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

モジュールpandasについて

データを集めて分析をするには, 収集したデータの整理が必要になる. モジュール pandas は, データを表にして管理したり, 表からさまざまな項目のデータを抽出したりするのに利用される.

次のコードを実行してから, 先の内容を読み進めること.

```
ser1 = pd.Series([66,76,82,63,59,52,87,61,73,68,55,67,64])

data1 = {"Number":["4401","4402","4403","4404","4405","4406","4407"],
         "Name":["Akao","Ozawa","Kurokawa","Tominaga","Negoro","Yano","Yokota"]
         "Math":[66,76,82,63,59,52,87],
         "English":[61,73,68,55,67,64,74],
         "Birth_Month":["February","June","August","November","April","October"

df1 = pd.DataFrame(data1)
```

Seriesオブジェクト

Series オブジェクトは, 1列の表を扱う, 配列のようなものである. 次のコードによって, 上で定義した ser1 を出力してみよ.

また, メソッド .values , .index を ser1 につけると, 出力がどのように変化するか, コメントアウトを外して確認せよ.

```
# 上のセルから, ser1というSeriesオブジェクトについて出力する
#ser1
ser1.values
# ser1.index
```

```
array([66, 76, 82, 63, 59, 52, 87, 61, 73, 68, 55, 67, 64])
```

DataFrameオブジェクト

2次元配列を扱うには, DataFrame オブジェクトを利用する.

```
# 上のセルから, df1というDataFrameオブジェクトを出力する
df1
# df1.T # 行と列を入れ替える(Tは転置, Transposeの意)
```

```
# df1.Birth_Month # 特定の1列のみを取り出す場合の書き方
# df1[["Math","Birth_Month"]] # 複数の列を取り出す場合の書き方
# df1["English"][5] # English列の第5行を取り出す場合
```

	Number	Name	Math	English	Birth_Month
0	4401	Akao	66	61	February
1	4402	Ozawa	76	73	June
2	4403	Kurokawa	82	68	August
3	4404	Tominaga	63	55	November
4	4405	Negoro	59	67	April
5	4406	Yano	52	64	October
6	4407	Yokota	87	74	June

条件に合う行の抽出

データフレームから, 条件に合う行を抽出するには, 以下のセルのように条件式を書けばよい.

```
# df1[df1["Birth_Month"] == "June"] # Birth_Month列がJuneであるデータのみを抽出
df1[df1["Math"] >= 60] # Mathの成績が60以上であるデータのみを抽出
```

	Number	Name	Math	English	Birth_Month
0	4401	Akao	66	61	February
1	4402	Ozawa	76	73	June
2	4403	Kurokawa	82	68	August
3	4404	Tominaga	63	55	November
6	4407	Yokota	87	74	June

行のソート(並び替え)

データフレームの特定の列を数値の順に並び替えるには, `sort.values()` メソッドを利用して, 以下のセルのように書けばよい. 詳しくは, こちらなどを参照.

```
df1.sort_values("English") # 英語の成績で昇順にソート
# df1.sort_values("English",ascending=False) #降順にソートしたい場合は, ascending=Fal
```

	Number	Name	Math	English	Birth_Month
3	4404	Tominaga	63	55	November
0	4401	Akao	66	61	February
5	4406	Yano	52	64	October
4	4405	Negoro	59	67	April

外部ファイルからデータを取得する方法

以下では, Notebookファイルと同一のフォルダに配置されたExcel, CSVのデータを読み込む方法, および, ColaboratoryのNotebookファイルと同一のフォルダに配置されたSpreadsheetデータの読み込み方について解説する.

以下では, 各自利用している環境に合わせて, 実行するコードを選ぶこと.

Jupyter Notebookのローカル環境下での作業用

CSV形式のファイルからデータを読み込む場合は, このまま下のセルを実行する.

