1日で学べる！ はじめてのPython

データサイエンス・オンライン講座《特別編》

2023年12月9日(土)、 23日(土)

主催：総務省統計局
運営委託：株式会社ネットラーニング

1

自己紹介

佐久間　貴士

- 千葉県立保健医療大学
  健康科学部歯科衛生学科講師
- 立正大学データサイエンス学部 非常勤講師
- 日本大学法学部 非常勤講師

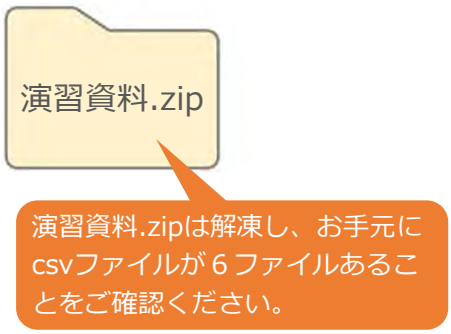
2

本日の講義資料

次の２点をお手元にご用意ください

①20231223_1日で学べる！はじめてのPython.pdf（投影しているPDF資料）

②演習資料.zip



3

講義資料の格納先について（ダウンロード方法）

① カリキュラム一覧で ＋ ボタンをクリックしてください。

② 事前接続テストや講座当日のZoom参加のためのページならびに講義資料等のページが表示されますので、 ボタンの横にあるタイトルをクリックしてください。

③ 下記2点の資料をクリックいただきダウンロードしてください。
- 20231223_1日で学べる！はじめてのPython_講義資料.pdf
- 演習資料.zip

4

講義のまえに

推奨環境：パソコン（Windows/Mac 問いません）

※ タブレットやスマートフォンでの操作については
ご質問に回答できない場合があります。

5

講義のまえに

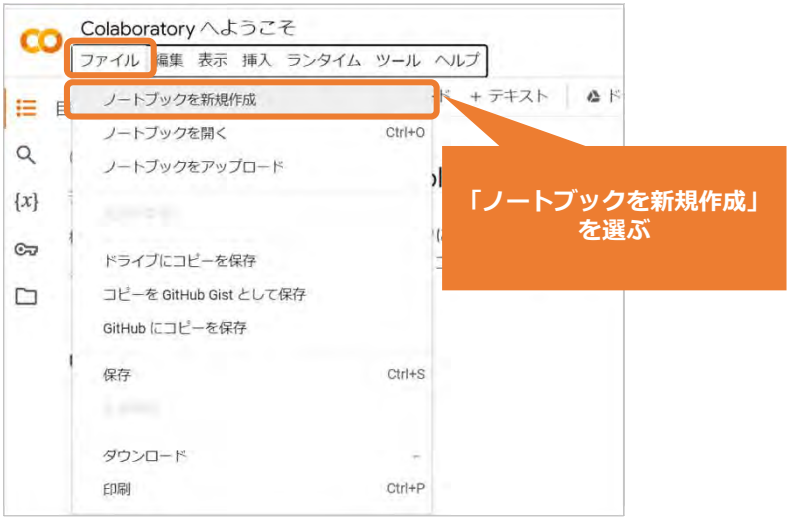
Google Colaboratory の起動

https://colab.research.google.com/ にアクセス

6

講義のまえに

Google Colaboratory の起動



7

講義のまえに

Google Colaboratory の起動

8

講義のまえに

プログラムの保存について

Colaboratoryで作成したノートブックは Google Drive※（https://drive.google.com/）内の「Colab Notebooks」に自動で保存されます。

※ Googleが提供しているクラウドストレージで、Googleアカウントでログインします

名前の変更について

Google Drive もしくは Google Colaboratory 、どちらからでも変更することができます。

9

講義のまえに

さまざまなプログラミング言語

- Python
- R
- C言語
- C++
- Java
- PHP

10

講義のまえに

ExcelとPythonのちがい

- Excelの限界
- 少ないコードで実装でき，標準ライブラリやコミュニティから提供されたモジュールが豊富，導入が容易
  - データ量が膨大でも重たくない
  - 再現性の高さ
  - オープンライセンス
  - ライブラリによる効率的なデータ分析
  - 機械学習と深層学習

