第2回
pandasによるデータの集計と可視化1

目次

- 演習で使用するモジュール
- pandas概要
- 事例：アンケートデータの集計
- Googleフォームにおける単純集計
- データをダウンロードしてExcelで集計
- サンプルデータ
- データのGoogle Colabへのアップロード
- 必要なモジュールのインポート
- データの取り込み
- 変数と尺度
- 特定の行や列を取り出す
- 質的変数（単数回答）の単純集計
- グラフの理想と現状
- 理想的なクオリティを実現するグラフ記述

演習で使用するモジュール

- 下記の表の赤文字のモジュールを使用します。

モジュール名	概要
matplotlib	グラフなどの可視化モジュール
seaborn	matplotlibの見栄えをより綺麗にするモジュール
NumPy	計算を効率的に行うためのモジュール
SymPy	数式・記号計算用モジュール
pandas	データ解析支援モジュール（Excelファイルの読み書きが可能）
openpyxl	Excelファイルの読み書きに特化したモジュール
xlwings	Excelアプリを直接制御できるモジュール
scipy	NumPyを利用した数値解析モジュール
scikit-learn	機械学習モジュール
NetworkX	グラフ・ネットワーク計算と可視化モジュール
Basemap	地図描画モジュール

pandas概要（1）

- Python Data Analysis Library
  - Pythonでデータ分析をする際に必須のライブラリ
    - Excelのようなシート（行・列）形式のデータを集計する様々な機能が備わっている
  - Pandasで使用されるオブジェクト型
    - シリーズ（Series）型
      - リストのような1次元の複数の要素からなる型

愛知県
静岡県
長野県
山梨県

    - データフレーム（DataFrame）型
      - 行と列からなる（Excelのシートのような）2次元の複数の要素からなる型

	項目A	項目B
愛知県	3463	4234
静岡県	12566	5335
長野県	2435	353
山梨県	15353	2444

pandas概要(2)

- pandasモジュールのインポート

```
import pandas as pd
```

pandasの別名としてpdを設定

- Series型オブジェクトの作成例

```
s = pd.Series(["愛知県","静岡県","長野県","山梨県"])
print(s)
```

```
0   愛知県
1   静岡県
2   長野県
3   山梨県
dtype: object
```

- DataFrame型オブジェクトの作成例

列名　　行名

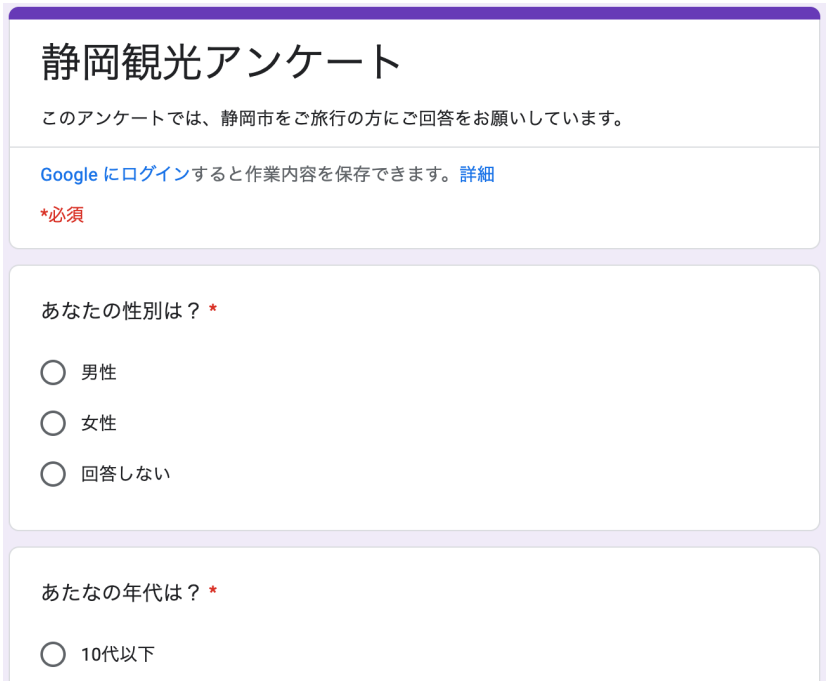
```
df = pd.DataFrame([[1, 2], [3, 4]], columns=["項目A", "項目B"], index=["A市", "B市"])
print(df)
```

```
    項目A  項目B
A市    1    2
B市    3    4
```

