試験するために，試食会において従来品のクッキー(B)と比較して，どちらをより好むかについてのアンケートをとった．その結果が次の通りである．

	A	B	計
どちらを好むか	10	5	15

1. アンケートをとった人数(標本の大きさ)をN，Aのクッキーを好むとした人の割合をp_{sample}とする．
   まず，N(`N`)，p_{sample}(`p_sample`)に，アンケートの結果から分かる適切な数値を入力せよ．
   次に，この標本から無作為に1人を抽出したとき，その1人の回答の仕方は二項分布$B(1,p_{\mathrm{sample}})$に従うとみなせる．この分布の分散vを `v` に入力せよ．

この計算結果(p_{sample}, v)は，母集団(顧客全体)から無作為に1人を抽出したとき，Aを選択する確率pに関する推定量の分布を与える．

```
N = 15
p_sample = 10 / 15
v = p_sample * (1-p_sample)
```

2. この結果から，新製品のクッキーがより好まれると結論付けてよいだろうか．母集団におけるAを選択する確率pについて，
   $$H_0\colon p=0.5,\quad H_1\colon p>0.5$$
   とし，有意水準5%の右片側検定により，判断せよ．ただし，母集団の分散は標本分散vに等しく既知であるとし，二項分布の正規分布への近似を利用し，z検定を行うこと．また，検定の根拠となる `p_sample` 及び `ppt` の値も明示すること．
   ヒント:
   1. 帰無仮説$H_0\colon p=0.5$に基づく，(右片側検定であるから)上位5%点を `ppt` に格納するには，次の書式を利用する．

      ```
      ppt = stats.norm.ppf(0.95,loc=0.5,scale=標準偏差)
      ```

      (`stats.norm.ppf(x,loc=平均,scale=標準偏差)` は，正規分布で下から数えてxとなるx座標を与える．つまり累積分布関数の逆関数である)．
      ここで，今回は標本数が `N` の標本調査を行っているため，標準偏差の計算にあたり，上で定義した `v` を直接利用するのではないことに注意されたい．また，平方根の計算には， `np.sqrt()` 関数を利用せよ．

```
percent = 0.95

ppt = stats.norm.ppf(0.95,loc=0.5,scale=np.sqrt(v/N))

print("p_sample={:.4f}，上位5%点はppt={:.4f}である．".format(p_sample,ppt))
if p_sample < ppt:
  print("p_sample<pptであるため，H_0は棄却されず，クッキーAがより好まれるとは言えない．")
else:
  print("p_sample>pptであるため，H_0は棄却され，クッキーAがより好まれると言える．")
```

```
p_sample=0.6667，上位5%点はppt=0.7002である．
p_sample<pptであるため，H_0は棄却されず，クッキーAがより好まれるとは言えない．
```

3. この検定の検出力を求めよ．効果量は$\Delta=0.1$とする．
   手順:
   1. 効果量 `Delta` に，0.1を入力する．
   2. $H_1\colon p=0.5+\Delta$のもとで，第2種の誤りを犯す確率βを求める．H_1が採択されるべきにも関わらずH_0が棄却できない確率がβであるから，H_1に基づく分布の下で，(右片側検定を行っているので)大きさ `N` の標本から求めたpの標本平均が上で求めた `ppt` を下回る確率を求めればよい．このためには，

      ```
      beta = stats.norm.cdf(ppt,loc=効果量を加味した母平均,scale=標準偏差)
      ```

      とすればよい(`stats.norm.cdf(x)` は `x` までの累積分布関数の値，`x` 以下の確率の総和である)．
   3. 検出力は，効果量Δに基づく$1-\beta$で計算される．`1-beta` を出力せよ:

      ```
      print("Delta={}としたときのこの検定の検出力は，{:.4f}である．".format(Delta,1-beta))
      ```

```
Delta = 0.1
beta = stats.norm.cdf(ppt,loc=0.5+Delta,scale=np.sqrt(v/N))
```

```
print("Delta={}としたときのこの検定の検出力は，{:.4f}である．".format(Delta,1-beta))
```