11

講義のまえに

- **実際に手を動かして、コードを入力してみよう！**
  本日作成するコードは、講義終了後にみなさまにお配りします。
  講義時間中は、コピー&ペーストではなく、ご自身でコードを打ち込んでみましょう。

12

今日の内容

1. データ分析に必要な統計学の基礎を学ぶ
2. 比較して2変数の関係を考える
3. データに基づいて判断を下すための手法を学ぶ
4. ビジネスにおける予想と分析結果の報告

13

今日の内容

1. データ分析に必要な統計学の基礎を学ぶ
2. 比較して2変数の関係を考える
3. データに基づいて判断を下すための手法を学ぶ
4. ビジネスにおける予想と分析結果の報告

14

第1章の内容

1. データの種類とは
2. 1変数の状況と把握
   - （1）可視化の活用
   - （2）代表値の活用
3. ビジネスにおける比較
   - （1）概要
   - （2）活用

15

1.データの種類とは

分析とは何か

収集した情報の整理，加工，取捨選択を経て分析するプロセスのこと

適切なデータ分析により，数値にもとづく合理的な意思決定が可能となるほか，今まで気づけなかった課題やチャンスに気づきやすくなる

16

1.データの種類とは

データ分析のメリット

- データドリブン（Data Driven）が可能になる
- 迅速な意思決定が可能になる
- 新たなビジネスチャンスを発見できる

17

1.データの種類とは

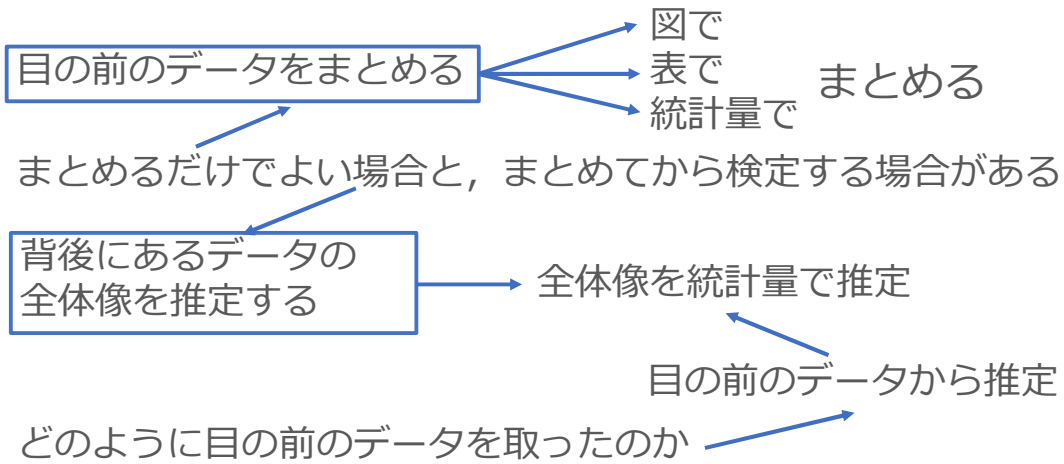
データ分析に用いられる主な10の手法

- バスケット分析
- アソシエーション分析
- クロス集計
- 因子分析
- クラスター分析
- 決定木分析
- ABC分析
- ロジスティック回帰分析
- 主成分分析
- グレイモデル

18

1.データの種類とは

統計的処理の方針



19

2. 1変数の状況と把握　（1）可視化の活用

1次元データを扱う

- 体位のデータ（体重，身長，その他）
- 医学データ（血圧，血糖値，○○値）
- 国民の所得
- 試験の点数
- 工業製品，農業生産物の質量など
- その他，数値の集合が統計の中心

