

データ観察と可視化

《学修項目》

- データの種類、質的データの要約、分割表、混合行列
- 様々なデータ可視化手法（質的データを中心として）
- ビッグデータの可視化、関係性の可視化、GIS情報との融合

《キーワード》

量的変数、質的変数、群データ、群平均、総平均、分割表、クロス集計表、観測度数、周辺度数、期待度数、リスク、オッズ、棒グラフ、パレート図、層状帯グラフ、beeswarm図、ビッグデータ可視化、関係グラフ、GIST情報

《参考文献、参考書籍》

- [1] 東京大学MIセンター公開教材 「1-3 データ観察」 (http://www.mi.u-tokyo.ac.jp/pdf/1-3_data_search.pdf) 「1-5 データ可視化」 (http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf)
《利用条件CC BY-NC-SA》 (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.ja>)
- [2] 応用基礎としてのデータサイエンス（講談社 データサイエンス入門シリーズ）
(<https://www.kspub.co.jp/book/detail/5307892.html>)
- [3] データサイエンスの考え方 社会に役立つAI×データ活用のために（オーム社）
(<https://www.ohmsha.co.jp/book/9784274227974/>)
- [4] Pythonによるあたらしいデータ分析の教科書 第2版（翔泳社）
(<https://www.shoehisha.co.jp/book/detail/9784798178776>)
- [5] 数理・データサイエンス・AI公開講座（放送大学）
(https://www.ouj.ac.jp/booklet/2022/29_2022_MDS-AI.pdf)

1. データの種類、質的データの要約 [1][2]

1.1 データの種類 [1]

データは、事実や参考資料・情報であり、コンピューターで扱うことができる記号化・数値化されたものを指す。データの種類には、**量的変数**と**質的変数**の2種類があり、量的変数には比率データと間隔データが、質的変数には名義尺度と順序尺度がある。量的変数は数量で表現され、比率データと間隔データに分けられる。質的変数は数量で表現が難しく、**名義尺度**と**順序尺度**に分類される。名義尺度は同じ値か否かを表現し、順序尺度は大小関係を表現する。

データの種類

データとは、物事の推論の基礎となる事実、また参考となる資料・情報です。また、コンピューターでプログラムを使った処理の対象となる記号化・数値化された資料、とデジタル大辞泉にて説明されています。データを集計する際に、データの種類として大きく分けて「量的変数」と「質的変数」の2種類があることに注意します。

量的変数：数量で表すことができ、さらに以下のように分類することもできます。

比率データ：四則演算すべて意味がある。例：体重、年収、長さ

間隔データ：和や差はできるが、積や除算には意味がない。

例：西暦年、温度（「温度70%減」とはいわない）

質的変数：数量で表すことが困難であるもので、さらに以下のように分類することもできます。

名義尺度：同じ値か否か

例：名前、性別、職業、既婚／未婚

順序尺度：大小関係あり

例：ランキング、成績の五段階評価

データの項目はデータによって異なります。

右の例では、「地域コード」「都道府県」「市」「世帯人員」「米」「食パン」「他のパン」等の項目があり、「米」以降の項目はそれぞれ一世帯あたり年間支出金額を示しています。

まずは、比較対象の設定を的確に行なうことが重要です。たとえば、各食品の支出金額と「世帯人員」との関連性に焦点を当てるのか、それとも食品支出金額の間の関連性に興味があるのか、といったことをはっきりさせます。

地域コード	都道府県	市	世帯人員	米	食パン	他のパン	..
R01100	北海道	札幌市	2.96	30994	8496	18942	..
R02201	青森県	青森市	2.98	23773	7777	17336	..
R03201	岩手県	盛岡市	3.15	25867	8270	20622	..
R04100	宮城県	仙台市	3.00	20207	7972	18989	..
R05201	秋田県	秋田市	2.88	19508	6461	17978	..
R06201	山形県	山形市	3.19	26733	7781	18735	..
R07201	福島県	福島市	3.00	24612	7077	18422	..
R08201	茨城県	水戸市	2.90	19367	8495	17673	..
R09201	栃木県	宇都宮市	2.85	22135	9053	19055	..
R10201	群馬県	前橋市	2.81	25322	7652	22129	..
R11100	埼玉県	さいたま市	3.04	24816	9350	22858	..
R12100	千葉県	千葉市	3.00	22629	10092	22679	..
R13100	東京都	東京都区部	2.93	22412	11064	24885	..
R14100	神奈川県	横浜市	2.84	24983	10722	23457	..
..

「都道府県所在市別・家計消費データ」を加工して作成
(<https://www.nstac.go.jp/SSDSE/>)

5

東京大学 数理・情報教育研究センター 河合玲一郎 2021 CC BY-NC-SA

(http://www.mi.u-tokyo.ac.jp/pdf/1-3_data_search.pdf#page=5)

1.2 質的データ [2]

対象 i のある属性 x について観測された数値または記号をデータと呼び X_i で記す。このとき、属性 x を変数 (variable) と呼ぶ。数値で与えられるデータについて、連続的な値をとるデータを連続データ、離散的な値をとるデータを離散データと言う。

データは尺度水準によっていくつかの尺度に分類される。尺度のうち、質的データとして分類されるには、以下の2つである[2]。

- **名義尺度**：対象の観測値が（順序関係の意味のない）数値または記号の集合によって表されるとき、対象は名義尺度で観測されているといい、そのデータを名義尺度データと呼ぶ。名義尺度データにおいては、等値関係 ($=$) しか意味がなく、数値で表現されていても、その大小関係には意味がない。
- **順序尺度**：対象の観測値が順序関係の意味のある数値または記号の集合で表されるとき、対象は順序尺度で観測されているといい、そのデータを順序尺度データと呼ぶ。

1.3 質的 × 量的データの要約

ここでは、質的変数と量的変数の組のデータが与えられたとき、それらの変数間の相関関係を測ることを考える。

表1.2.2のように、 l カテゴリの質的変数 x と、量的変数 y の組データが与えられているとき、カテゴリ群を考える。このとき、群ごとに量的変数 y のデータが分けられている（ベクトル化されている）状況（表1.2.3）を考える[2]。

表1.2.2 量質混在2変数データ、表1.2.3 ℓ カテゴリ群のデータ ([2]より引用)

表 1.2.2 量質混在 2 変数データ.

	質的変数 x	量的変数 y
1	x_1	y_{11}
\vdots	\vdots	\vdots
n_1	x_1	$y_{n_1 1}$
$n_1 + 1$	x_2	y_{12}
\vdots	\vdots	\vdots
$n_1 + n_2$	x_2	$y_{n_2 2}$
\vdots	\vdots	\vdots
$\sum_{j=1}^{\ell-1} n_j + 1$	x_ℓ	$y_{1\ell}$
\vdots	\vdots	\vdots
$\sum_{j=1}^{\ell-1} n_j + n_\ell$	x_ℓ	$y_{n_\ell \ell}$

表 1.2.3 ℓ 群のデータ.

	質的変数 x			
	x_1	x_2	\cdots	x_ℓ
y_{11}	y_{12}	\cdots	\vdots	$y_{1\ell}$
y_{21}	y_{22}	\cdots	\vdots	$y_{2\ell}$
\vdots	\vdots	\ddots	\vdots	\vdots
$y_{n_1 1}$	\vdots	\vdots	\cdots	$y_{n_\ell \ell}$
$y_{n_2 2}$				
平均	\bar{y}_1	\bar{y}_2	\cdots	\bar{y}_ℓ

1.3.1 群データ、群平均、総平均

$x_j (j = 1, 2, \dots, l)$ を質的変数とし、 $y_{ij} (i = 1, 2, \dots, n_j; j = 1, 2, \dots, l)$ を群 j に属する i 番目のデータとする。このようなデータを群データと呼ぶ。このとき、群平均 \bar{y}_i およびすべてのデータの総平均 \bar{y} は、それぞれ以下で表される[2]。

$$\bar{y}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} y_{ij} \quad (j = 1, 2, \dots, \ell), \quad \bar{y} = \frac{1}{n} \sum_{j=1}^l \sum_{i=1}^{n_j} y_{ij}$$

ここで、 $n = \sum_{j=1}^l n_j$ (データ数の総和) である。

カテゴリ l が群（つまり、ある性質 l によって性格づけられている）と仮定したとき、群平均 \bar{y}_i を用いて、それぞれの群でデータの平均に差があるのか無いのかを判別できる。

1.3.2 全分散、群内分散、群間分散

群平均 \bar{y}_i および総平均 \bar{y} を用いて、全分散 s_y^2 、群内分散 w_y^2 、群間分散 b_y^2 が、それぞれ以下で定義される[2]。

$$s_y^2 = \frac{1}{n} \sum_{j=1}^l \sum_{i=1}^{n_j} (y_{ij} - \bar{y})^2,$$

$$w_y^2 = \frac{1}{n} \sum_{j=1}^l \sum_{i=1}^{n_j} (y_{ij} - \bar{y}_j)^2, \quad b_y^2 = \frac{1}{n} \sum_{j=1}^l \sum_{i=1}^{n_j} (\bar{y}_j - \bar{y})^2$$

ここで、 $n = \sum_{j=1}^l n_j$ (データ数の総和) である。

全分散 s_y^2 、群内分散 w_y^2 、群間分散 b_y^2 について等式 $s_y^2 = w_y^2 + b_y^2$ が成立する。全分散の中で群間分散が相対的に大きいということは、量的変数が質的変数に大きく影響を受けている（カテゴリによってバラツキに差がある）と考えることができる。逆に、群間分散が相対的に小さいということは、その影響が小さい（カテゴリによってバラツキにあまり差がない）と考えることができる。

この考え方に基づいて、質的変数と量的変数の関係の強さを表す指標である **相関比** η_{xy}^2 が以下で定義される [2]。相関比は0から1の間をとり、質的変数xの変化に量的変数yが影響を受けなければ0となり、関係が強い（相関がある）ほど1に近づく。

$$w^2 \quad b^2$$

1.4 質的 x 質的データの要約

1.4.1 クロス集計表と度数分布 [1]

