動態マクロ経済学

Week 5

佐藤健治
sato@eco.osakafu-u.ac.jp

2020/6/5

準備運動：IPython を起動してください

```
import numpy as np

rng = np.random.default_rng(100)   # 100 は乱数の種
x = rng.choice(range(100, 500), 15)
np.mean(x)
np.median(x)
np.quantile(x, 0.9)
x > 300
x[x > 300]
```

True/False のベクトル

True/False のベクトル

```
x > 400
```

```
array([ True,  True, False, False, False, False,
       False, False, False,  True,  True, False,
       False,  True,  True])
```

```
x[x > 400]
```

```
array([406, 433, 489, 478, 416, 469])
```

True/False の足し算

True/False が四則演算で使われると，1/0 に置き換えられる。

```
True + True + False
```

```
2
```

```
True + False + False + True + True
```

```
3
```

条件を満たす要素の数・割合

- 不等式条件を満たす要素の数

```
np.sum(x > 400)
```

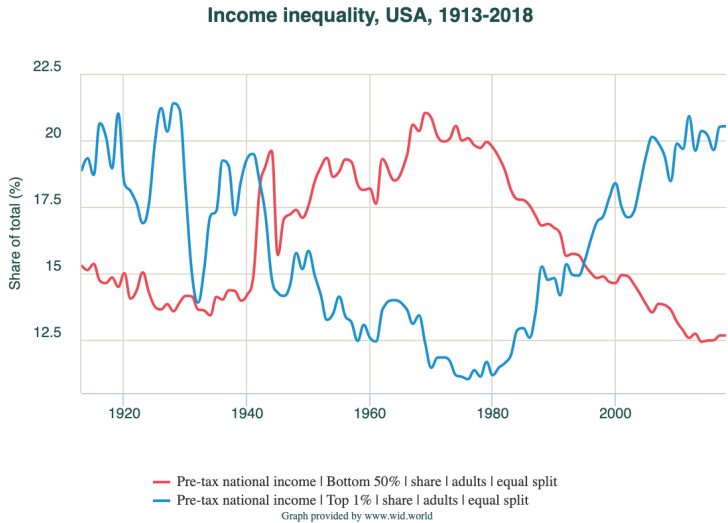
```
6
```

- 不等式条件を満たす要素の割合

```
np.mean(x > 400)
```

```
0.4
```

分配の不平等 (Source: World Inequality Database)



2011年 「Occupy Wall Street」運動

格差・不平等の指標

基本的に GDP の拡大は善き事とされる。ただし，全体の拡大から得られる利益を個々人が等しく享受している訳ではない。分配の不平等に関する議論も注目を集めているので，指標の定義を確認しておこう。

- 相対的貧困, 絶対的貧困
- ジニ係数
- Top X% 所得シェア

準備：順序統計量

格差・不平等の指標の基本は順序統計量。数値データ

$$x_1, x_2, \ldots, x_n$$

に対して，小さい順に並べ替えたものを以下のように表す。

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

$x_{(1)}$ は最小値，$x_{(n)}$ は最大値

中央値

中央値 (median) は，順序を並べ替えてちょうど真ん中に位置するデータのこと。次のように定義する。

$$\mathrm{med}(\{x_1, \ldots, x_n\}) = \begin{cases} x_{(\frac{n+1}{2})} & n \text{ が奇数のとき} \\ \frac{x_{(\frac{n}{2})} + x_{(\frac{n}{2}+1)}}{2} & n \text{ が偶数のとき} \end{cases}$$

```
np.median(x)
```

```
338.0
```

q-分位数

$0 \leq q \leq 1$ に対して，q-**分位数**と呼ばれる統計量 Q_q は次のように定義

$$\frac{x_{(i)} \leq Q_q \text{ を満たす } i \text{ の数}}{n} = q$$

中央値は $Q_{1/2}$ である。厳密な定義は難しいので省略。

```
np.quantile(x, 0.9)    # 下から 90% の値
```

```
474.4
```

格差・不平等の指標の計算

- 相対的貧困：
  - 中央値の計算
  - 中央値の半分以下の割合
- 絶対的貧困
  - 一定値以下の割合 → 計算方法は簡単なので説明しない。
- ジニ係数
  - ローレンツ曲線の計算
- Top X% 所得シェア
  - 下から数えて (100-X)% の点以上の値を「合計」して，全体の合計に占める割合を計算