2次元リスト

事例:アンケートデータの集計

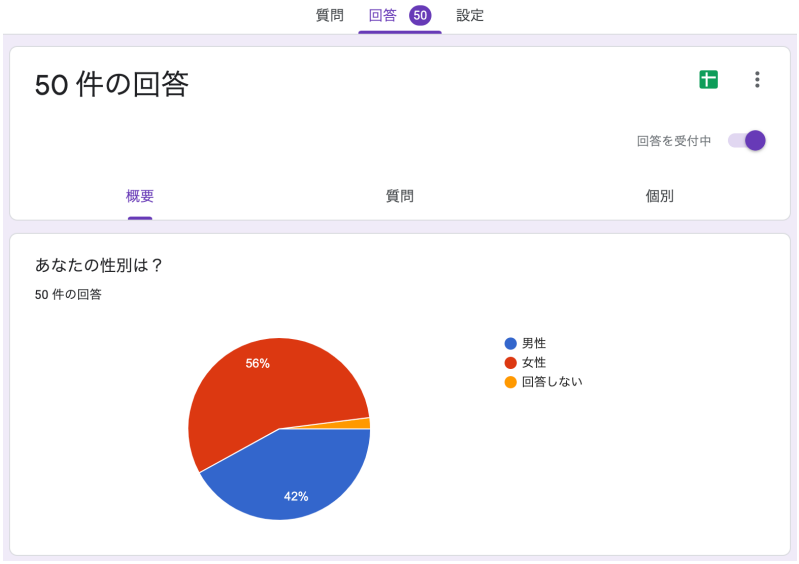
- シナリオ:Googleフォームでアンケートを収集したと仮定します。



https://forms.gle/3p2RTx52FKfhLF5A8

Googleフォームにおける単純集計

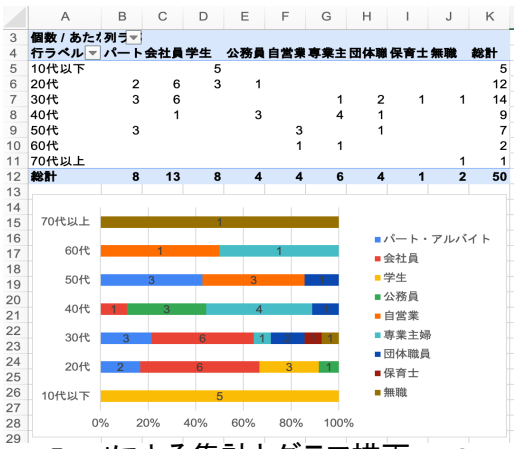
- 単純集計＝Grand Total（GT）
  - 単純集計であれば、Googleフォームで簡易集計により自動グラフ化が可能です。
  - しかし、見た目の細かな調整ができなかったり、クロス集計ができなかったりといったデメリットがあります。



データをダウンロードしてExcelで集計

- Googleフォームのデータは Excel形式でダウンロードして集計することも可能です。
  - Excelのピボットテーブル機能によってクロス集計も可能です。
  - 集計結果からグラフ作成が可能です。
  - しかし・・・
    - クロス集計の項目を様々に変えるには煩雑なマウス操作が必要、または、あらかじめ集計項目数分のクロス集計用のシートを作成しておく必要がある。
    - クロス集計の項目を様々に変えてグラフ化する場合、グラフのデザインや見た目、サイズなどを揃えるのが面倒。

- Pythonのpandasモジュールを活用する
  - 利点
    - 単純集計やクロス集計が自動化できる
    - グラフの作成が（デザインや見た目、サイズなどを揃えて）自動化できる