複数の項目を組み合わせ、合計や平均、標準偏差などを集計した「**クロス集計表**」は、データの全体像を把握するのに有用であるが、データのばらつきを理解するのは困難である。そこで、データを適当な区間に分け、各区間に含まれるデータ数を表にまとめた「**度数分布**」を用いることで、データの傾向を把握することができる。度数分布は、データ数が多くても全体の傾向がわかりやすくなるため、データの解析において重要な役割を果たしている。

クロス集計表

2種類の項目を組み合わせて、合計、平均、標準偏差等を集計したものをおこなうと、「**クロス集計表**」とよび、データの全体像が把握しやすくなります。

右のクロス集計表は、都道府県ごとに一世帯当たりの年間支出金額を食品大分類ごとの合計を示したものです。前出データの「米」、「食パン」、「他のパン」は、右のクロス集計表ではすべて「穀物」に分類されています。

	穀類	魚介類	肉類	乳卵類	野菜・海	果物	...
北海道	81474	79328	83095	41262	104045	36067	..
青森県	71992	90933	83349	38677	106830	38863	..
岩手県	80203	78310	76514	51711	118250	42415	..
宮城県	70942	87815	84141	48489	120474	43636	..
秋田県	68139	84401	80686	42682	117898	41537	..
山形県	79598	74850	93770	48522	118255	47970	..
福島県	73184	76986	76085	48264	110835	49477	..
茨城県	67318	68527	75129	48425	100131	41079	..
栃木県	74050	72694	82490	46981	114270	42331	..
群馬県	77456	71940	67833	44449	105257	42123	..
埼玉県	80828	73940	88061	49560	121177	42751	..
千葉県	78500	80770	88786	50870	123233	45467	..
東京都	81177	79327	95859	50541	125815	44229	..
神奈川県	82257	83487	97515	49915	127908	46446	..
..
..

「都道府県所在市別・家計消費データ」を加工して作成
(<https://www.nstac.go.jp/SSDSE/>)

前述のクロス集計表はデータを要約する上で効果的ですが、それらの表の値を見ているだけでは、データがどのように、どの程度ばらついているかを把握するのは困難です。

まずデータの値を適当な範囲で区切って、それぞれの区間にに入るデータ数を表にします。これを度数分布とよび、データ数が多くても全体の傾向がわかりやすくなります。

各階級の上限と下限の差を階級幅、それらの中央値を階級値とよびます。たとえば「2.72～2.76」の階級の階級幅は0.04（人）、階級値は2.74（人）となります。

〈データ〉		〈度数分布〉	
都道府県	世帯人員	世帯人員	都道府県数
北海道	2.96	2.72～2.76	1
青森県	2.98	2.76～2.80	1
岩手県	3.15	2.80～2.84	5
宮城県	3.00	2.84～2.88	4
秋田県	2.88	2.88～2.92	5
山形県	3.19	2.92～2.96	6
福島県	3.00	2.96～3.00	7
茨城県	2.90	3.00～3.04	6
栃木県	2.85	3.04～3.08	3
群馬県	2.81	3.08～3.12	1
埼玉県	3.04	3.12～3.16	3
千葉県	3.00	3.16～3.20	4
東京都	2.93	3.20～3.24	1
神奈川県	2.84		
.	.		
.	.		

東京大学 数理・情報教育研究センター 河合玲一郎 2021 CC BY-NC-SA

7

(http://www.mi.u-tokyo.ac.jp/pdf/1-3_data_search.pdf#page=7)

1.4.2 質的データ間における観測度数、周辺度数、期待度数 [2]

ここでは、ともに質的な2つの変数の組のデータが与えられているときの集計と要約について考えてみる。

N 個の対象について n カテゴリの値 $x_i (i = 1, 2, \dots, n)$ のいずれかをとる変数 x と、 m カテゴリの値 $y_j (j = 1, 2, \dots, m)$ のいずれかをとる変数 y の組データが与えられているとする（表1.2.4）。このとき N 個のデータの中で $x = x_i, y = y_j$ となる対象の個数（度数）を f_{ij} で表し、表1.2.5 の形でまとめたものを分割表あるいはクロス集計表と呼ぶ[2]。分割表を用いた集計については、表中の値は度数（対象の個数）であることもあるし、全体数 N で割った比率（パーセンテージ）である場合もあるが、相対的な意味は変わらない。

表1.2.4 質的な2変数データの組情報、表1.2.5 $n \times m$ 分割表（[2]より引用）

表 1.2.4 質的な 2 変数データ.

	質的変数 x	質的変数 y
1	$x_1 \sim x_n$ のいずれか	$y_1 \sim y_m$ のいずれか
2	$x_1 \sim x_n$ のいずれか	$y_1 \sim y_m$ のいずれか
:	:	:
N	$x_1 \sim x_n$ のいずれか	$y_1 \sim y_m$ のいずれか

表 1.2.5 $n \times m$ 分割表.

変数 x	変数 y				
	y_1	y_2	\cdots	y_m	
x_1	f_{11}	f_{12}	\cdots	f_{1m}	$f_{1\cdot}$
x_2	f_{21}	f_{22}	\cdots	f_{2m}	$f_{2\cdot}$
:	:	:	\ddots	:	:
x_n	f_{n1}	f_{n2}	\cdots	f_{nm}	$f_{n\cdot}$
	$f_{\cdot 1}$	$f_{\cdot 2}$	\cdots	$f_{\cdot m}$	$f_{\cdot\cdot} (= N)$

1.4.3 観測度数、周辺度数、期待度数

2つの質的変数 x, y について、データが $n \times m$ 分割表（表1.2.5）の形で与えられているとき、 $x = x_i, y = y_j$ となる対象の個数（度数）を観測度数 f_{ij} と呼ぶ。

また、周辺度数 $f_{i\cdot}, f_{\cdot j}, f_{..}$ は、それぞれ以下で定義される[2]。周辺度数 $f_{i\cdot}, f_{\cdot j}$ は、それぞれ変数 x_i, y_j を固定した場合の度数を考えることができる。 $f_{..}$ は変数 x, y を固定しないので、全個数 N と一致する。

$$f_{i\cdot} = \sum_{j=1}^m f_{ij} \quad (i = 1, 2, \dots, n), \quad f_{\cdot j} = \sum_{i=1}^n f_{ij} \quad (j = 1, 2, \dots, m),$$

$$f_{..} = \sum_{i=1}^n \sum_{j=1}^m f_{ij} = \sum_{i=1}^n f_{i\cdot} = \sum_{j=1}^m f_{\cdot j} = N$$

周辺度数を用いて、変数 x, y が無関係な場合の度数（つまり周辺度数を固定した場合、周辺度数に比例するようにデータを配分したときの度数）としての期待度数 $e_{ij} = \frac{f_{i\cdot} \cdot f_{\cdot j}}{N}$ が定義できる。

1.4.4 2x2分割表の要約(1) 一致係数 [2]

特に、表1.2.8 の形で与えられる2値（ある/ない、True/False）の質的変数については、さらに多くの関連性（一致）の指標が提案されている[2]。

表1.2.8 2x2分割表 ([2]より引用)

表 1.2.8 2 × 2 分割表.

変数 x	変数 y		
	y_1	y_2	
x_1	a	b	n_1
x_2	c	d	n_2
	n_3	n_4	N

表1.2.8において、変数 x, y が2値 $x = \{x_1, x_2\}, y = \{y_1, y_2\}$ を取るとき、Jaccard係数 s_{xy}^J 、Russel-Rao係数 s_{xy}^R 、単純一致係数 s_{xy}^S はそれぞれ以下で定義される。

$$s_{xy}^J = \frac{a}{a + b + c + d} \quad (\text{Jaccard 係数})$$

$$s_{xy}^R = \frac{a}{a + b + c} \quad (\text{Russel-Rao 係数})$$

$$s_{xy}^S = \frac{a + d}{a + b + c + d} \quad (\text{単純一致係数})$$

一致の指標としては比率であるので0から1までの値を取る。係数が1に近いほど変数間の関連が強く、0に近いほど関連が弱い。ここで x_1, y_1 が正例（ある、True）、 x_2, y_2 が負例（ない、False）であるとすると、正例と負例に特に制限がなければ単純一致係数を用いる。一方、正例のみに興味があるときには Jaccard係数、Russel-Rao係数を用いることがある（負例に興味がない、意味が無いと考えている時）。

【計算例】 具体的な2x2分割表として表1.2.9について考える[2]。今回の観測データ ($N=200$) が、例えば変数 x, y として、変数 x : 毎食後必ず歯を磨く (x_1 =はい、 x_2 =いいえ)、変数 y : 歯周病の経験の有無 (y_1 =ない、 y_2 =ある)、観測度数は各々の回答件数であったとする。総数 $N=200$ のうち、歯をしっかり磨いている (x_1) かつ歯周病になったことが無い (y_1) ケースは10ある。逆にあまり歯を磨いていない (x_2) かつ歯周

病になった (y_2) ケースは100ある。周辺度数を見ると、歯をあまり磨かない回答の割合が、しっかり磨く回答の3倍にのぼっている。他方、歯周病になったという回答の割合が、歯周病になったことは無いという回答の2.3倍ある。

表1.2.9 2x2分割表の例 ([2]より引用)

表 1.2.9 分割表の例.