押さえるポイント

全体のようすをつかむ　データの可視化 → 分布のかたち
統計量を知る　代表値：平均，分散，四分位数

20

2. 1変数の状況と把握　（1）可視化の活用

データの整理

- 性質に注目すると以下の３種類に分けられる
  1. カテゴリカルデータ（名義データ）
  2. 順位データ（順序データ）
  3. 計算データ（数量データ）

ｱ．体重　ｲ．身長　ｳ．性別　Ⅰ．A・B・C・Dの成績
ｵ．100点満点のテストの得点　ｶ．あなたは走るのは速いですかという質問に対して，速い・ふつう・遅いの回答選択肢が用意されている場合　ｷ．100m走のタイム　ｸ．給料　ｹ．ある道路を自動車が通った台数　ｺ．年齢
ｻ．体重を重い・普通・軽いに分けた場合　ｼ．好きな食べ物を聞いた場合

21

2. 1変数の状況と把握　（1）可視化の活用

データの整理

- データの性質を基礎にした分類以外の分類でよく使われるもの
  1. 時系列データ：時間の流れとともに観測して得られるデータ
  2. 横断面データ：一時点のみにおけるデータ

  縦断的研究，横断的研究

ｱ．あるクラスで一斉に行ったテストの結果　ｲ．A君の前期，後期の成績
ｳ．A君の前期の統計学，環境学，論理学の成績　Ⅰ．東京における1年間365日の気温　ｵ．平成28年4月1日の全国各地の気温
ｶ．4月に行った学生健康診断の結果　ｷ．ある人の過去10年間の健康診断の結果

22

2. 1変数の状況と把握　（1）可視化の活用

重要な統計量

- 平均（mean）：$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$
- 分散（variance）：$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)^2$
- 標準偏差（standard deviation，sd）：$\sigma = \sqrt{\sigma^2}$
- 四分位数（Quantile）：Q_1，Q_2，Q_3
- メジアン，中央値（median）：Q_2

Averageも平均を表す概念だが、意味がやや広く、メジアン等の「真ん中」をも指す

23

2. 1変数の状況と把握　（1）可視化の活用

データの種類

- 質的変量（質的変数）　性別，学年，出身地など
- 量的変量（量的変数）
  - 離散変量（離散変数）　家族の人数，テストの点数など
  - 連続変量（連続変数）　身長，体重など

質的変量と量的変量

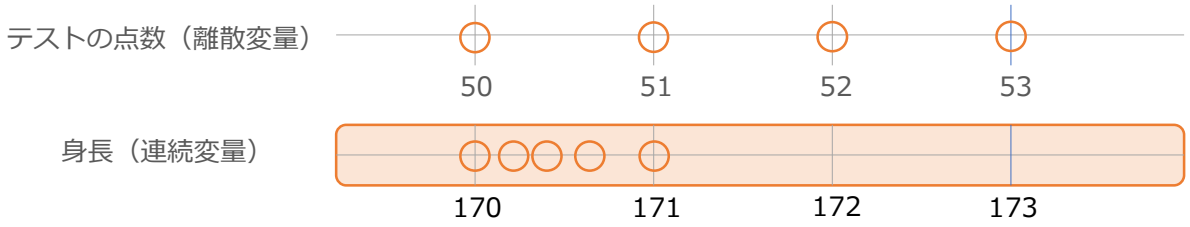
- 数値が量的な変量を持つ変量を量的変量，意味を持たないものを質的変量という。
- 質的変量か量的変量かを見分けるには，「算術平均をとって意味があるか」を考えてみる方法がある。

24

2. 1変数の状況と把握　（1）可視化の活用

離散変量と連続変量

- **離散変量**は家族の人数やテストの点数など，とびとびの値しかとらない変数である。
- 一方，身長と体重などは正確に測ろうとする場合，無限に細かい数値になる。（身長171.2865…cm）このような変量は**連続変量**である。