Excelによる集計とグラフ描画

サンプルデータ

- デスクトップ上に用意したデータファイルを使用します。
  - 「python.xlsx」を開いて内容を確認します。
  - このファイルは、先ほど紹介したシナリオのGoogleフォームのデータを、Excel形式でダウンロードしたものです。

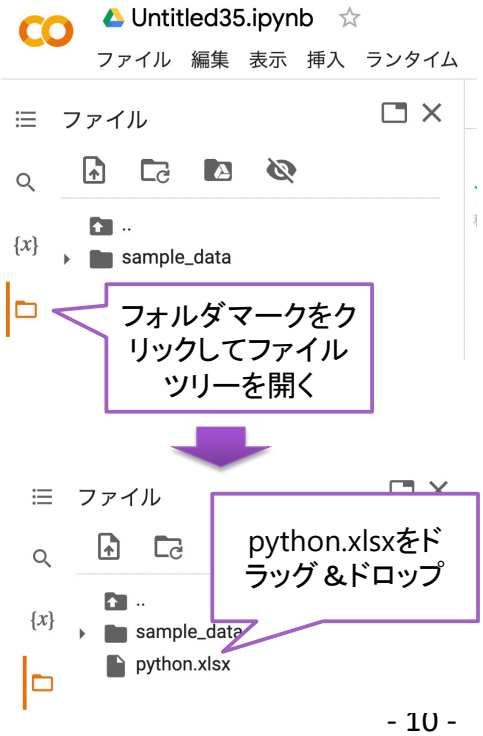
	A	B	C	D	E	F	G
1	タイムスタンプ	あなたの性	あなたの年	あなたの職	あなたの居	今回の静岡観光の目的は？（複数選択可）	今回の静岡
2	9/24/2024 14:54:10	男性	20代	学生	静岡県	温泉, まち歩き	5
3	9/24/2024 14:54:34	男性	10代以下	学生	神奈川県	グルメ, ショッピング, 温泉	4
4	9/24/2024 14:56:16	女性	30代	会社員	東京都	グルメ, ショッピング, まち歩き, 美術館・博物館等	5
5	9/24/2024 14:56:36	男性	40代	団体職員	神奈川県	温泉, ドライブ・ツーリング	4
6	9/24/2024 14:57:07	女性	20代	公務員	千葉県	グルメ, ショッピング, 名所旧跡の観光, テーマパーク	3
7	9/24/2024 14:57:38	男性	50代	自営業	千葉県	温泉, 自然鑑賞, 美術館・博物館等, 行祭事・イベント	4
8	9/24/2024 14:58:32	女性	50代	パート・ア	静岡県	グルメ, ショッピング, 温泉, 美術館・博物館等	4
9	9/24/2024 14:59:20	女性	20代	学生	静岡県	グルメ, 温泉, まち歩き	4
10	9/24/2024 15:00:00	回答しない	30代	無職	山梨県	温泉, 行祭事・イベント	3
11	9/24/2024 15:00:20	女性	10代以下	学生	神奈川県	グルメ, ショッピング, まち歩き, 自然鑑賞	4
12	9/24/2024 15:01:12	男性	30代	会社員	東京都	グルメ, 温泉, 美術館・博物館等	2
13	9/24/2024 15:01:35	女性	50代	パート・ア	長野県	グルメ, ショッピング, 行祭事・イベント	5
14	9/24/2024 15:01:55	男性	60代	自営業	愛知県	温泉, ドライブ・ツーリング	5
15	9/24/2024 15:02:46	女性	20代	パート・ア	千葉県	グルメ, ショッピング, まち歩き	4
16	9/24/2024 15:03:04	男性	30代	会社員	東京都	温泉	4
17	9/24/2024 15:03:27	男性	40代	公務員	埼玉県	温泉, 美術館・博物館等	3

python.xlsxの例

データのGoogle Colabへのアップロード

- python.xlsxをGoogle Colabにアップロードして、読み込めるようにします。

- 手順
  - Google Colab左のフォルダマークをクリックする
  - ファイルツリーが開くので、デスクトップのpython.xlsxをドラッグ &ドロップしてアップロードする
  - アップロード後は、再びフォルダマークをクリックしてファイルツリーは閉じてもよい