変数 x	変数 y		
	y_1	y_2	
x_1	10	40	50
x_2	50	100	150
	60	140	200

表1.2.9について、 x_1 かつ y_1 を正例と仮定し、観測度数から計算すると、Jaccard係数 $s_a^J = 1/20$ 、Russel-Rao係数 $s_a^R = 1/10$ 、単純一致係数 $s_{ad}^S = 11/20$ となる。

正例のみに注目した解釈として、Jaccard係数によると真面目に歯も磨くし（結果的に）歯周病にもなったことがない割合は5%しか存在しない。しかし、今回の観測データは総数200のうちの半数100が、歯も磨かないし（結果的に）歯周病にもなっている、いわば自業自得な層（負例）の存在を得ている。この「自業自得」の100ケースを除外して考えれば、Russel-Rao係数により正例が10%存在することになる。

一方、こういった負例も含めた総合的な解釈としては、単純一致係数により 55% は歯磨き習慣と歯周病との関連があるとも言える。つまり、どのような観点からデータ分析と研究を行いたいか、によって正例・負例の定義、採用する一致係数の扱いが異なってくる（もちろん、医学的な因果関係の解明は、データサイエンスとは別の領域になる）。

1.4.5 2x2分割表の要約(2) リスク（差・比）、オッズ（比） [2]

表1.2.8 の形で与えられる2値（ある/ない、True/False）の質的変数について、変数 x が要因（条件）であり、変数 y が結果であるという前提で、要因に占める結果ありの割合、また結果どうしの比について指標が提案されている[2]。

表1.2.8において、前者はリスク R として $a/(a+b), c/(c+d)$ で定義する。後者はオッズ O として $a/b, c/d$ で定義する。リスク（オッズ）は要因 x が 2 値 $x = \{x_1, x_2\}$ どちらかのケースであるから、リスク差（RD）・リスク比（RR）・オッズ比（OR）を計算することで有意な指標として利用できる。それぞれの指標は以下で定義される。

$$\begin{aligned} RD &= \frac{a}{a+b} - \frac{c}{c+d} \\ RR &= \frac{a}{a+b} / \frac{c}{c+d} \\ OR &= \frac{a}{b} / \frac{c}{d} \end{aligned}$$

【計算例】具体的な2x2分割表として、再び表1.2.9について考える[2]。変数 x ：毎食後必ず歯を磨く（ x_1 =はい、 x_2 =いいえ）が要因であり、変数 y ：歯周病の経験の有無（ y_1 =ない、 y_2 =ある）が結果であると仮定する（あくまで仮定である。医学的な因果関係は別の課題）。3つの指標を計算するとそれぞれ、 $RD = -2/15$ 、 $RR = 3/5$ 、 $OR = 1/2$ となる。

まず RD がマイナスの値を取っているし、RR が比3:5であること、ORについても比1:2であることから、結果への影響は要因2の方が高いと判断される。もちろん、どの指標が利用できるかは対象とするデータ分析研究の目的・設計による。

(再掲) 表1.2.9 2x2分割表の例 ([2]より引用)

表 1.2.9 分割表の例.

変数 x	変数 y		
	y_1	y_2	
x_1	10	40	50
x_2	50	100	150
	60	140	200

あらかじめ集団を条件で2つに分けておき、それらの集団を追跡してどのような結果になるか調査する手法を「コホート研究(cohort study)、前向き研究 (prospective study)」と言う。前向き研究の場合には最初に対象集団を固定しているから、リスク比(RR)、オッズ比(OR)とも有意である。

他方、ある結果となった集団とそうでない集団に対して、要因の有無を（逆問題として）調査する手法を「ケースコントロール研究 (case control study)、後向き研究 (retrospective study)」と言う。後向き研究の場合はオッズ比(OR)は利用可能であるが、対象集団を固定していないのでリスク比(RR)は利用できない（そうでない集団はいくらでも増やすことができる）。今回の具体例だと、「自業自得」な層は後でいくらでも集めてくるこ
レバーチャ

1.4.6 2x2分割表の要約(3) 混合行列、正解率、精度、再現率、特異度、F値 [2]

表1.2.8 の形で与えられる2値（ある/ない、True/False）の質的変数について、特に変数 x が真値、変数 y が予測値、 $\langle x_1, y_1 \rangle$ が正例、 $\langle x_2, y_2 \rangle$ が負例である状況を考える。このとき、2x2分割表は特に「混合行列」と呼ばれる[2]。

混合行列での各観測度数の呼び方について、aは真陽性(True Positive, TP)、bは偽陰性(False Negative, FN)、cは偽陽性(False Positive, FP)、dは真陰性(True Negative, TN) と呼ばれる。

(再掲) 表1.2.8 2x2分割表、表1.2.10 混合行列 ([2]より引用)

表 1.2.8 2×2 分割表.

変数 x	変数 y		
	y_1	y_2	
x_1	a	b	n_1
x_2	c	d	n_2
	n_3 n_4		N

表 1.2.10 混同行列.

真値	予測値	
	正例	負例
正例	真陽性 (TP) a	偽陰性 (FN) b
負例	偽陽性 (FP) c	真陰性 (TN) d

混合行列においてTP/FN/FP/TNが得られたとき、真値と予測値との間での要約指標として、**正解率(accuracy)**、**精度あるいは適合度(precision)**、**感度あるいは再現率 (recall)**、**特異度(specificity)**の4つの指標が、それぞれ以下で定義される。これらは機械学習系で得られた予測モデルの評価時において重要な指標となる。

$$\text{正解率} = \frac{a + d}{a + b + c + d} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

$$\text{精度} = \frac{a}{a + c} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{再現率} = \frac{a}{a + b} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{特異度} = \frac{d}{c + d} = \frac{\text{TN}}{\text{FP} + \text{TN}}$$

精度と再現率はトレードオフの関係にある。なぜなら、偽陽性FPと偽陰性FNがそれぞれの分母に入っていて、精度を上げようとするならFPを小さくする方策つまり厳しく正例判定を行う必要があるが、他方再現率を上げようとするならFNを小さくする方策つまり厳しく負例判定 (=正例判定を緩くする) を行う必要があるためである。そこで、これらのバランスをとって、両者の調和平均で定義されたのが**F値**である。

$$F = \frac{2}{\frac{1}{\text{精度}} + \frac{1}{\text{再現率}}} = \frac{2 \times \text{精度} \times \text{再現率}}{\text{精度} + \text{再現率}} = \frac{2\text{TP}}{2\text{TP} + \text{FN} + \text{FP}}$$

2. 基本的なデータ可視化手法（質的データを中心として） [2]

2.1 データ可視化の目的

可視化の主な目的は「比較する」「変化を見る」「構成を見る」「分布を見る」「相関を知る」の5つに分けられる。

1. データの可視化の目的

表2.1は、可視化の目的とデータの尺度に応じて基本的なグラフを分類したものであり、それぞれのグラフ単独での目的に基づいて分類していることが明記されている（同じグラフを複数並べての比較は含まれていない）[2]。

(再掲) 表2.1 データの量質 × 目的ごとの可視化法 ([2]より引用)

	比較	構成	分布	変化
量的データ	レーダーチャート		箱ひげ図 ヒストグラム バイオリンプロット 散布図 散布図行列	折れ線グラフ
質的データ	棒グラフ	円グラフ 帯グラフ	モザイクプロット ヒートマップ	
	パレート図 積み上げ棒グラフ 層別帶グラフ			

統計グラフを利用する際には、データの尺度や変数の数に留意する必要がある。表3.2では、これらに応じた可視化法が区分されている。量的データにはヒストグラムや箱ひげ図、質的データには円グラフや棒グラフがよく用いられる。また、2変数以上の質的データには層状帶グラフ、モザイクプロットやヒートマップがよく用いられる [2]。

(再掲) 表2.2 データの量質 × 変数の数ごとの可視化法 ([2]より引用)

	1変数データ	2変数データ	多変数データ
量的データ	箱ひげ図 ヒストグラム バイオリンプロット 折れ線グラフ	散布図	散布図行列 レーダーチャート
質的データ	棒グラフ パレート図 円グラフ 帯グラフ	積み上げ棒グラフ 層別帶グラフ モザイクプロット ヒートマップ	モザイクプロット行列

以下では、質的データを主な対象とし、可視化・観察・分析する手法のいくつかについて説明していく。

2.2 1変数 量的x質的データの可視化

2.2.1 グラフによる可視化(1) 棒グラフ

2. グラフによる可視化：A. 比較（1）

データを表やグラフにまとめると大小関係がわかります。
棒グラフ（右）を作成すると比較しやすくなります。

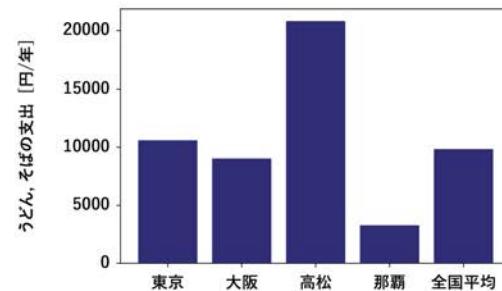
表：1世帯のうどん、そばの平均支出金額
(購入+外食)

都市名	年間支出金額
東京都（区部）	10,594
大阪市	9,026
高松市	20,807
那覇市	3,271
全国平均	9,838

総務省統計局「2019(令和元)年家計調査」
(調査年月：令和元年)

棒グラフ：1世帯のうどん、そばの平均支出金額
(購入+外食)

可視化
→



東京都、大阪市は全国平均に近く、
高松市は全国平均の2倍以上うどん、そばを
消費していることがわかります。

参考情報：[うどんに関する統計情報](#)

東京大学 数理・情報教育研究センター 小林亮太 2021 CC BY-NC-SA

5

(http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf#page=5).