データの種類によって，まとめ方が異なる

25

2. 1変数の状況と把握　（1）可視化の活用

データの尺度

- データの分類方法としては，**尺度**による分類方法もある。

- 質的変量
  - 名義尺度
  - 順序尺度
- 量的変量
  - 間隔尺度
  - 比例尺度

※離散変量と連続変量のそれぞれに，間隔尺度と比例尺度のものがある

尺度による分類

質的変量	名義尺度
	順序尺度
離散変量 量的変量 連続変量	間隔尺度 比例尺度

26

2. 1変数の状況と把握　（1）可視化の活用

データの尺度

i. 名義尺度（性別，出身地など）
データ同士を区別するためにつけたもの。性別で，男ー1,女ー2などとしているが，男女を入れ替えても問題ない。

ii. 順序尺度（テストの順位，成績評価など）
テストの順位や成績評価など，順番に意味があるものである。これは，入れ替えることはできない。

iii. 間隔尺度（テストの点数，日付など）
テストの点数のように，順番に意味があり，さらにそれが等間隔に並んでいるもの。比例尺度との違いは，ゼロが絶対的な意味を持つかどうか。

iv. 比例尺度（身長，体重，家族の人数など）
比例尺度ともいう。体重40kgは20kgの2倍というように，比にも意味がある。

27

2. 1変数の状況と把握　（2）代表値の活用

ライブラリのインポート

ライブラリ（モジュール）
プログラムファイルで、複数の関数が定義されている。

- numpy
  高度な数値計算と科学技術計算を支援するために設計された強力なライブラリ
  （多次元配列，ブロードキャスト，数学関数，など）
- pandas
  データ操作と分析のための効果的なデータ構造を提供するライブラリ
  （データフレーム，データ操作，時系列データ処理，など）

28

2. 1変数の状況と把握　（2）代表値の活用

ライブラリのインポート

```
# pandas を pd という名前でインポート
import pandas as pd

# numpy を np という名前でインポート
import numpy as np
```

＃（ハッシュ記号）をつけることで、プログラムの内容を説明する「コメント」を記述できます

以降は ”pd.関数名”, ”np.関数名” と入力すればそれぞれの機能が使える

基本的に「import ライブラリ名」のように読み込む
「from ライブラリ名 import モジュール名.関数名」でもOK

29

2. 1変数の状況と把握　（2）代表値の活用

ライブラリのインポート

```
import numpy as np
import pandas as pd

# google.colabからファイルをインポート
from google.colab import files

# uploadするファイルを選択できるインタフェースを表示させる
uploaded = files.upload()
```

演習資料.zipの中のsample_cross.csvを選択してください

30

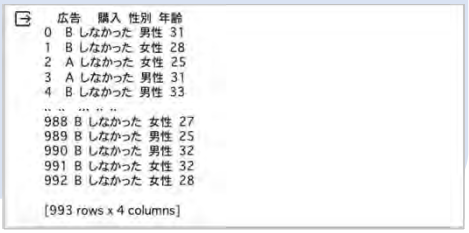
2. 1変数の状況と把握　（2）代表値の活用

ライブラリとデータのインポート

```
import io

# Pandasのread_csv関数を使って、csvファイルを読み込み、
  変数ad df に格納
ad_df =
pd.read_csv(io.BytesIO(uploaded['sample_cross.csv']))

# ad_dfをprint関数で表示させる
# print関数は()内を表示させる関数
print(ad_df)
```



	広告	購入	性別	年齢
0	B	しなかった	男性	31
1	B	しなかった	女性	28
2	A	しなかった	女性	25
3	A	しなかった	男性	31
4	B	しなかった	男性	33
...	...	...	...	...
988	B	しなかった	女性	27
989	B	しなかった	男性	25
990	B	しなかった	男性	32
991	B	しなかった	女性	32
992	B	しなかった	女性	28

[993 rows x 4 columns]