必要なモジュールのインポート

- pandasモジュールと、matplotlib.pyplotモジュールをインポートします。また、日本語フォントをインストールします。

```
#データ解析用モジュール
import pandas as pd
#日本語フォントのインストール
!pip install japanize-matplotlib
#グラフ描画用モジュール
import matplotlib.pyplot as plt
import japanize_matplotlib
```

- 実行すると、日本語フォントのインストールが開始されます。

データの取り込み

- pandasではExcelファイルを直接読み込むことができます。
  - 新しくコードセルを追加して下記を記述します。

```
file = pd.ExcelFile("python.xlsx")
data = file.parse("フォームの回答")
data.head()
```

- コードの解説
  - ExcelFileメソッドでExcelファイルを開く
  - 開いたファイルオブジェクトのparseメソッドでシート名を指定して取り込む（DataFrame型変数として取り込める）
  - 取り込んだデータはheadメソッドで先頭の5件について表示できる

取り込んだデータには自動的に連番のindexが割り当てられます。

	タイムスタンプ	あなたの性別は?	あなたの年代は?	あなたの職業は?	あなたの居住地は?	今回の静岡観光の目的は?（複数選択可）
0	2021-09-25 14:54:10	男性	20代	学生	静岡県	温泉, まち歩き
1	2021-09-25 14:54:34	男性	10代以下	学生	神奈川県	グルメ, ショッピング, 温泉

変数と尺度

- アンケートのデータ分析で扱う変数は、4つの尺度に分けて考えることができます。

変数	尺度	大小	差	比	説明	例
質的変数	名義尺度	×	×	×	単に分類するためのラベル名	性別
	順序尺度	○	×	×	分類の大小関係や順序のみに意味を持つ変数	順位、満足度
量的変数	間隔尺度	○	○	×	データの間隔や差に意味はあるが、0が相対的な量のため2が1の2倍とは言えない変数	摂氏
	比例尺度	○	○	○	0が無い状態を表しており、差だけでなく比も意味を持つ変数	長さ、年齢

B	C	D	E	F	G	H
あなたの	あなたの	あなたの	あなたの	今回の静	今回の静	今回の静
男性	20代	学生	静岡県	温泉, まち	5	8000
男性	10代以下	学生	神奈川県	グルメ, シ	4	7500
女性	30代	会社員	東京都	グルメ, シ	5	12000
男性	40代	団体職員	神奈川県	温泉, ドラ	4	14000
女性	20代	公務員	千葉県	グルメ, シ	3	13000
男性	50代	自営業	千葉県	温泉, 自然	4	17000
女性	50代	パート・	静岡県	グルメ, シ	4	11000
女性	20代	学生	静岡県	グルメ, 温	4	6000

質的変数

量的変数

※年代などの幅のある区間として扱われる変数＝順序尺度
※満足度は、とても満足＝7、満足＝6、やや満足＝5、ふつう＝4・・・、などと数値に置き換えたとしても、各評定値が等間隔とは限らないので、順序尺度
※順序尺度を便宜的に「等間隔」であるとみなし、間隔尺度として分析する場合もある

特定の行や列を取り出す(1)

- pandasのDataFrame型の変数では特定の行や列を取り出すことができます。

	タイムスタンプ	あなたの性別は？	あなたの年代は？	あなたの職業は？	あなたの居住地は？	今回の静岡観光の目的は？（複数選択可）	今回の静岡観光の満足度は？	今回の静岡観光の１人あたりの宿泊費は？（単位は円、数字のみでお答えください。例：15000）
0	2024-09-24 14:54:10	男性	20代	学生	静岡県	温泉, まち歩き	5	8000
1	2024-09-24 14:54:34	男性	10代以下	学生	神奈川県	グルメ, ショッピング, 温泉	4	7500