- 香川県 | うどん県統計情報コーナー (<https://www.pref.kagawa.lg.jp/tokei/sogo/udonken/kfvn.html>)

2.2.2 Pythonによる棒グラフ生成の実装例

- オリジナルデータとプログラム例（公開教材[1]） (https://github.com/MDASH-shinshu/MDASH-TDS/tree/main/3/resources/additional_material/p5_map/3_bar)

In []:

```
# CSVデータを カレントディレクトリ直下のフォルダ(一時作業領域)へダウンロードする.  
!wget -nc https://raw.githubusercontent.com/MDASH-shinshu/MDASH-T-DS/main/3/resources/udon.csv  
  
# wgetしなくても,Google colab.の左メニュー [ファイル] アイコンをクリックして,ブラウザへファイルをドラッグ&ドロップし  
# ファイル (udon.csv)がダウンロード・配置できることを確認する  
!ls -al ./
```

File ‘udon.csv’ already there; not retrieving.

```
total 548  
drwxr-xr-x 1 root root 4096 Apr 9 01:22 .  
drwxr-xr-x 1 root root 4096 Apr 8 23:41 ..  
-rw-r--r-- 1 root root 16749 Apr 9 01:23 band.png  
-rw-r--r-- 1 root root 153 Apr 9 01:03 capacity_ratio.csv  
-rw-r--r-- 1 root root 33860 Apr 9 01:07 circle.png  
drwxr-xr-x 4 root root 4096 Apr 6 13:38 .config  
-rw-r--r-- 1 root root 460676 Apr 9 00:26 house_prices_train.csv  
drwxr-xr-x 2 root root 4096 Apr 9 01:02 .ipynb_checkpoints  
-rw-r--r-- 1 root root 7943 Apr 9 00:26 JJIHw.csv  
-rw-r--r-- 1 root root 50 Apr 9 00:07 pareto_data.csv  
drwxr-xr-x 1 root root 4096 Apr 6 13:39 sample_data  
-rw-r--r-- 1 root root 65 Apr 9 00:26 udon.csv
```

In []:

```
# オリジナルのCSVファイルをpandasで読み込んでデータフレームdfに格納  
import pandas as pd  
import matplotlib.pyplot as plt  
df = pd.read_csv('udon.csv', header=None)  
df.columns = ['city', 'volume']  
df.head()
```

Out[86]:

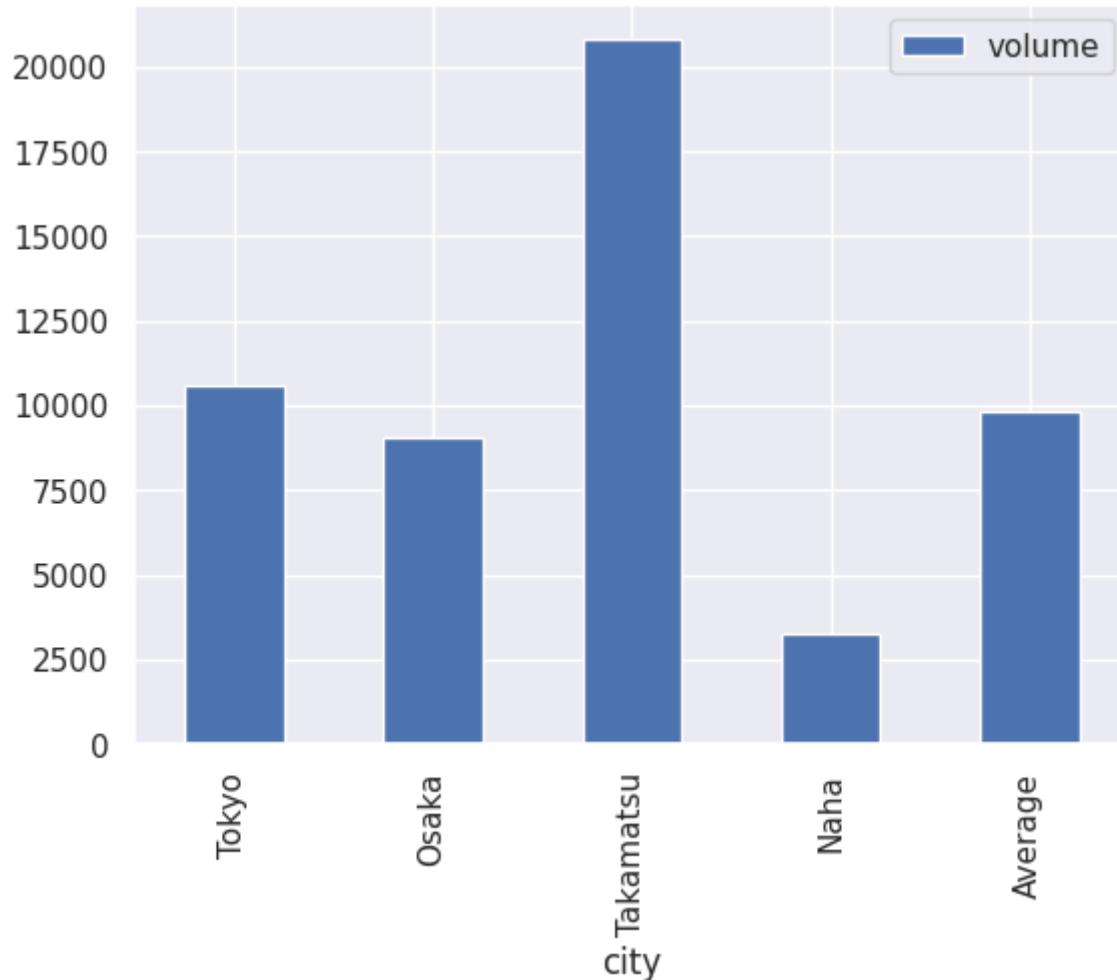
	city	volume
0	Tokyo	10594
1	Osaka	9026
2	Takamatsu	20807
3	Naha	3271
4	Average	9838

In []:

```
# matplotlib経由でデータフレームの棒グラフを表示する  
df.plot.bar('city', 'volume')
```

Out[87]:

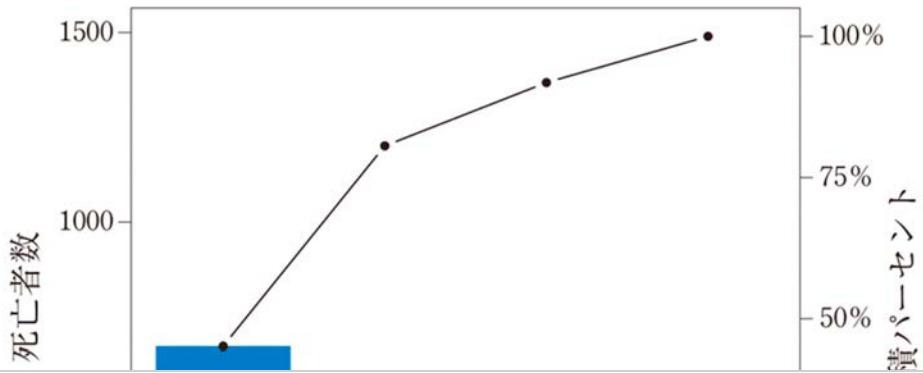
<Axes: xlabel='city'>



2.2.3 グラフによる可視化(2) パレート図

パレート図は、度数を棒グラフで表示し、累積相対度数を折れ線グラフで表示するグラフで、品質管理の7つ道具の1つである。左側の縦軸に度数を、右側の縦軸に累積相対度数を表示し、比較と構成の把握を行える。項目を多い順に表示し、全体に占める累積比率をパーセントで表す。

図1.3.1は、有名なタイタニック号沈没事故の死亡者数に関するデータをパレート図で可視化したものである。乗組員、3等、2等、1等客室の順で死亡者数が多い。乗組員と3等客室の船客で死者の75%以上を占めている。



2.2.4 Pythonによるパレート図生成の実装例

- オリジナルデータとプログラム例（【python】matplotlibでパレート図を作成する方法 (<https://mori-memo.hateblo.jp/entry/2022/06/05/224841>)）

In []:

```
# CSVデータを カレントディレクトリ直下のフォルダ(一時作業領域)へダウンロードする。
!wget -nc https://raw.githubusercontent.com/MDASH-shinshu/MDASH-T-DS/main/3/resources/pareto_
# wgetなくても,Google colab.の左メニュー [ファイル] アイコンをクリックして,ブラウザへファイルをドラッグ&ドロップし
# ファイル (pareto_data.csv)がダウンロード・配置できたことを確認する
!ls -al ./
```

File ‘pareto_data.csv’ already there; not retrieving.

```
total 548
drwxr-xr-x 1 root root 4096 Apr 9 01:22 .
drwxr-xr-x 1 root root 4096 Apr 8 23:41 ..
-rw-r--r-- 1 root root 16749 Apr 9 01:23 band.png
-rw-r--r-- 1 root root 153 Apr 9 01:03 capacity_ratio.csv
-rw-r--r-- 1 root root 33860 Apr 9 01:07 circle.png
drwxr-xr-x 4 root root 4096 Apr 6 13:38 .config
-rw-r--r-- 1 root root 460676 Apr 9 00:26 house_prices_train.csv
drwxr-xr-x 2 root root 4096 Apr 9 01:02 .ipynb_checkpoints
-rw-r--r-- 1 root root 7943 Apr 9 00:26 JJlHw.csv
-rw-r--r-- 1 root root 50 Apr 9 00:07 pareto_data.csv
drwxr-xr-x 1 root root 4096 Apr 6 13:39 sample_data
-rw-r--r-- 1 root root 65 Apr 9 00:26 udon.csv
```

In []:

```
# オリジナルのCSVファイルをpandasで読み込んでデータフレームdfに格納
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
df = pd.read_csv('pareto_data.csv', header=None)
df.columns = ['label', 'volume']
df.head()
```

Out[89]:

	label	volume
0	APPLE	20
1	BANANA	40
2	CACAO	80
3	DURIAN	5
4	EGG PLANT	15

In []:

```
# パレート図を生成する関数の定義 https://mori-memo.hateblo.jp/entry/2022/06/05/224841
def pareto(df, label_name, cnt_name):
    """ plot pareto chart
    input
        df: data for plotting (DataFrame)
        label_name: row name for label (str)
        cnt_name: row name for count (str)
    """
    label = df[label_name].tolist()
    cnt = df[cnt_name].tolist()
    data = pd.DataFrame(label, columns=['label'])
    data['volume'] = cnt
    data = data.sort_values(by='volume', ascending=False)
    sum_data = data['volume'].sum()
    data['accum'] = np.cumsum(data['volume'])
    data['accum_ratio'] = data['accum'] / sum_data * 100

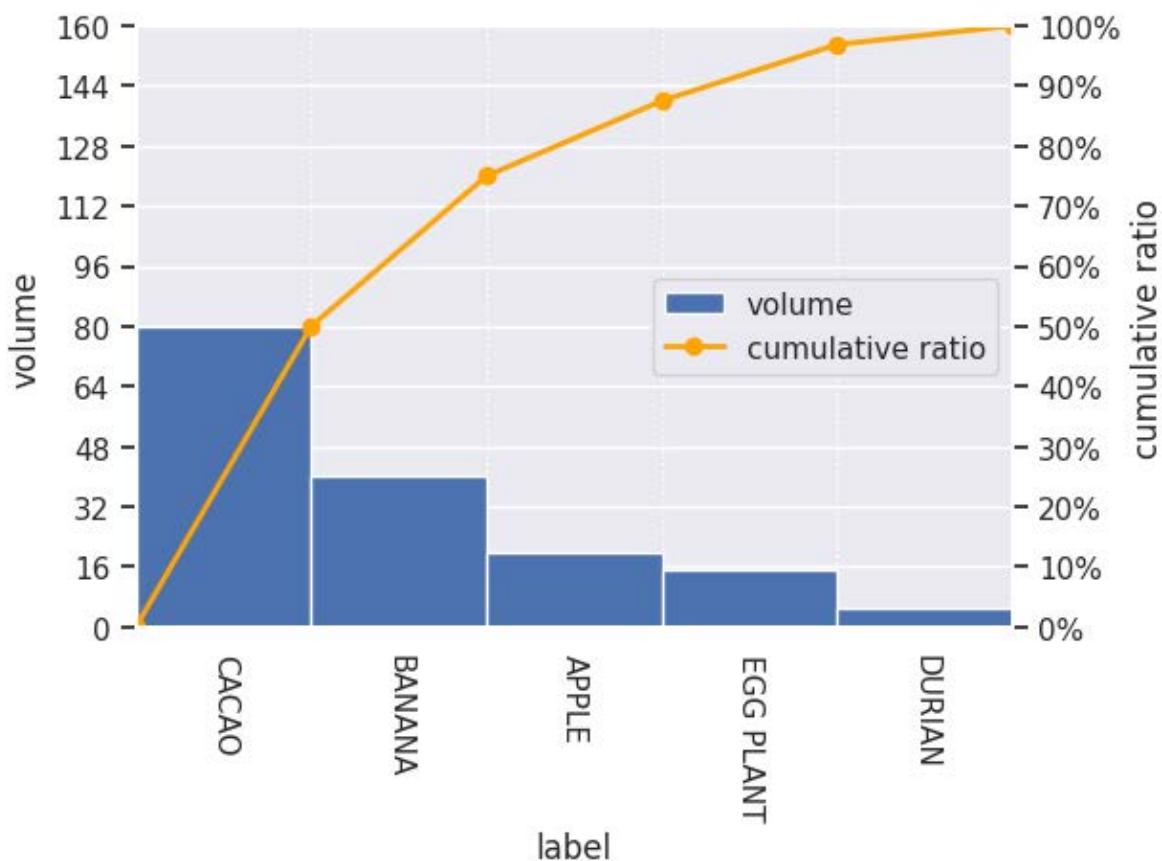
    labels = len(data)

    percent_labels = [str(i) + "%" for i in np.arange(0, 100+1, 10)]
    accum_data = [0] + data['accum_ratio'].tolist()
    fig, ax = plt.subplots()
    bar = ax.bar(range(labels), data['volume'], align="edge", width=1, label=cnt_name)
    ax.set_ylim([0, sum_data])
    ax.set_xlim([0, labels])
    ax.set_xticks([0.5 + i for i in range(labels)])
    ax.set_xticks([1 + i for i in range(labels)], minor=True) #副目盛り
    ax.set_xticklabels(data['label'].tolist(), rotation=-90)
    ax.set_yticks(np.arange(0, sum_data+1, sum_data / 10))
    ax.grid(axis='x')
    ax.grid(axis='x', ls=':', which='minor') #副目盛り
    ax.set_xlabel(label_name)
    ax.set_ylabel(cnt_name)

    twin_ax=ax.twinx()
    twin_ax.plot(range(labels + 1), accum_data, marker='o', color='orange', label='cumulative ratio', linewidth=2)
    twin_ax.set_ylabel('cumulative ratio')
    twin_ax.set_ylim([0, 100])
    twin_ax.set_yticks(np.arange(0, 100+1, 10))
    twin_ax.set_yticklabels(percent_labels)
    twin_ax.grid(False)
    fig.legend(loc="center right", bbox_to_anchor=(1,0.5), bbox_transform=ax.transAxes)
    plt.tight_layout()
    plt.show()
```

In []:

```
#パレート図の表示  
pareto(df, 'label', 'volume')
```



2.3 2変数・多変数 量的x質的データの可視化

2.3.1 グラフによる可視化(3) 円グラフ、帯グラフ

2. グラフによる可視化：C. 構成比（1）

円グラフ（上）や帯グラフ（下）を作成すると、
全体に占める割合（構成比）を比較しやすくなります。

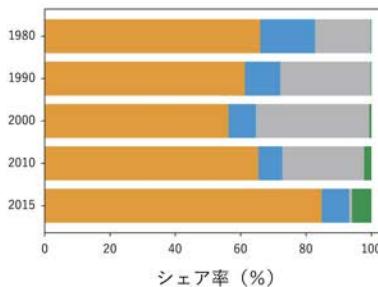
□ ダニ一

2. グラフによる可視化：C. 構成比（2）

比較対象が多い場合には、
帯グラフ（左）や積み上げ縦棒グラフ（右）が便利です。

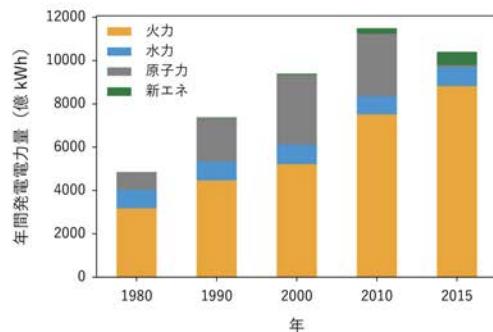
例：1980, 1990, 2000, 2010, 2015 年の日本の発電量の比較

帯グラフ



出典：資源エネルギー庁/エネルギー白書2019 [リンク](#)

積み上げ縦棒グラフ



どちらのグラフからも、火力発電の割合は2000年まで減少、それから増加していること、
新エネの割合は2010年以降急増していることがわかります。
積み上げ縦棒グラフ（右）から、日本の発電量は2010年がピークであったことがわかります。

東京大学 数理・情報教育研究センター 小林亮太 2021 CC BY-NC-SA

13

(http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf#page=13)

2.3.2 Pythonによる円グラフ、帯グラフ生成の実装例

- オリジナルデータとプログラム例（公開教材[1]）(https://github.com/MDASH-shinshu/MDASH-TDS/tree/main/3/resources/additional_material/p12_band/Ex1)

In []:

```
# CSVデータを カレントディレクトリ直下のフォルダ(一時作業領域)へダウンロードする.  
!wget -nc https://raw.githubusercontent.com/MDASH-shinshu/MDASH-T-DS/main/3/resources/capacity_ratio.csv  
  
# wgetしなくても,Google colab.の左メニュー [ファイル] アイコンをクリックして,ブラウザへファイルをドラッグ&ドロップしてもOK  
# ファイル (capacity_ratio.csv)がダウンロード・配置できたことを確認する  
!ls -al ./
```

File ‘capacity_ratio.csv’ already there; not retrieving.

```
total 548  
drwxr-xr-x 1 root root 4096 Apr 9 01:22 .  
drwxr-xr-x 1 root root 4096 Apr 8 23:41 ..  
-rw-r--r-- 1 root root 16749 Apr 9 01:23 band.png  
-rw-r--r-- 1 root root 153 Apr 9 01:03 capacity_ratio.csv  
-rw-r--r-- 1 root root 33860 Apr 9 01:07 circle.png  
drwxr-xr-x 4 root root 4096 Apr 6 13:38 .config  
-rw-r--r-- 1 root root 460676 Apr 9 00:26 house_prices_train.csv  
drwxr-xr-x 2 root root 4096 Apr 9 01:02 .ipynb_checkpoints  
-rw-r--r-- 1 root root 7943 Apr 9 00:26 JJIHW.csv  
-rw-r--r-- 1 root root 50 Apr 9 00:07 pareto_data.csv  
drwxr-xr-x 1 root root 4096 Apr 6 13:39 sample_data  
-rw-r--r-- 1 root root 65 Apr 9 00:26 udon.csv
```

In []:

```
# オリジナルのCSVファイルをpandasで読み込んでデータフレームdfに格納  
import pandas as pd  
import matplotlib.pyplot as plt  
data = pd.read_csv('capacity_ratio.csv', header=0)  
data.columns = ['category', '2010', '2015']  
data.head()
```

Out[93]:

	category	2010	2015
0	Thermal	0.654291	0.848091
1	Hydro	0.072904	0.083733
2	Nuclear	0.250754	0.009071
3	Renewable	0.022051	0.059105

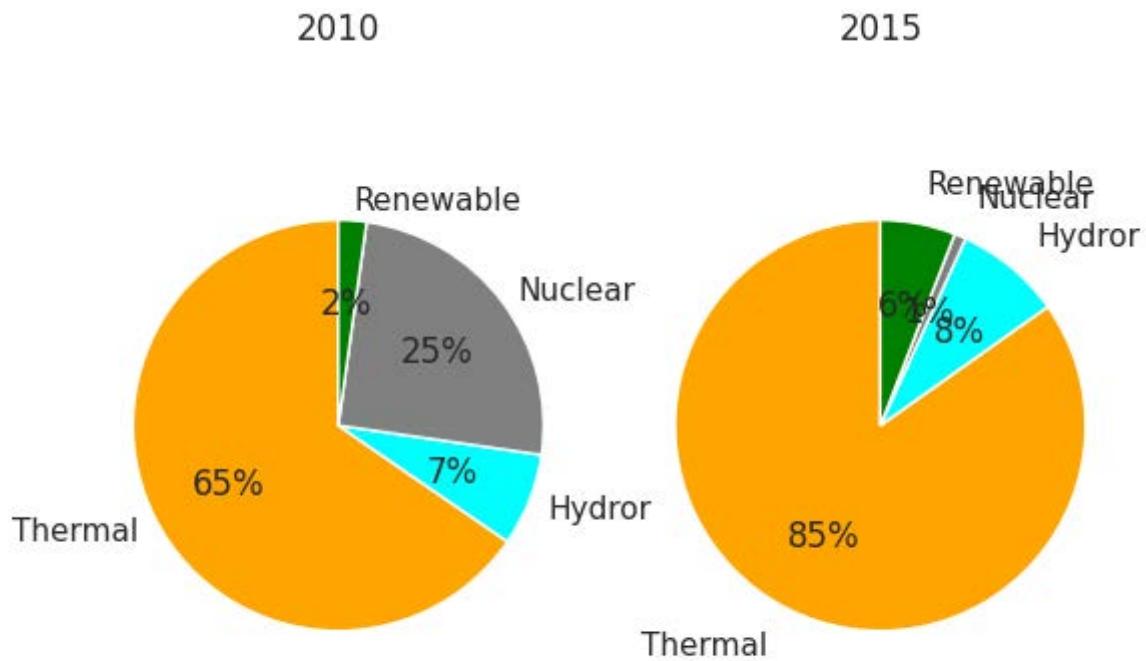
In []:

```
# Figure, labelを設定
fig = plt.figure()
labels = ["Thermal", "Hydror", "Nuclear", "Renewable"]
data.index = labels
colors = ["orange", "cyan", "gray", "green"]

# 2010年の円グラフ
fig.add_subplot(1, 2, 1)
plt.title("2010")
plt.pie(data['2010'], labels = data.index, startangle=90, autopct = "%1.0f%%", colors = colors, labeldistance = 1.1, wedgeprops = {"edgecolor": "black"}, textprops = {"color": "white"})
plt.axis('equal')

# 2015年の円グラフ
fig.add_subplot(1, 2, 2)
plt.title("2015")
plt.pie(data['2015'], labels = data.index, startangle=90, autopct = "%1.0f%%", colors = colors, labeldistance = 1.1, wedgeprops = {"edgecolor": "black"}, textprops = {"color": "white"})
plt.axis('equal')

# グラフを保存
plt.savefig('circle.png')
```