31

2. 1変数の状況と把握　（2）代表値の活用

ライブラリとデータのインポート

```
# ageという変数を用意
# numpyの中にあるarray関数（配列）を使って、先ほど定義した
  ad_dfの中の[年齢]というカラムから100件データを並べる

age = np.array(ad_df['年齢'])[:100]
age
```

```
array([31, 28, 25, 31, 33, 31, 30, 30, 24, 30, 22, 23, 31, 25, 22, 23, 27
       27, 27, 27, 29, 32, 32, 32, 26, 26, 29, 26, 25, 25, 28, 30, 24, 30,
       28, 30, 27, 25, 32, 32, 28, 29, 30, 30, 32, 23, 31, 25, 32, 25, 31,
       31, 30, 26, 31, 28, 28, 27, 28, 29, 28, 23, 24, 23, 27, 23, 28, 32,
       30, 24, 23, 29, 24, 31, 28, 25, 31, 33, 31, 30, 30, 24, 30, 22, 23,
       31, 25, 22, 23, 27, 27, 27, 27, 29, 32, 32, 32, 26, 26, 29])
```

32

2. 1変数の状況と把握　（2）代表値の活用

ライブラリとデータのインポート

```
# 変数age_df にPandasのDataFrameというデータ構造で格納
age_df = pd.DataFrame({'年齢':age})

# age_df を表示
age_df
```

DataFrameは二次元配列でデータを表示させる記述です
各レコードには「インデックス」という番号が表示されます

	年齢
0	31
1	28
2	25
3	31
4	33
...	...
95	32
96	32
97	26
98	26
99	29

100 rows × 1 columns

33

2. 1変数の状況と把握　（2）代表値の活用

データの中心の指標（平均値：mean）

```
# sum関数で求めた（合計）をlen関数で求めた配列数で割る＝算術平均の計算
sum(age) / len(age)
# 上記と同じ作業をNumpyのmean関数でも求めることができる
np.mean(age)
# 先ほどデータフレームage_dfに入れた関数を平均するというコード
age_df.mean()
```

```
27.8

27.8

年齢    27.8
dtype: float64
```

34

2. 1変数の状況と把握　（2）代表値の活用

データの中心の指標（中央値：median）

データを大きさの順に並べた時にちょうど中央に位置する値

```
# Numpyのsort関数（小さい順に並ぶ）でageを読み込み、変数sorted_ageに格納
sorted_age = np.sort(age)
# print()関数と同じ sorted_ageを表示させる
sorted_age
```

```
array([22, 22, 22, 22, 23, 23, 23, 23, 23, 23, 23, 23, 23, 24, 24, 24, 24,
       24, 24, 25, 25, 25, 25, 25, 25, 25, 25, 25, 25, 26, 26, 26, 26, 26,
       27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 27, 28, 28, 28, 28, 28, 28,
       28, 28, 28, 28, 29, 29, 29, 29, 29, 29, 29, 30, 30, 30, 30, 30, 30,
       30, 30, 30, 30, 30, 30, 31, 31, 31, 31, 31, 31, 31, 31, 31, 31,
       31, 31, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 33, 33])
```

35

2. 1変数の状況と把握　（2）代表値の活用

データの中心の指標（中央値：median）

```
n = len(sorted_age)                # 変数nにsorted_ageの個数を入れる
# ifはexcelのifと同義で、論理式を分岐させる指示
# == 2つの値を比較して等しいかどうかを調べる
if n % 2 == 0:                     # %は余りを表示させる数式
   m0 = sorted_age[n//2 -1]        # //は切り捨ての除算
   m1 = sorted_age[n//2]           # 偶数、奇数の判定
   median = (m0 + m1) / 2
else:
    median = sorted_age[(n+1)//2 - 1]
median
```

```
28.0
```