相対的貧困

相対的貧困率は等価可処分所得の中央値に基づいて計算される。

$$\text{個人の等価可処分所得} = \frac{\text{属する世帯の可処分所得の総額}}{\sqrt{\text{属する世帯の人員数}}}$$

個人	世帯	可処分所得
1	A	400
2	A	300
3	A	0
4	B	1,000
5	B	0
6	C	250
7	D	150

$\Longrightarrow$

個人	等価可処分所得
1	404.145
2	404.145
3	404.145
4	707.107
5	707.107
6	250
7	150

参考) Python で上のような集約をするには？

```
import pandas as pd
frame = pd.DataFrame({
    'household': list('AAABBCD'),
    'income': [400, 300, 0, 1000, 0, 250, 150.]
}, index=range(1, 8))
frame
```

	household	income
1	A	400.0
2	A	300.0
3	A	0.0
4	B	1000.0
5	B	0.0
6	C	250.0
7	D	150.0

参考) つづき

```
income = (frame.groupby('household')
 .transform(lambda x: x.sum() / np.sqrt(x.size)))
income
```

```
        income
1   404.145188
2   404.145188
3   404.145188
4   707.106781
5   707.106781
6   250.000000
7   150.000000
```

説明していない機能を使いました。結果は Pandas の Series オブジェクトですが，NumPy 配列と同じように使えるので変換しないでおきます。

相対的貧困 ⇔ 等価可処分所得 $<$ (中央値 / 2)

```
income < np.median(income) / 2
```

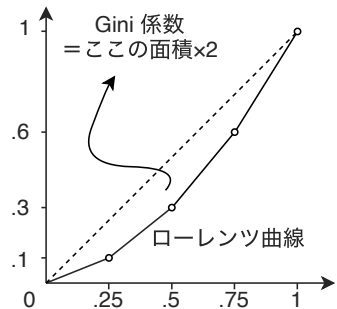
```
    income
1    False
2    False
3    False
4    False
5    False
6    False
7     True
```

True の割合を数えたものが「**相対的貧困率**」 → 各自計算してみよう。

ジニ係数

簡単な数値例 $\boldsymbol{x} = (300, 100, 400, 200)$

1. 整列 $\boldsymbol{x}_{(\cdot)} = (100, 200, 300, 400)$
2. 累積和 (cumulative sum) の計算 → $(100, 300, 600, 1000)$
3. 総和 $\sum \boldsymbol{x}$ で割る。
   $\boldsymbol{Y} = (0, 0.1, 0.3, 0.6, 1)$。ゼロを付加。
4. 累積度数 $(0, 0.25, 0.5, 0.75, 1)$ に対して $\boldsymbol{Y}$ をプロット→ ローレンツ曲線
5. 対角線（完全平等線）とローレンツ曲線で囲まれた領域の面積を求める。
6. 2 倍する



面積の計算

- ジニ係数（面積計算の部分, 5 & 6）の公式。証明は講義ノート参照。

$$G = 1 - \frac{2\left(\sum_{j=1}^{n} Y_j\right) - 1}{n}$$

- 注意。この公式は構成員全員のデータが分かっている場合の公式。
  - 実際には次のようなデータを使わなければならないことが多い。
    - 年収階級 200〜300 万円 → 20%
    - 年収階級 300〜400 万円 → 18%
    - etc.
  - 階級値でジニ係数を計算する場合も基本の考え方は変わらない。詳細は例えば，久保川・国友 『統計学』を見よ。

ジニ係数の数値例

```
xlow = rng.choice(np.arange(10, 50.), 10)
xhigh = rng.choice(np.arange(80, 200.), 10)
x = np.r_[xlow, xhigh]
x
```

```
array([ 46.,  10.,  37.,  38.,  17.,  30.,  49.,
         37.,  21.,  22., 155.,  97., 149., 148.,
        151., 172., 144., 166., 199.,  95.])
```

```
y = np.sort(x)   # Step 1
y
```

```
array([ 10.,  17.,  21.,  22.,  30.,  37.,  37.,
         38.,  46.,  49.,  95.,  97., 144., 148.,
        149., 151., 155., 166., 172., 199.])
```