In []:

```
# 帯グラフ用にデータを加工
years = ['2015', '2010']
data = 100* data[years]
data
```

Out[95]:

	2015	2010
Thermal	84.809115	65.429070
Hydro	8.373285	7.290421
Nuclear	0.907058	25.075440
Renewable	5.910542	2.205068

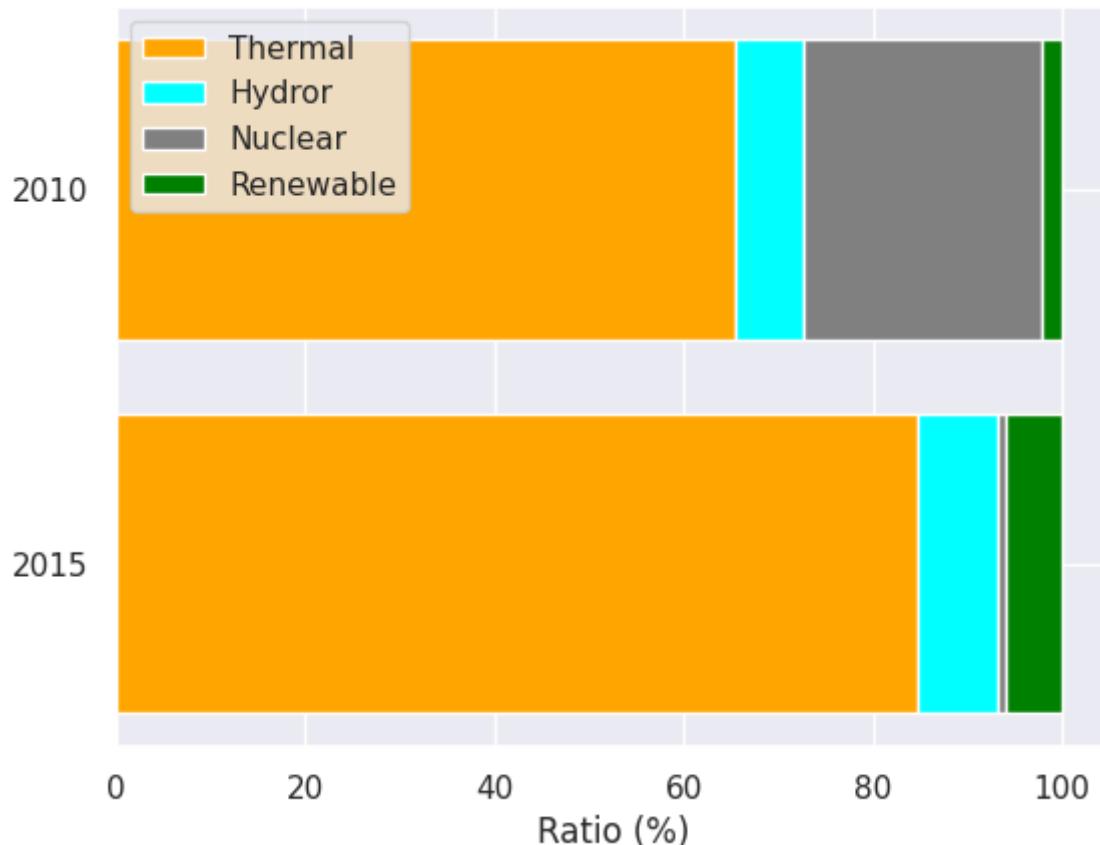
In []:

```
# Figure, ax軸, label, 色を設定
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
labels=["Thermal", "Hydror", "Nuclear", "Renewable"]
data.index = labels
colors = ["orange", "cyan", "gray", "green"]
color_list = dict(zip(labels, colors))

# 縦軸設定、グラフを出力
label_y = ["2015", "2010"]
for i in data.index:
    ax.barh(label_y, data.loc[i], left=data.loc[:i].sum()-data.loc[i], color=color_list[i])

# 凡例を出力
ax.legend(data.index)
plt.xlabel('Ratio (%)')

# グラフを保存
plt.savefig('band.png')
```



ヒートマップ、モザイクプロット、散布図については、chap.2を参考にされたい。

2.3.2 グラフによる可視化(4) 層状棒グラフ、beeswarm（蜜蜂）図

- 引用：【Python入門講座】タイタニック号のデータ分析 (<https://takun-physics.net/12932/>)
- データセット Kaggle | Titanic - Machine Learning from Disaster (<https://www.kaggle.com/c/titanic/data>)

In []:

```
# 必要なライブラリをインポート
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
```

In []:

```
# Kaggle タイタニックデータをダウンロードしていく
import seaborn as sns
sns.get_dataset_names #データセットの確認
df_titanic = sns.load_dataset("titanic") #タイタニックのデータ
df_titanic
```

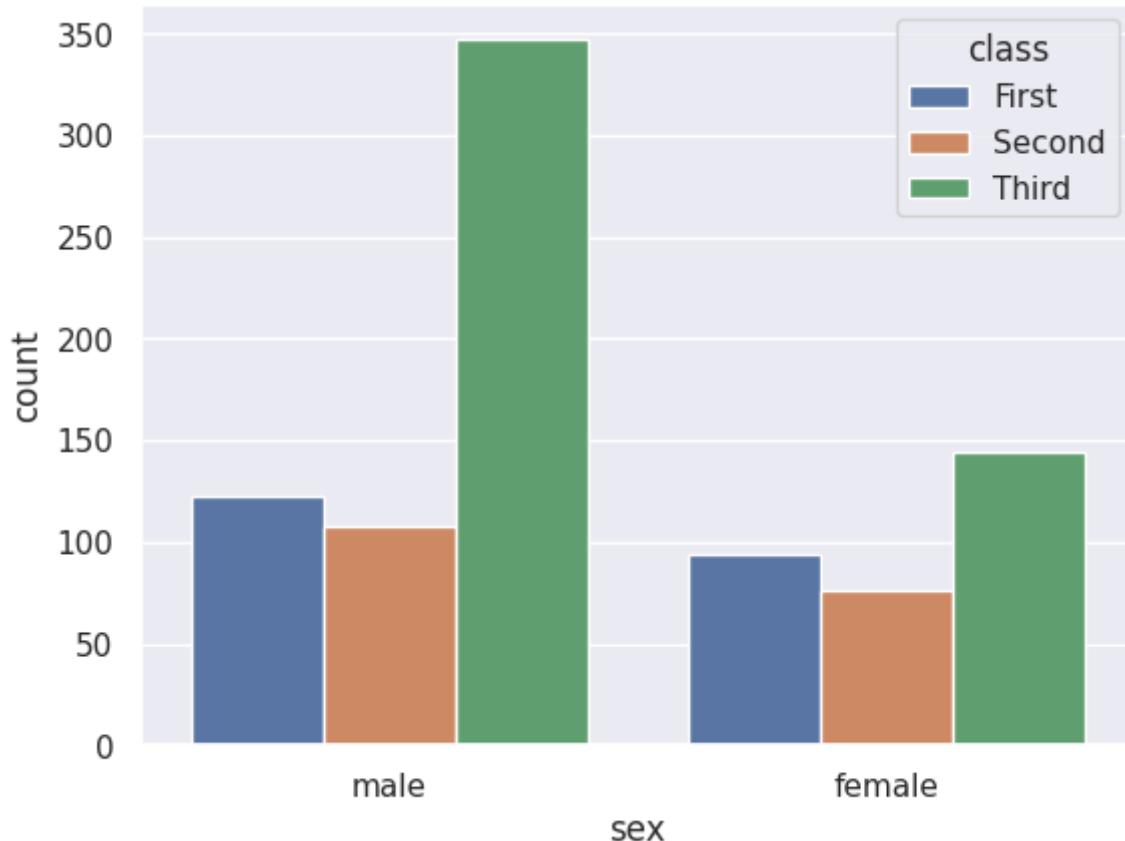
In []:

```
# 性別の中で客室等級(1,2,3)による分類
sns.countplot(x='sex', data=df_titanic, hue='class')

# 男性の方が乗客数が多かった(特に3等客室)ということがわかる
```

Out[120]:

```
<Axes: xlabel='sex', ylabel='count'>
```



In []:

```
# 列survivedの値がゼロは死者なので、客室等級(pclass)別の割合を見てみる  
sns.distplot(df_titanic[df_titanic['survived'] == 0]['pclass'], norm_hist=True, bins=3)  
plt.legend(['Not survived'])
```

死者数は客室等級3の人がかなり多い

<ipython-input-124-ef4e906f6bb9>:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).