列名

0行目

1行目

- 特定の列を取り出す
  - 変数[“列名”]
  - 変数.列名

取り出した特定の行や列は、Series型というデータ型なります

```
#タイムスタンプの列を取り出し
x = data["タイムスタンプ"]
print(x)
x = data.タイムスタンプ
print(x)
```

特定の行や列を取り出す(2)

- 行インデックス名や列ラベル名を指定して取り出す(行インデックス名が行番号の場合は行番号で指定)
  - 変数.loc[行インデックス名, 列ラベル名]

```
#0行目の性別を取り出し
x = data.loc[0, "あなたの性別は？"]
print(x)
#0行目の性別と満足度を取り出し（リストで指定）
x = data.loc[0, ["あなたの性別は？","今回の静岡観光の満足度は？"]]
print(x)
```

- 行番号と列番号を指定して取り出す
  - 変数.iloc[行番号, 列番号]

行番号や列番号を「:」とすると、全行・全列が取り出せます。

```
#0行目0列目を取り出し
x = data.iloc[0, 0]
print(x)
#0行目の0列目から3列目までを取り出し（スライス指定）
x = data.iloc[0, 0:4]
print(x)
#0行目の1列目と3列目を取り出し
x = data.iloc[0, [1, 3]]
print(x)
```


特定の行や列を取り出す(3)

- 列の値を検索して特定の行を抽出する
  - 変数[変数[“列ラベル名”] == “検索値”]
- 列の値を検索して特定の行を抽出する(複数条件or一致)
  - 変数[(変数[“列ラベル名1”] == “検索値1”) | (変数[“列ラベル名2”] == “検索値2”)]
- 列の値を検索して特定の行を抽出する(複数条件and一致)
  - 変数[(変数[“列ラベル名1”] == “検索値1”) & (変数[“列ラベル名2”] == “検索値2”)]

```
#居住地が静岡県の行だけを取り出し（検索）
x = data[data["あなたの居住地は？"] == "静岡県"]
print(x)
#居住地が静岡県または愛知県の行だけを取り出し（検索）
x = data[(data["あなたの居住地は？"] == "静岡県") | (data["あなたの居住地は？"] == "愛知県")]
print(x)
#居住地が静岡県かつ職業が学生の行だけを取り出し（検索）
x = data[(data["あなたの居住地は？"] == "静岡県") & (data["あなたの職業は？"] == "学生")]
print(x)
```

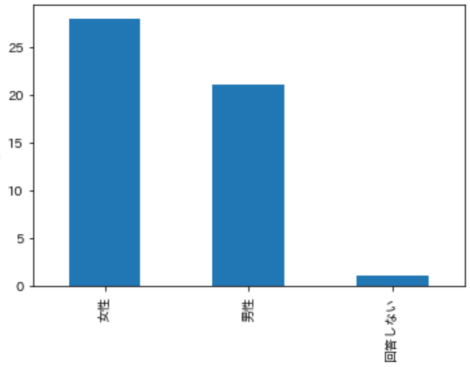
質的変数(単数回答)の単純集計(1)

- value_countsメソッドにより、質的変数(単数回答)の項目ごとに単純集計が実行できる
  - 構文 変数.value_counts()
- 集計により新たなSeries型変数を生成し、そのSeries型変数を元に縦棒グラフを描画できる(DataFrame型は行列形式、Series型はベクトル形式)
  - 構文 変数.plot.bar()

```
sei = data["あなたの性別は？"].value_counts()
print(sei)
sei.plot.bar()
plt.show()
```