36

2. 1変数の状況と把握　（2）代表値の活用

データのばらつきの指標（分散と標準偏差）

平均が40歳でも,全員が40歳の場合のデータと,0歳が半分、80歳が半分のデータでは全く異なる。ばらつきを求めるために**偏差(deviation)**を計算する。

```
# Numpyのmean（平均）を求める
mean = np.mean(age)
# ageから平均を引き、平均との差＝ばらつき（偏差）を求める
deviation = age - mean
deviation
```

```
array([ 3.2,  0.2, -2.8,  3.2,  5.2,  3.2,  2.2,  2.2, -3.8,  2.2, -5.8,
       -4.8,  3.2, -2.8, -5.8, -4.8, -0.8, -0.8, -0.8, -0.8,  1.2,  4.2,
        4.2,  4.2, -1.8, -1.8,  1.2, -1.8, -2.8, -2.8,  0.2,  2.2, -3.8,
        2.2,  0.2,  2.2, -0.8, -2.8,  4.2,  4.2,  0.2,  1.2,  2.2,  2.2,
        4.2, -4.8,  3.2, -2.8,  4.2, -2.8,  3.2,  3.2,  2.2, -1.8,  3.2,
        0.2,  0.2, -0.8,  0.2,  1.2,  0.2, -4.8, -3.8, -4.8, -0.8, -4.8,
        0.2,  4.2,  2.2, -3.8, -4.8,  1.2, -3.8,  3.2,  0.2, -2.8,  3.2,
        5.2,  3.2,  2.2,  2.2, -3.8,  2.2, -5.8, -4.8,  3.2, -2.8, -5.8,
       -4.8, -0.8, -0.8, -0.8, -0.8,  1.2,  4.2,  4.2,  4.2, -1.8, -1.8,
        1.2])
```

37

2. 1変数の状況と把握　（2）代表値の活用

データのばらつきの指標（分散と標準偏差）

```
# ageのdfのコピーを作る
summary_df = age_df.copy()
# int型とタイプを指定
summary_df['偏差'] = deviation.astype(int)
summary_df
```

シングルクォーテーションを記載することで日本語で記述できる！

	年齢	偏差
0	31	3
1	28	0
2	25	-2
3	31	3
4	33	5
...	...	...
95	32	4
96	32	4
97	26	-1
98	26	-1
99	29	1

100 rows × 2 columns

38

2. 1変数の状況と把握　（2）代表値の活用

データのばらつきの指標（分散と標準偏差）

```
summary_df.mean()
np.mean(deviation **2)
```

```
9.7
```

```
# Numpyのvar関数でも同様に分散を求めることができる。
np.var(age)
```

```
9.7
```

39

2. 1変数の状況と把握　（2）代表値の活用

データのばらつきの指標（分散と標準偏差）

```
# Pandasのvarを使って分散を計算してみると…
age_df.var()
```

```
年齢      9.79798
dtype: float64
```

40

2. 1変数の状況と把握　（2）代表値の活用

データのばらつきの指標（分散と標準偏差）

```
summary_df['偏差二乗'] = np.square(deviation)
summary_df

summary_df.mean()
```

```
年齢     27.80
偏差      0.25
偏差二乗    9.70
dtype: float64
```

	年齢	偏差	偏差二乗
0	31	3	10.24
1	28	0	0.04
2	25	-2	7.84
3	31	3	10.24
4	33	5	27.04
...	...	...	...
95	32	4	17.64
96	32	4	17.64
97	26	-1	3.24
98	26	-1	3.24
99	29	1	1.44

100 rows × 3 columns

41

2. 1変数の状況と把握　（2）代表値の活用

データのばらつきの指標（分散と標準偏差）

```
import statistics
import math

print(statistics.mean(age))
print(sum(age) / len(age))
print(statistics.median(age))
print(statistics.mode(age))
print(statistics.variance(age))
print(statistics.pstdev(age))
```

```
27
27.8
28.0
30
9
3.0
```

42

3. ビジネスにおける比較　（1）概要

ABテスト

- AパターンとBパターンでどちらが効果があるのかをテストする
  ＝**複数のものを比較するテスト**
- Webマーケティングの領域で非常に頻繁に使われている