ジニ係数の数値例（つづき）

```
Y = np.r_[0, y.cumsum()] / y.sum()  # Step 2, 3
Y
```

```
array([0.        , 0.00560852, 0.01514302,
       0.02692092, 0.03925967, 0.05608525,
       0.07683679, 0.09758833, 0.11890073,
       0.14469994, 0.17218172, 0.2254627 ,
       0.2798654 , 0.36062815, 0.44363432,
       0.52720135, 0.61189007, 0.69882221,
       0.79192372, 0.88839035, 1.        ])
```

```
gini = 1 - (2 * np.sum(Y) - 1) / y.size   # Step 4, 5, 6
gini
```

```
0.39189568143578235
```

トップ X% シェア

- Top percentile share
  - 所得額上位 1% の人が稼ぐ所得の，総所得に占める割合。
- Top decile shpare
  - 所得上位 10% の人が稼ぐ所得の，総所得に占める割合。
- Atkinson, Piketty and Saez (2011). "Top Incomes in the Long Run of History," *Journal of Economic Literature*.
  - この数十年で，英語圏・インド・中国で上位のシェアが急増している。

Top percentile share の数値例

さきほどの x を使いまわそう。

```
x
```

```
array([ 46.,  10.,  37.,  38.,  17.,  30.,  49.,
        37.,  21.,  22., 155.,  97., 149., 148.,
       151., 172., 144., 166., 199.,  95.])
```

Top percentile share を計算するには，データを小さい順に並べ変えて，
下から 99% の位置にあるべきデータ $Q_{0.99}$ を推定する。

$$\frac{x_{(i)} \leq Q_{0.99} \text{ を満たす } i \text{ の数}}{n} = 0.99$$

Top percentile share の数値例（つづき）

$Q_{0.99}$ は np.quantile() で計算できる。

```
np.quantile(x, 0.99)
```

```
193.86999999999995
```

あとは総和と比率計算をするだけ。

```
np.sum(x[x > np.quantile(x, 0.99)]) / np.sum(x)
```

```
0.11160964666292765
```

注意点

- 価格指数や数量指数についても同じ注意があてはまるのですが・・・・**実測は理論よりもはるかに難しいです**。
- この講義では簡単な方の「理論」を説明して，理想的な環境下での計算方法を紹介しました。
  - 計測の難しさを想像してみよう。例えば，
    - 大富豪の多くはタックスヘイブンに資産をおいているので，レポートされた納税額は実際に納税すべきだった額よりもはるかに少ないかもしれない。
    - 家計ベースで貧困を捉える指標は，家庭内での格差（例：性別・年齢にもとづく差別）などを隠してしまう。
- データに潜むバイアスについて意識的になりましょう。

技術的なコメント (See [week05.ipynb] の参考セクション)

- インタラクティブな数値計算ではない普通のプログラミングでは True/False は「条件分岐の `if-else`」や「条件付けループの `while`」と一緒に使うことが多い。
  - テキストで簡単に紹介しているので勉強しておいてください。
- ここまでくると，条件分岐で色々できそうな気がしてきます。まさにその通りで，ぜひ色々やってもらいたいのですが，1 つ注意。例えば Python + NumPy で `for` の中で `if-else` を使うようなループを書くと高い確率で実行速度が遅くなります。
- Python+NumPy を使うときの基本的な方針：
  - step-by-step の計算を積み上げていく必要があるとき以外はループを使わない
  - どうしてもループを使いたい，高速化したいときは Numba や Python 以外の選択肢を検討する（Julia, C++ などを使う）

課題

格差指標を計算してもらう Moodle 課題

- CSV データ（例：相対的貧困に使った frame データのようなデータ）を共有するので各自読みこむ。
  - 期末テストでもそういうタイプの課題を出したいので，練習です。
  - CSV データから Pandas の DataFrame を作る方法は前回の宿題を参照。
    - ちなみに，Pandas の DataFrame は色々なデータ型の NumPy 配列をひとまとめで処理するためのもの。基本は NumPy 配列と同じなので難しく考える必要はない。