Excel形式のファイルからデータを読み込む場合は, 下のセルの$8 \sim 9$行目をコメントアウト(行頭に#を加える)し, $2 \sim 5$のコメントアウトを外す.

```
# Excelファイルから読み込む場合
# excel_file_name = "DogOrCat_Test.xlsx" # 読み込むファイル名を文字列として入力(拡張子を
# xl_data = pd.ExcelFile(excel_file_name) # データをxl_dataに格納
# sheets = xl_data.sheet_names # Excelのシートのリストを作成
# df = xl_data.parse(sheets[0]) # 0番目のシートのデータをDataFrame形式にしてdfに格納

# CSV形式のファイルから読み込む場合
csv_file_name = "DogOrCatOr_4-3.csv" # 読み込むファイル名を文字列として, csv_file_nameに
df = pd.read_csv(csv_file_name) # データをDataFrame形式にしてdfに格納
```

Google Colab内での作業用

Google Colaboratoryを使用する場合は, Google Drive内に置いたスプレッドシートからデータを読み込む必要がある.

基本的には, スプレッドシートのURLを下のコードの所定の場所に貼り付け, 順にコードを実行すればよい. 詳細は, このQiitaの記事などを参考にされたい.

```
# 必要なモジュールのインストール
!pip install --upgrade -q gspread
```

```
# 必要なモジュールのインポート
from google.colab import auth
```

```
from oauth2client.client import GoogleCredentials
import gspread

# GoogleDriveをColaboratoryにロードする(認証作業あり)
auth.authenticate_user()
gc = gspread.authorize(GoogleCredentials.get_application_default())

# 文字列urlに読み込むスプレッドシートのURLを入力
url = "https://docs.google.com/spreadsheets/d/1ym4x88o522lPyvcLuv-SmB5D4o3xa1afI

sheet = gc.open_by_url(url).get_worksheet(0) # urlに指定したURLから, スプレッドシートのC
df = pd.DataFrame(sheet.get_all_values()) # sheetのデータをDataFrame形式にしてdfに格
df
```

	0	1
0	RandNum	DogOrCatOr
1	0.002127418695	「犬派」
2	0.0116918293	中立
3	0.02330211606	中立
4	0.03087775439	「猫派」
...	...	...
110	0.9854067944	中立
111	0.9866553689	「猫派」
112	0.9869737949	「犬派」
113	0.9903752154	「犬派」
114	0.9928164818	「犬派」

115 rows × 2 columns

模擬アンケート結果の分析

先週採ったアンケートの結果の分析を行おう.

問題設定

「模擬アンケート①」に回答した10回生を母集団とし, 回答者から無作為に抽出した20人の回答結果から, 母集団における「犬派」と回答した人数を, 信頼度95%の信頼区間を求めることにより, 推測する.

1. 抽出するサンプル数 n を20に設定する.
2. 信頼度を設定する. ここではαを, 信頼区間が母集団における「犬派」と回答した人数を推定し損なう確率として, 信頼度($100(1-\alpha)$%)を間接的に設定する.

分析

1. 回答結果を収めたデータフレーム(df , 上のセルで定義したもの)の行数(総回答数)を L とする.
2. リスト$[1, L]$(1以上L以下の整数値)から20個(n に格納した値)の一様乱数(どの整数も等確率で選ばれる, 一様分布に従う乱数)を非復元抽出(randlist)し, 抽出した乱数に対応する回答のみを含むデータフレームを構成する(df.iloc[randlist]).
3. df.iloc に含まれる「犬派」の回答件数を計上し, n_dog に格納する. このとき, 抽出した標本における「犬派」の割合 p_dog は

   ```
   p_dog = n_dog / n
   ```

   と計算される.
4. 母集団における「犬派」の割合(**母比率**)を, 標本における「犬派」の割合 p_dog から, 信頼度を95%として, 以下の考え方に従って区間推定せよ.