例）Webサイトのデザイン，インターネット広告，
ランディングページ最適化，メール配信のセグメント

43

3. ビジネスにおける比較　（1）概要

カイ二乗検定（独立性の検定：test for independence）

- 2つの変数XとYについて，関係があるのか，それとも独立であるのか
  - 帰無仮説H_0：属性間には関係がない「XとYは独立である」
  - 対立仮説H_1：属性間には関係がある「XとYは独立ではない」

- 独立性の検定にはカイ二乗分布が使われるのでカイ二乗検定（chi-square test）と呼ばれる

  大前提：カテゴリカルデータ（名義データへの適用）

44

3. ビジネスにおける比較　（1）概要

帰無仮説と対立仮説

- 帰無仮説（null hypothesis）H_0：
  「有意差がない」という仮説
  「無に帰すことも予定している」仮説であり，通常は否定したい仮説を設定する

- 対立仮説（alternative hypothesis）H_1：
  「有意差がある」という仮説
  帰無仮説が間違っていると確信されたとき（棄却された）に採用される

45

3. ビジネスにおける比較　（2）活用

ABテスト（仮説を立てる）

例 1

ある商品の広告プランとしてAとBがあり，どちらがより購買意欲につながっているのか

- もし広告の種類と購入の有無が独立なら購入の割合に変化はない
- そうでないなら，購入の割合に差が出るはず

→ 独立性の検定が使える

46

3. ビジネスにおける比較　（2）活用

ABテスト

例１

ある商品の広告プランとしてAとBがあり，どちらがより購買意欲につながっているのか

	購入した	購入しなかった	合計
広告A	60	1,000	1,060
広告B	40	400	440
合計	100	1,400	1,500

47

3. ビジネスにおける比較　（2）活用

ABテスト（クロス集計表を作成する）

```
# pandas を pd という名前でインポート
import pandas as pd
# scipy から 一部分（stats）のみインポート

from scipy import stats
#カラム（columns。縦列）のタイトルを付与。
df = pd.DataFrame([[60, 1000], [40, 400]],
                  index=['A', 'B'], columns=['購入した', '購入していない'])
df
```

	購入した	購入していない
A	60	1000
B	40	400

48

3. ビジネスにおける比較　（2）活用

ABテスト（検定を行う）

```
#カイ二乗検定で検定していくため、以下のとおり変数を指定
chi2, p, dof, exp = stats.chi2_contingency(df, correction=False)

print("期待度数", "¥n", exp)
print("自由度", "¥n", dof)
print("カイ二乗値", "¥n", chi2)
print("p値", "¥n", p)
```

＼マークには注意！
Windowsでは円記号「￥」で、
Macは「＼」記号で表示されます

```
期待度数
 [[ 70.66666667 989.33333333]
 [ 29.33333333 410.66666667]]
自由度
 1
カイ二乗値
 5.880911541288903
p値
 0.015305895674955605
```

結果はカイ二乗値(chi2)，p値(p)，自由度(dof)，期待度数(exp)で出力される

標準だとイェイツの修正が入るので，correction=Falseにして，補正が入らないように設定

49

3. ビジネスにおける比較　（2）活用

ABテスト（期待度数について）

例１

ある商品の広告プランとしてAとBがあり，どちらがより購買意欲につながっているのか

来客数の合計が1,500名
100名が商品を購入している
広告AもBも1/15の割合で購買意欲につながっている

期待値

A:1060*(1/15)=70.7
B:440*(1/15)=29.3

$$\frac{(\text{観測データ}-\text{期待度数})^2}{\text{期待度数}}$$

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

50

3. ビジネスにおける比較　（2）活用

ABテスト（実行結果の確認）

例１

ある商品の広告プランとしてAとBがあり，どちらがより購買意欲につながっているのか

```
chi2, p, dof, exp = stats.chi2_contingency(df, corre

print("期待度数", "\n", exp)
print("自由度", "\n", dof)
print("カイ二乗値", "\n", chi2)
print("p値", "\n", p)
```

```
期待度数
 [[ 70.66666667 989.33333333]
 [ 29.33333333 410.66666667]]
自由度
 1
カイ二乗値
 5.880911541288903
p値
 0.015305895674955605
```