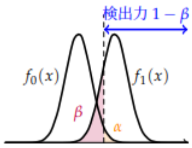
Delta=0.1としたときのこの検定の検出力は，0.2052である

4. この検定の検出力を80%(0.80)としたいとき，次の考えに従って，適切な標本数(アンケート実施人数)nを求める式を入力し，計算結果を出力せよ．
    1. 帰無仮説$H_0: p=0.5$に基づいて$p=0.5$とした母集団の分布における，上側5%点$x_{\rm ppt}$を与える式は
    $$x_{\rm ppt} = 0.5 + z(0.05)\cdot\sqrt{\frac{p(1-p)}{n}} \quad (z(0.05)\text{は標準正規分布における上側5\%点を与える座標})$$
    である．今回は，問題の設定上，先に得られた標本分散を母分散として用いるため，上式の根号の中では$p=p_{\rm sample}$として計算する．
    2. 効果量を$\Delta=0.1$とすると，$H_1: p=0.5+\Delta$に基づく母集団の分布における，(右片側検定であるから)下側20%点を与える式は
    $$x_{\rm beta\_bound} = 0.5 + \Delta - z(0.20)\cdot\sqrt{\frac{p(1-p)}{n}}$$
    で与えられる．この式においても，根号の中では$p=p_{\rm sample}$の値を用いる．
    3. $x_{\rm ppt} = x_{\rm beta_bound}$として，$n$に関する方程式とみなして，$n$について解く．実数解$n$に最も近い整数値(`round(n)`)が，今回求めるべき標本数である．

    ヒント: 標準正規分布における上側100α%点$z(\alpha)$を求めるコードは(平均 `0`，標準偏差 `1` とすればよいから)

    ```
    stats.norm.ppt(1-0.05,loc=0,scale=1)
    ```

    を利用せよ．


```
ppt_0 = stats.norm.ppf(0.95,loc=0,scale=1) # H_0の下での標準正規分布の上側5%点
ppt_1 = stats.norm.ppf(0.80,loc=0,scale=1) #H_1の下での標準正規分布の上側20%点
n = ( (ppt_0 + ppt_1) / Delta )**2 * p_sample * (1 - p_sample)
print("検出力が0.80に一番近くなる標本の大きさは，n={}である．".format(round(n)))
```

適切な標本の大きさは，n=137である．

最後に，答え合わせのために，次のコードを実行してみよ(正しい `v` が，上のセルで入力されている必要がある)．

```
for n in range(1,1000):
  beta = stats.norm.cdf(stats.norm.ppf(0.95,loc=0.5,scale=np.sqrt(v/n)),loc=0.5+Delta,scale=np.sqrt(v/n))
  print("n={}のとき，検出力は{:.4f}である．".format(n,1 - beta))
```

(考察)

対応のある検定

配布ファイル `116_Tempo.csv` に記録されている，2021年12月の神戸，大阪，京都の日毎平均気温のデータをもとに，この3地点の12月の気温について，差があるかどうか判断したい．

0. まずは，変数を標準化してt統計量をつくるための関数を用意しておく．
$$t = \frac{m}{\sqrt{v/\nu}} \quad (m\text{: 平均}, v\text{: 標本分散}, \nu\text{: 標本の大きさ})$$

```
def t_val(m,v,nu): # m: 平均，v: 標本分散，nu: 標本の大きさ
  return m / np.sqrt(v / nu)
```

1. まず，`df1` に配布ファイル `116_Tempo.csv` のデータを読み込もう．データの読み込みには，以下のコードを利用せよ．

    ```
    df1 = pd.read_csv("/content/drive/MyDrive/.../116_Tempo.csv")
    ```

    ここでの "`/content/drive/MyDrive/.../116_Tempo.csv`" は，配布された `116_Tempo.csv` のドライブ上の保存場所(パス)と配布データセットのファイル名を表す．各自で適切なものを設定すること．

正しく読み込めているかを，`df1` の最初の数行を出力することで確認せよ．コードは

```
df1.head()
```

が利用できる．

```
df1 = pd.read_csv("/content/drive/MyDrive/2021_DS教材開発共有用/DS1/116_Tempo.csv")
df1.head()
```