   1. 母集団(大きさL = 全回答数)のうち「犬派」がk_{dog}件であるとする. 母比率は$\frac{k_{\mathrm{dog}}}{L}$である.
   2. そこから無作為に大きさ$n = 20$の標本を(非復元)抽出する. 標本に含まれる「犬派」の割合$p_{\mathrm{dog}} = \frac{n_{\mathrm{dog}}}{n}$は, 母比率の推定量となる.
   3. 母集団から大きさ$n = 20$の標本を無作為抽出するとき, 標本の中に含まれる「犬派」の割合は, 二項分布$B\left(1, p_{\mathrm{dog}}\right)$に従うとみなせる. この二項分布を近似する正規分布$N\left(1 \cdot p_{\mathrm{dog}}, 1 \cdot p_{\mathrm{dog}} \cdot \left(1 - p_{\mathrm{dog}}\right)\right)$から大きさ$n = 20$の標本を抽出すると考えて, 上下$2.5\%$点の値($z_{0.025} = 1.96$)を求めることで, 信頼度$95\%$の信頼区間
   $$\left[p_{\mathrm{dog}} - z_{0.025} \cdot \sqrt{\frac{p_{\mathrm{dog}} \cdot \left(1 - p_{\mathrm{dog}}\right)}{20}}, \quad p_{\mathrm{dog}} + z_{0.025} \cdot \sqrt{\frac{p_{\mathrm{dog}} \cdot \left(1 - p_{\mathrm{dog}}\right)}{20}}\right]$$
   の計算式に代入する.

分析結果の集約

1. クラスルームに配布されている, フォーム「114_Estimation_Interval_test_4-3」に, 上下側信頼限界を入力する.

コードのあとに, 母比率の推定に関する解説を載せている. 必要に応じて参照してほしい.

```
 1 # 抽出するサンプル数を指定する
 2 n = 20
 3
 4 # 信頼度の設定(1-aが信頼度, つまり, aは閾値を超える辺境の確率を定めている)
 5 a = 0.05
 6
 7 # 標準正規分布の累積分布関数の値がa/2となるxの値をパーセント点から求めている(累積分布関数の逆関数
 8 z_ppv = stats.norm.ppf(1 - a / 2, loc=0, scale=1)
 9 # t_ppv = sp.stats.t.ppf(1 - a / 2, n - 1)
10
```

```
print("標準正規分布に基づく",100*(1-a/2),"%点の値は:",z_ppv)
# print("自由度",n-1,"のティー分布に基づく",100*(1-a/2),"%点の値は:",t_ppv)

L = len(df) # DataFrameの行数の取得
ser = np.arange(1,L+1) # リスト[1,L]
randlist = np.random.choice(ser,n,replace=False) # N個の乱数のリストを生成

# 回答結果のDataFrameから上の乱数リストのindexを抜き出し, 標本の中で「犬派」と回答している件数をカ
print("抽出された標本の回答結果は以下の通り:\n",df.iloc[randlist,1])
n_dog = len(df.iloc[randlist,1][df.iloc[randlist,1] == "「犬派」"])
p_dog = n_dog / n
print("「犬派」の回答件数は, ",n_dog,"件であり, その標本に占める割合は, ",p_dog)


# 標本平均の信頼度(1-a)の上側信頼限界と下側信頼限界を求めている
I_dogL = p_dog - z_ppv * np.sqrt(p_dog * (1 - p_dog) / n)
I_dogU = p_dog + z_ppv * np.sqrt(p_dog * (1 - p_dog) / n)
print(100*(1-a),"%信頼区間は",[I_dogL,I_dogU])

# I_dog_tL = p_dog - t_ppv * np.sqrt(p_dog * (1 - p_dog) / n)
# I_dog_tU = p_dog + t_ppv * np.sqrt(p_dog * (1 - p_dog) / n)
# print("ティー分布に基づいて計算した",100*(1-a),"%信頼区間は",[I_dog_tL,I_dog_tU])
```

```
--NORMAL--
```

```
標準正規分布に基づく 97.5 %点の値は: 1.959963984540054
抽出された標本の回答結果は以下の通り:
 66    「猫派」
70    「犬派」
90    「犬派」
10    「犬派」
60    「犬派」
43    「猫派」
26    「犬派」
84    「犬派」
11    「犬派」
99    「犬派」
41    「犬派」
28    「犬派」
68    「猫派」
36    「犬派」
85    「犬派」
38    「犬派」
51    「犬派」
95      中立
25    「犬派」
92      中立
Name: 1, dtype: object
「犬派」の回答件数は, 15 件であり, その標本に占める割合は, 0.75
95.0 %信頼区間は [0.5602273032177509, 0.9397726967822491]
```