自由度＝（行数-1）＊（列数-1）

自由度1の時のカイ二乗分布の値
5%で3.84，1%で6.63，0.5%で7.88

5.88は3.84を上回っているが6.63を下回っている

AとBが同じ前提の時
この事象は5%以下の確率でしか起きない

51

3. ビジネスにおける比較　（2）活用

ABテスト（実行結果の確認）

例１

ある商品の広告プランとしてAとBがあり，どちらがより購買意欲につながっているのか

```
chi2, p, dof, exp = stats.chi2_contingency(df, correction=False)

print("期待度数", "\n", exp)
print("自由度", "\n", dof)
print("カイ二乗値", "\n", chi2)
print("p値", "\n", p)
```

```
期待度数
 [[ 70.66666667 989.33333333]
 [ 29.33333333 410.66666667]]
自由度
 1
カイ二乗値
 5.880911541288903
p値
 0.015305895674955605
```

p値（p-value）
帰無仮説を考えた時にその結果が出る確率
（有意水準と照らし合わせるための数値）

有意水準 ＞ p値 ＝有意差がある

52

3. ビジネスにおける比較　（2）活用

ABテスト

例２

Webサイトから商品購入を促すバナー（画像）を
２種類用意し，どちらが良いかテストすることにした

	バナーA	バナーB
クリック数	10,000	10,000
コンバージョン数	400	340
コンバージョン率	???	???

※コンバージョン率：アクセスしてきたユーザーのうち、どのくらいがコンバージョンに至ったかを示す数値
コンバージョン：（訪問者がWebサイトの）目標としているアクションを起こしてくれた状態のこと

53

3. ビジネスにおける比較　（2）活用

ABテスト（仮説を立てる）

例２

Webサイトから商品購入を促すバナー（画像）を
２種類用意し，どちらが良いかテストすることにした

- H_0:バナーAとバナーBにはコンバージョン数の差異を決定づける明らかな差が「ない」（独立である：関連がない）
- H_1:バナーAとバナーBにはコンバージョン数の差異を決定づける明らかな差が「ある」（独立ではない：関連がある）

仮説：両バナーにおいてコンバージョン数の差異を決定づける明らかな差が「ある」に違いない

54

3. ビジネスにおける比較　（2）活用

ABテスト（ABテストができるようクロス集計を行う）

例２

Webサイトから商品購入を促すバナー（画像）を
２種類用意し，どちらが良いかテストすることにした

	good click	no good click	total
バナーA	?	?	?
バナーB	?	?	?
合計	?	?	?

※コンバージョンがあったクリックとなかったクリックに分ける必要がある

55

3. ビジネスにおける比較　（2）活用

ABテスト

例２

Webサイトから商品購入を促すバナー（画像）を
２種類用意し，どちらが良いかテストすることにした

	good click	no good click	total
バナーA	400	9,600	10,000
バナーB	340	9,660	10,000
合計	740	19,260	20,000

56

3. ビジネスにおける比較　（2）活用

ABテスト（クロス集計表を作成する）

```
import pandas as pd
from scipy import stats

df = pd.DataFrame([[400, 9600], [340, 9660]],
                  index=['バナーA', 'バナーB'], columns=['goog click', 'no
good click'])
df
```

	goog click	no good click
バナーA	400	9600
バナーB	340	9660

57

3. ビジネスにおける比較　（2）活用

ABテスト（検定を実施し実行結果を確認する）

```
chi2, p, dof, exp = stats.chi2_contingency(df, correction=False)

print("期待度数", "¥n", exp)
print("自由度", "¥n", dof)
print("カイ二乗値", "¥n", chi2)
print("p値", "¥n", p)
```

```
期待度数
 [[ 370. 9630.]
 [ 370. 9630.]]
自由度
 1
カイ二乗値
 5.051780752715333
p値
 0.02460064285694141
```

58