	2021_12	神戸	大阪	京都
0	1	10.1	10.8	10.3

2. `df1` に，神戸と大阪の気温差

$$(\text{大阪}) - (\text{神戸})$$

の列を追加せよ．列のタイトルは，「神戸vs大阪」とせよ．

ヒント:

1. データフレーム `df1` の「神戸」列のデータを取り出すには，`df1["神戸"]` とすればよい．
2. データフレーム `df1` に，「神戸」列と「大阪」列の各データの和の列を「神戸+大阪」という列タイトルで追加するには，

```
df1["神戸+大阪"] = df1["神戸"] + df1["大阪"]
```

とすればよい．

```
df1["神戸vs大阪"] = df1["大阪"] - df1["神戸"]
df1.head()
```

	2021_12	神戸	大阪	京都	神戸vs大阪
0	1	10.1	10.8	10.3	0.7
1	2	7.9	8.0	6.8	0.1
2	3	9.8	9.6	8.0	-0.2
3	4	9.4	10.0	8.0	0.6
4	5	7.8	7.7	6.3	-0.1

3. データフレーム `df1` の「神戸vs大阪」列のデータの平均，不偏標本分散，標本の大きさをそれぞれ `mean_kobe_osaka`，`var_kobe_osaka`，`len_kobe_osaka` に格納し，その結果を表示せよ．

ヒント:

1. データの平均を求めるには，`np.mean(データ)` 関数が使える．
2. データの分散を求めるには，`np.var(データ,ddof=r)` 関数が使える．ここで，`ddof` の値は，$(\text{標本数}) - (\text{自由度})$，すなわち束縛条件の数である．今回は，`ddof=1` とする．
3. 標本の大きさを求めるには，`len(データ)` が使える．

```
mean_kobe_osaka, var_kobe_osaka, len_kobe_osaka = np.mean(df1["神戸vs大阪"]), np.var(df1["神戸vs大阪"],ddof=1),len(df1)
print("神戸と大阪の気温差の平均は{:.4f}，不偏標本分散は{:.4f}，標本数は{}である．".format(mean_kobe_osaka,var_kobe_osaka,len_kobe_osaka))
```

```
神戸と大阪の気温差の平均は-0.1097，不偏標本分散は0.3142，標本数は31である．
```

4. 神戸と大阪に気温差が認められるか否かについて，気温差が正規分布に従うと仮定し，

$$H_0 : \mu_d = 0, \quad H_1 : \mu_d \neq 0 \qquad (\mu_d\text{は気温差の平均})$$

として，有意水準5%の両側t検定を行え．検定の根拠となるt統計量，およびt分布の上側パーセント点も表示すること．

ヒント:

1. 神戸と大阪の気温差のt統計量は，上で定義した `t_val()` を使え．
2. （両側検定であるから）自由度 `n-1` のt分布の上側$100\alpha/2\%$点の座標を与える関数は，

```
t_ppt = stats.t.ppf(1 - a/2, n - 1)
```

を利用せよ．

```
t_kobe_osaka = t_val(mean_kobe_osaka, var_kobe_osaka, len_kobe_osaka)
t_ppt = stats.t.ppf(0.975,len_kobe_osaka-1)
print("神戸と大阪の12月の気温差のt統計量は{:.4f}であり，自由度{}のt分布における上側2.5%点は{:.4f}である．".format(t_kobe_osaka, len_kobe_osaka-1, t_ppt))

if abs(t_kobe_osaka) < t_ppt:
  print("H_0は棄却されず，神戸と大阪の12月の日毎平均気温に差はあるとはいえない．")
else:
  print("H_0は棄却され，神戸と大阪の12月の日毎平均気温に差はあるといえる．")

```

```
神戸と大阪の12月の気温差のt統計量は-1.0894であり，自由度30のt分布における上側2.5%点は2.0423である．
H_0は棄却されず，神戸と大阪の12月の日毎平均気温に差はあるとはいえない．
```