集計結果は新たなSeries型変数となる

```
女性          28
男性          21
回答しない     1
```



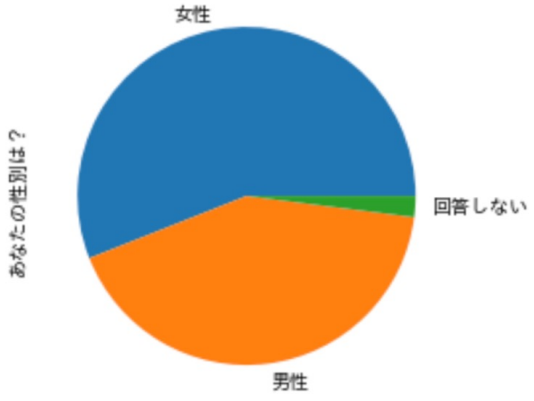
pandasでは内部的にmatplotlibを利用してグラフが描画される

plt.show()で描画を確定できる

質的変数(単数回答)の単純集計(2)

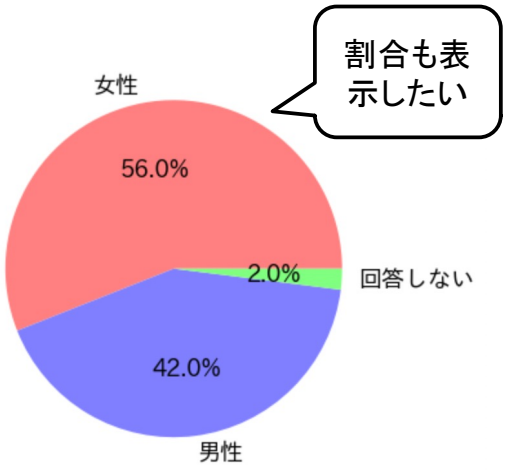
- 棒グラフbar以外にも、円グラフpieなど様々な種類のグラフを描くことができる
  - bar(縦棒グラフ)、barh(横棒グラフ)、pie(円グラフ)、scatter(散布図)、line(折れ線グラフ)など

```
sei = data["あなたの性別は？"].value_counts()
print(sei)
sei.plot.bar()
plt.show()
sei.plot.pie()
plt.show()
```

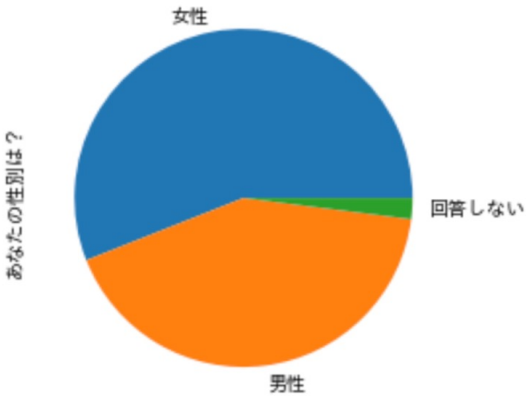


グラフの理想と現状

- グラフを、報告書に掲載できるクオリティに仕上げたい



割合も表示したい



現状

あなたの性別は?		割合
女性	28	56.0%
男性	21	42.0%
回答しない	1	2.0%
総計	50	100.0%

集計表も表示したい

理想

理想的なクオリティを実現するグラフ記述

- グラフ毎にこのようなコードを毎回書いていたら大変面倒　→ 第3回に続く

```
[8]  plt.rcParams["font.size"] = 18 #フォントサイズ設定
     pd.options.display.notebook_repr_html = True #データフレームをHTMLで綺麗に表示

     ret = data["あなたの性別は？"].value_counts()
     ret.plot.pie(autopct="%.1f%%", figsize=(6, 6), wedgeprops={"linewidth": 0, "edgecolor": "white"},
       label = "",
       colors=("#ff8080", "#8080ff", "#80ff80"))
     plt.show()

     #集計表を描く
     kensu = data["あなたの性別は？"].count() #回答件数
     ret["総計"] = kensu                      #総計行を追加
     final = pd.DataFrame(ret)                #集計結果をデータフレームに変換
     #割合列を追加して、ラムダ式で割合を計算
     final["割合"] = final["count"].map(lambda x: '{:.01f}'.format(x / kensu * 100) + "%")
     display(final) #集計表を整形して出力
```