一般論(母比率の推定)

1. 母集団Uにおける, 性質Aをもつ個体の含まれる割合を調べたい. 大きさnの標本を無作為抽出すると, 性質Aをもつ個体がk個あった. このことから, 母集団における性質Aをもつ個

体の比率(母比率)の信頼度$100(1-\alpha)$%の信頼区間は

$$\left[\frac{k}{n}-z_{\frac{\alpha}{2}}\cdot\sqrt{\frac{\frac{k}{n}\cdot\left(1-\frac{k}{n}\right)}{n}},\quad \frac{k}{n}+z_{\frac{\alpha}{2}}\cdot\sqrt{\frac{\frac{k}{n}\cdot\left(1-\frac{k}{n}\right)}{n}}\right]$$

で与えられる. 教科書では, $p_0=\frac{k}{n}$として表記している.

2. 厳密には, 母分散が未知であるから, ここで推定するためのパーセント点として正規分布のものではなく, 母分散を標本不偏分散に置き換えた, 自由度$n-1$のティー分布のパーセント点を利用することになる.

$$\left[\frac{k}{n}-t_{\frac{\alpha}{2}}(n-1)\cdot\sqrt{\frac{\frac{k}{n}\cdot\left(1-\frac{k}{n}\right)}{n}},\quad \frac{k}{n}+t_{\frac{\alpha}{2}}(n-1)\cdot\sqrt{\frac{\frac{k}{n}\cdot\left(1-\frac{k}{n}\right)}{n}}\right]$$

ティー分布については教科書で触れられていないが, KP等で統計的処理を行い, 推定を行う際には知っておくべき存在である.

【解説】母比率の推定

十分な大きさの母集団Uにおいて, 性質Aをもつ個体の含まれる割合がp_0であるとする. このとき, 母集団Uから無作為に1個の個体を抽出して, 性質Aをもつ確率はp_0である. 確率変数Xを, 抽出した個体が性質Aをもつとき$X=1$, もたないとき$X=0$で定義すると, Xは二項分布$B(1,p_0)$に従う: $X\sim B(1,p_0)$.

母集団Uから個体をn個抽出し, 大きさnの標本$[s_1,s_2,\ldots,s_n]$を作る. s_iの性質の有無を確率変数X_iで表すことにする. $X_i\in B(1,p_0)$であり, (二項分布は再生性をもつので)

$$\sum_{i=1}^{n}X_i=X_1+X_2+\cdots+X_n\sim B(n,p_0).$$

$B(n,p_0)$を, (平均と分散を等しくする)正規分布$N\left(np_0,np_0(1-p_0)\right)$に置き換える:

$$\sum_{i=1}^{n}X_i\sim N\left(np_0,np_0(1-p_0)\right)$$

この結果から, $\overline{X}=\frac{1}{n}\sum_{i=1}^{n}X_i$の平均と分散は

$$\begin{aligned}E\left(\overline{X}\right)&=E\left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)=\frac{1}{n}E\left(\sum_{i=1}^{n}X_i\right)=\frac{1}{n}\cdot np_0\\&=p_0,\\V\left(\overline{X}\right)&=V\left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)=\frac{1}{n^2}V\left(\sum_{i=1}^{n}X_i\right)=\frac{1}{n^2}\cdot np_0(1-p_0)\\&=\frac{p_0(1-p_0)}{n}.\end{aligned}$$

ゆえに, $\overline{X}\sim N\left(p_0,\frac{p_0(1-p_0)}{n}\right)$である.

あとの流れは, 区間推定の求め方に従う.

＜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）