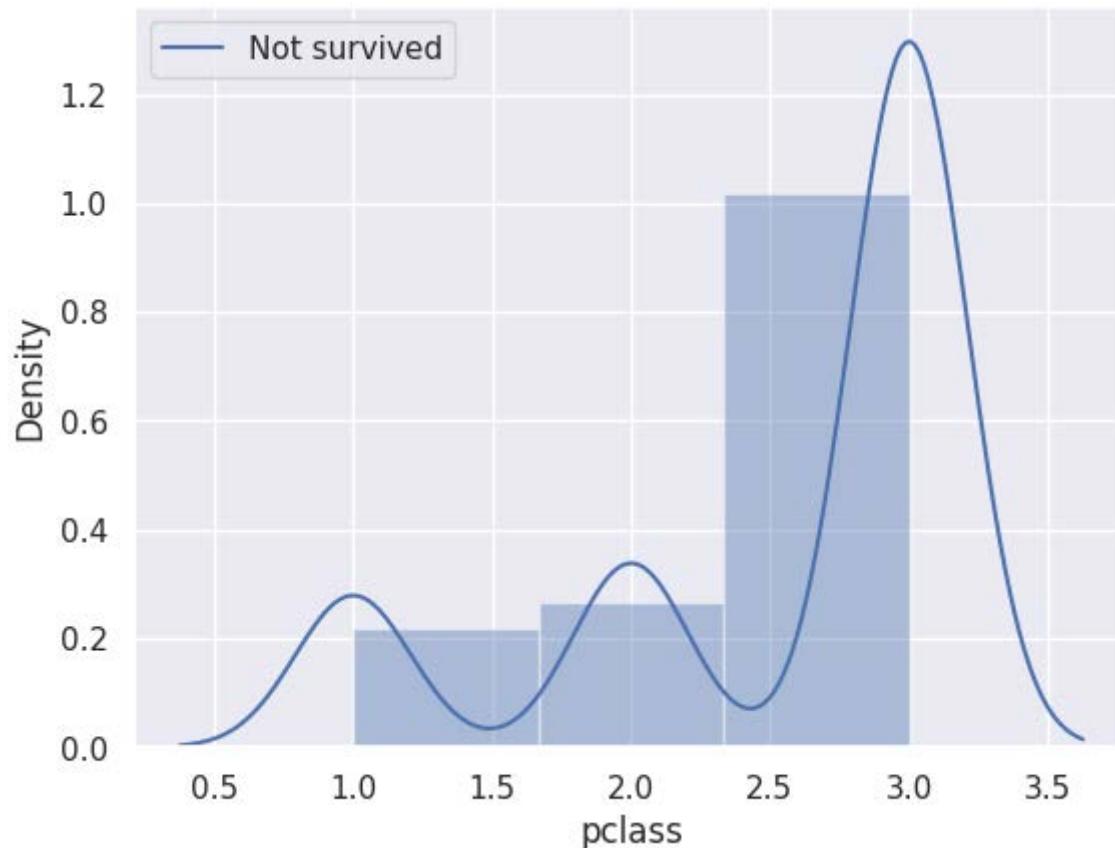
For a guide to updating your code to use the new functions, please see

<https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751> (<http://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751>)

```
sns.distplot(df_titanic[df_titanic['survived'] == 0]['pclass'], norm_hist=True, bins=3)
```

Out[124]:

<matplotlib.legend.Legend at 0x7eff985ec250>



In []:

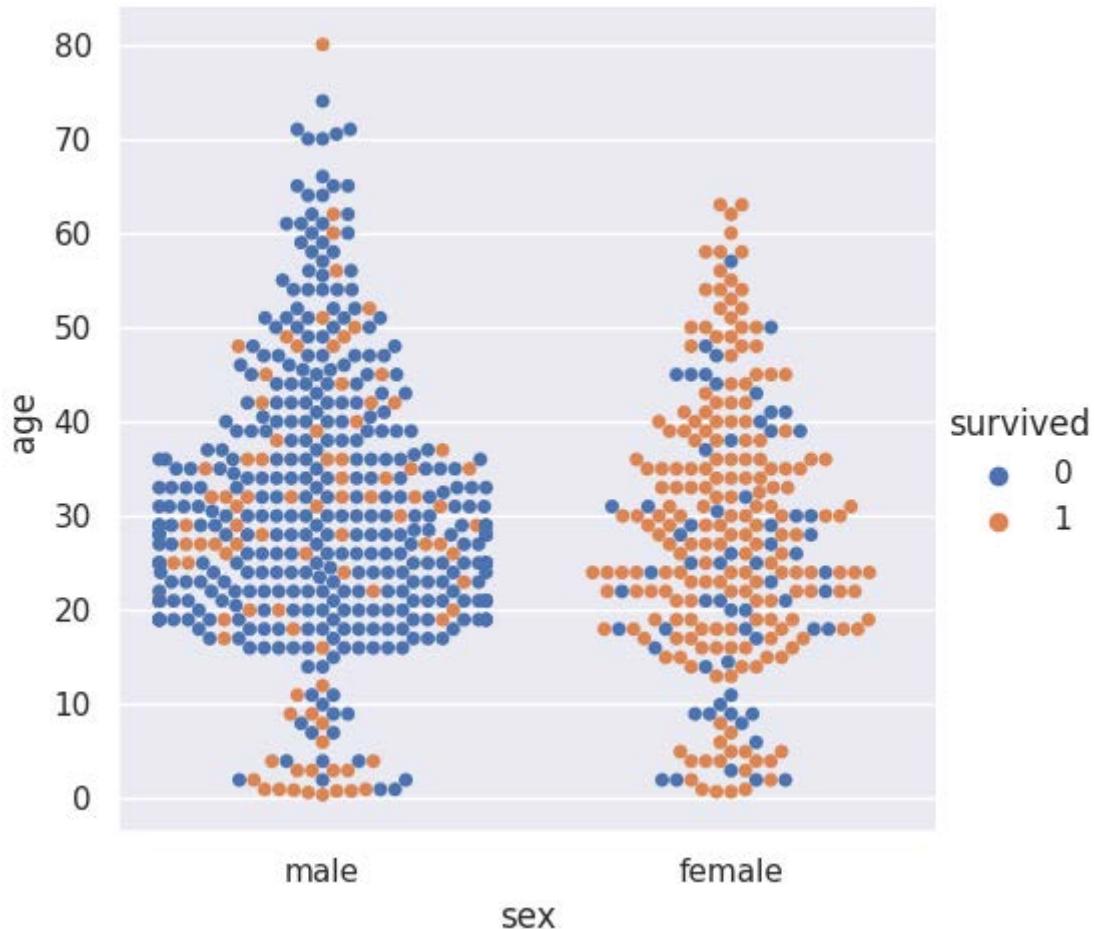
```
# N=891の各レコードについて、beeswarm(蜜蜂)図によって性別:年齢で、各データポイントが重ならないようにplotす  
# 各年齢(層)での横の広がりが度数のイメージとなる。  
sns.catplot(x='sex', y='age', data=df_titanic, kind='swarm', hue='survived')
```

```
# 0が死亡者、1が生存者。男女で比較すると女性の方が生存者が多く、男性でも10代の人は比較的生存者が多い  
# 映画でも描かれているが、女性と子ども優先で、救助艇によって離船していたという話がある(議論、諸説あり)
```

```
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 8.  
8% of the points cannot be placed; you may want to decrease the size of the markers or  
use stripplot.  
warnings.warn(msg, UserWarning)
```

Out[128]:

```
<seaborn.axisgrid.FacetGrid at 0x7effd1795cd0>
```



In []:

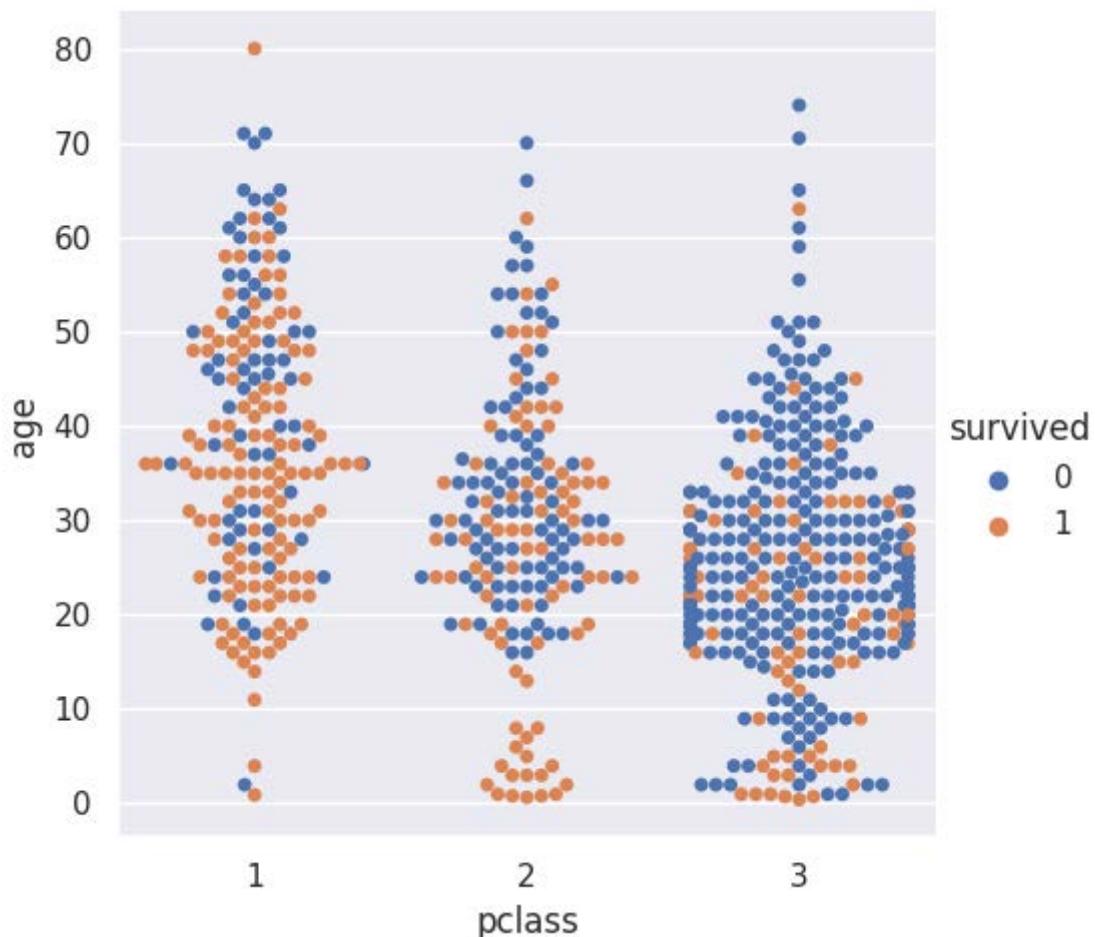
```
# 最後にチケットクラスと年齢で死者/生存者を確認してみる  
sns.catplot(x='pclass', y='age', data=df_titanic, kind='swarm', hue='survived')
```