5. 同様の手続きにより，神戸と京都に気温差があるか否かについて，

$$H_0 : \mu_d = 0, \quad H_1 : \mu_d \neq 0$$

とした有意水準5の両側t検定を行え．

```
# 神戸と京都の気温差をdf1の「神戸vs京都」列に格納する
df1["神戸vs京都"] = df1["京都"] - df1["神戸"]
df1.head()
```

	2021_12	神戸	大阪	京都	神戸vs大阪	神戸vs京都
0	1	10.1	10.8	10.3	0.7	0.2
1	2	7.9	8.0	6.8	0.1	-1.1
2	3	9.8	9.6	8.0	-0.2	-1.8

```
# 神戸と京都の気温差の平均, 分散, 標本の大きさをそれぞれの変数に格納する

mean_kobe_kyoto, var_kobe_kyoto, len_kobe_kyoto = np.mean(df1["神戸vs京都"]), np.var(df1["神戸vs京都"],ddof=1), len(df1)

# t統計量を求める(上側2.5%点はすでに大阪との対比のところで作ったt_pptを利用すればよい)

t_kobe_kyoto = t_val(mean_kobe_kyoto, var_kobe_kyoto, len_kobe_kyoto)

# 統計量を比較し, 検定結果を表示する

print("神戸と京都の12月の気温差のt統計量は{:.4f}であり, 自由度{}のt分布における上側2.5%点は{:.4f}である.".format(t_kobe_kyoto, len_kobe_kyoto-1, t_ppt))

if abs(t_kobe_kyoto) < t_ppt:
  print("H_0は棄却されず, 神戸と京都の12月の日毎平均気温に差はあるとはいえない.")
else:
  print("H_0は棄却され, 神戸と京都の12月の日毎平均気温に差はあるといえる.")
```

```
神戸と京都の12月の気温差のt統計量は-13.3505であり, 自由度30のt分布における上側2.5%点は2.0423である.
H_0は棄却され, 神戸と京都の12月の日毎平均気温に差はあるといえる.
```

(考察)

Welch's t-Test

配布ファイル 116_AvsB.csv には, 40人にA, Bの2種類の問題集を無作為に割り当てて使用してもらい, 使用の前後に受験したテストの成績を記録している(データは架空です).

1. まず, df2 に配布ファイル 116_AvsB.csv のデータを読み込もう. データの読み込みには, 以下のコードを利用せよ.

   ```
   df2 = pd.read_csv("/content/drive/MyDrive/.../116_AvsB.csv")
   ```

   ここでの "/content/drive/MyDrive/.../116_AvsB.csv" は, 配布された 116_AvsB.csv のドライブ上の保存場所(パス)と配布データセットのファイル名を表す. 各自で適切なものを設定すること.

正しく読み込めているかを, df2 の最初の数行を出力することで確認せよ.

```
df2 = pd.read_csv("/content/drive/MyDrive/2021_DS教材開発共有用/DS1/116_AvsB.csv")
df2.head()
```

	ID	事前テスト	使用問題集	事後テスト
0	1	57	A	78
1	2	54	B	65
2	3	52	A	74
3	4	51	A	69
4	5	54	A	77

2. データフレーム df2 に,「差分」という列タイトルで,「事前テスト」と「事後テスト」のそれぞれの差を計算した列を作れ.

```
df2["差分"] = df2["事後テスト"] - df2["事前テスト"]
df2.head()
```

	ID	事前テスト	使用問題集	事後テスト	差分
0	1	57	A	78	21
1	2	54	B	65	11
2	3	52	A	74	22
3	4	51	A	69	18
4	5	54	A	77	23

3. データフレーム df2 のデータのうち, 使用問題集がAであるものBであるもののみを抜き出し, それぞれデータフレーム df2_A, df2_B に格納せよ.

   ヒント:

   1. データフレームの中から特定の条件を満たす行のみを取り出すには, df2[条件] を利用する. 今回の場合, 例えば「使用問題集」列にAが入力されている行のみを抜き出すので, 条件は

      ```
      df2["使用問題集"] == "A"
      ```

      を用いればよい.
   2. 正しく抜き出せているかを見るには, df2_A.head() などを実行し, df2_A の中身を見るとよい.

```
df2_A = df2[df2["使用問題集"] == "A"]
df2_B = df2[df2["使用問題集"] == "B"]
```

4. データフレーム df2_A , df2_B における「差分」列のデータの平均，不偏標本分散，標本の大きさを取り出し，それぞれ mean_X , var_X , len_X（X は A , B のいずれか）とし，それらのデータを出力せよ．分散計算において，自由度を加味することを忘れないように注意せよ．

```
mean_A, var_A, len_A = np.mean(df2_A["差分"]), np.var(df2_A["差分"],ddof=1), len(df2_A)
mean_B, var_B, len_B = np.mean(df2_B["差分"]), np.var(df2_B["差分"],ddof=1), len(df2_B)
print("問題集Aを利用した群の平均は{:.4f}，不偏標本分散は{:.4f}，標本の大きさは{}である．".format(mean_A,var_A,len_A))
print("問題集Bを利用した群の平均は{:.4f}，不偏標本分散は{:.4f}，標本の大きさは{}である．".format(mean_B,var_B,len_B))
```

```
問題集Aを利用した群の平均は16.6000，不偏標本分散は46.5684，標本の大きさは20である．
問題集Bを利用した群の平均は19.9000，不偏標本分散は34.5158，標本の大きさは20である．
```

5. 問題集Aを利用した群と問題集Bを利用した群の平均の差の分散を求め，var_com に格納せよ．また，この計算結果をもとに，平均の差のt統計量 t_AB を計算し，その値を出力せよ．

```
var_com = var_A / len_A + var_B / len_B
t_AB = (mean_A - mean_B) / np.sqrt(var_com)
t_AB
```

```
-1.638931458635994
```

6. t検定を行うために，ウェルチの近似式から計算される値 nu_welch を計算し，nu_welch に最も近い整数 nu_star を求めよ．ここで求めた nu_star が，次に行うt検定の自由度を与える．

```
nu_welch = var_com**2 / ((var_A/len_A)**2/(len_A-1) + (var_B/len_B)**2/(len_B-1))
nu_star = round(nu_welch)
nu_star
```

```
37
```

7. 上で求めた自由度 nu_star を用いて，問題集Aを用いたときの成績の上昇具合μ_A，問題集Bを用いた時の成績の上昇具合μ_Bについて，

$$H_0 \colon \mu_A - \mu_B = 0, \quad H_1 \colon \mu_A - \mu_B \neq 0$$

として有意水準5%の両側t検定（Welch検定）を行え．

```
t_AB_ppt = stats.t.ppf(0.975,nu_star)

print("t統計量は{:.4f}であり，自由度{}のt分布の上側2.5%点は{:.4f}である．".format(t_AB,nu_star,t_AB_ppt))

if abs(t_AB) < t_AB_ppt:
    print("H_0は棄却されず，使用問題集AとBによる成績の上昇具合に差はあるとはいえない．")
else:
    print("H_0は棄却され，使用問題集AとBによる成績の上昇具合に差はあるといえる．")
```

```
t統計量は-1.6389であり，自由度37のt分布の上側2.5%点は2.0262である．
H_0は棄却されず，使用問題集AとBによる成績の上昇具合に差はあるとはいえない．
```

8. 効果量を 3（つまり3点以上差があれば2つの問題集に差があるとみなすに十分でである）として，この検定の検定力を求めよ．今回は両側検定を行っているので，H_0 の採択域に注意せよ．

```
d = 3
t_border = stats.t.ppf(0.975,nu_star)
b = stats.t.cdf(t_border,nu_star,loc=d) - stats.t.cdf(-t_border,nu_star,loc=d)
print("この検定の検出力は{:.4f}である．".format(1-b))
```

```
この検定の検出力は0.8318である．
```

（考察）

＜本授業の学び＞

本授業で学んだことを，下のテキストボックスに記入して下さい．

（ここに本授業の学びを記入する）