等級が上がるほど生存率は上がり、一等客優先で離船させていたことがわかる

```
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 7.  
9% of the points cannot be placed; you may want to decrease the size of the markers or  
use stripplot.  
warnings.warn(msg, UserWarning)  
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 2  
1.4% of the points cannot be placed; you may want to decrease the size of the markers  
or use stripplot.  
warnings.warn(msg, UserWarning)  
/usr/local/lib/python3.9/dist-packages/seaborn/categorical.py:3544: UserWarning: 1  
4.1% of the points cannot be placed; you may want to decrease the size of the markers  
or use stripplot.  
warnings.warn(msg, UserWarning)
```

Out[129]:

```
<seaborn.axisgrid.FacetGrid at 0x7effd1645190>
```



3. ビッグデータの可視化、関係性の可視化 [1]

3.1 ビッグデータの可視化

5. ビッグデータの可視化（1）

ビッグデータとは、巨大で複雑なデータのことです。
ビッグデータが現れる状況として以下が考えられます。

- A. 長い：長期間あるいは高頻度のデータ
- B. 種類が多い
- C. 新しいタイプ：関係性、地図データなど

例：世界の気温

B. 種類が多い

A. 長い

	東京	パリ	アブダビ	...
2020年12月	7.7	6.4	21.8	...
2020年11月	14.0	9.4	26.1	...
...

気象庁「世界の天候データツール（ClimatView 月統計値）」[リンク](#)

東京大学 数理・情報教育研究センター 小林亮太 2021 CC BY-NC-SA

25

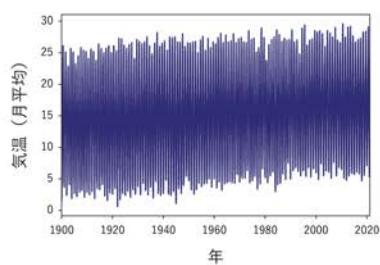
(http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf#page=25)

- 気象庁「世界の天候データツール（ClimatView 月統計値）」
(<https://www.data.jma.go.jp/gmd/cpd/monitor/climatview/frame.php>)

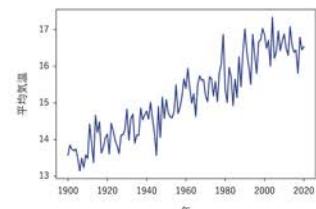
5. ビッグデータの可視化（2）

- A. 長いデータは平均を計算すると可視化しやすくなります。

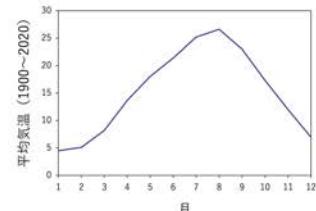
例：東京の気温



① 年ごと（1900年, 1901年…）の平均



② 月ごと（1月, 2月…）の平均



左は、東京の1900～2020年の月平均気温の折れ線グラフです。このグラフは複雑に変動しているように見えます。
右のように、年ごと（上）、月ごと（下）の平均をとったグラフを作成することで、長期的な振る舞いと年単位での振る舞いを観察できます。

東京大学 数理・情報教育研究センター 小林亮太 2021 CC BY-NC-SA

26

(http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf#page=26)

- ・ 気象庁 | 観測開始からの毎年の主な気象データ（東京都 東京）
(https://www.data.jma.go.jp/obd/stats/etrn/view/annually_s.php?prec_no=44&block_no=47662&year=&month=&day=&view=p1)
- ・ 気象庁 | 観測開始からの毎月の最高気温の平均値（東京都 東京）
(https://www.data.jma.go.jp/obd/stats/etrn/view/monthly_s3.php?prec_no=44&block_no=47662&year=&month=&day=&view=a2)

5. ビッグデータの可視化（3）

B. 一般に、種類が多い（4種類以上）ビッグデータの可視化は難しい問題です。この場合、次元削減技術（主成分分析：1.4節）を用いるとよいです。

C. 新しいタイプのビッグデータの可視化の方法

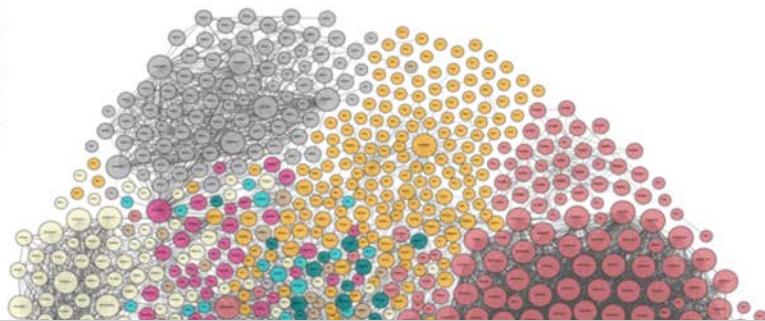
以下で3つの例を紹介します：

- ① グラフデータ → 関係性の可視化
- ② 地図データ → リアルタイム可視化
- ③ 映像データ → 軌跡の可視化

ビッグデータ可視化の例（Linked Open Data）

アノテーション

さまざまなLinked Open Data



3.2 関係性の可視化

6. 関係性の可視化（1）

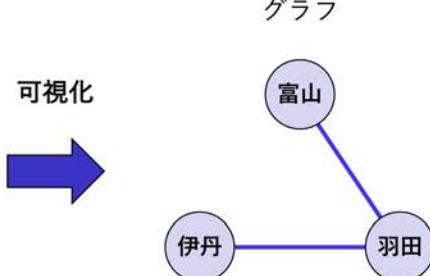
人や物の間につながりがあるかどうかを表現するため、グラフ（ネットワーク）を用います。

例1：航空機の国内線ネットワーク

羽田（東京） ⇌ 富山： 直行便あり

羽田 ⇌ 伊丹（大阪）： 直行便あり

伊丹 ⇌ 富山： 直行便なし



上の例では対象とする物は空港です。2つの空港を結ぶ直行便がある場合、2つの空港はつながっているとします。

直行便についての情報は、右図のように、空港を丸、直行便を線で表現した“グラフ”で示されます。このように、丸（頂点：空港）が線（枝：つながりを示す）で結ばれた图形を“グラフ”と呼びます。

参考文献

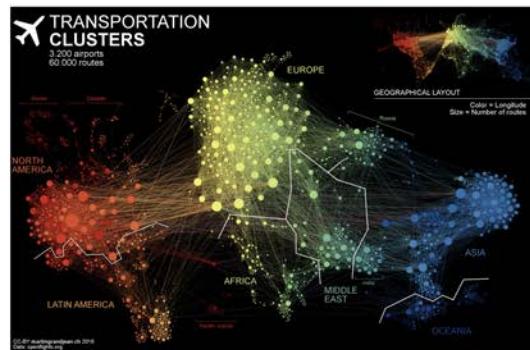
- 複雑ネットワーク (Wikipedia) [リンク](#)

- ANA 空港路線マップ (<https://www.anawings.co.jp/networkmap/>)

6. 関係性の可視化（2）

人や物の間につながりがあるかどうかを表現するため、グラフ（ネットワーク）を用います。

例2：航空機の国際線ネットワーク



出典：Martin Grandjean氏のホームページ [リンク](#)

世界の3,200空港の路線図のグラフ。

丸の大きさは空港からの路線数、丸の色は経度を表します。路線数の多い空港はハブ空港と呼ばれます。

このデータを活用すると、人や物の流れをシミュレーションできたり、効率的な輸送計画を立てたりできます。

東京大学 数理・情報教育研究センター 小林亮太 2021 CC BY-NC-SA

29

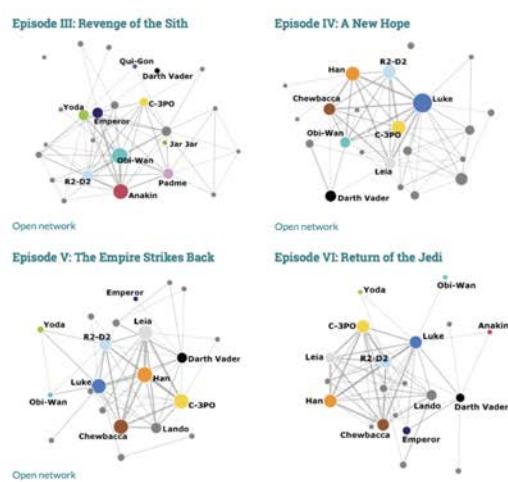
(http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf#page=29)

- [Connected World: Untangling the Air Traffic Network by Martin Grandjean](https://www.martingrandjean.ch/connected-world-air-traffic-network/)
(<https://www.martingrandjean.ch/connected-world-air-traffic-network/>)

6. 関係性の可視化（3）

人や物の間につながりがあるかどうかを表現するため、グラフ（ネットワーク）を用います。

例3：スター・ウォーズ（映画）の人間関係



出典：Evelina Gabašová 氏のホームページ [リンク](#)

スター・ウォーズの登場人物（ロボット、宇宙人も含める）のグラフ。

映画の同じシーンに現れた登場人物たちを「つながっている」と定義して、グラフを作成した。左のグラフから、映画の重要人物を可視化できることがわかります。

上段はスター・ウォーズ III (左)、IV (右)、下段はスター・ウォーズ V (左)、VI (右) の結果を表す。グラフデータは以下のリンクから入手できる。

[データへのリンク](#)

東京大学 数理・情報教育研究センター 小林亮太 2021 CC BY-NC-SA

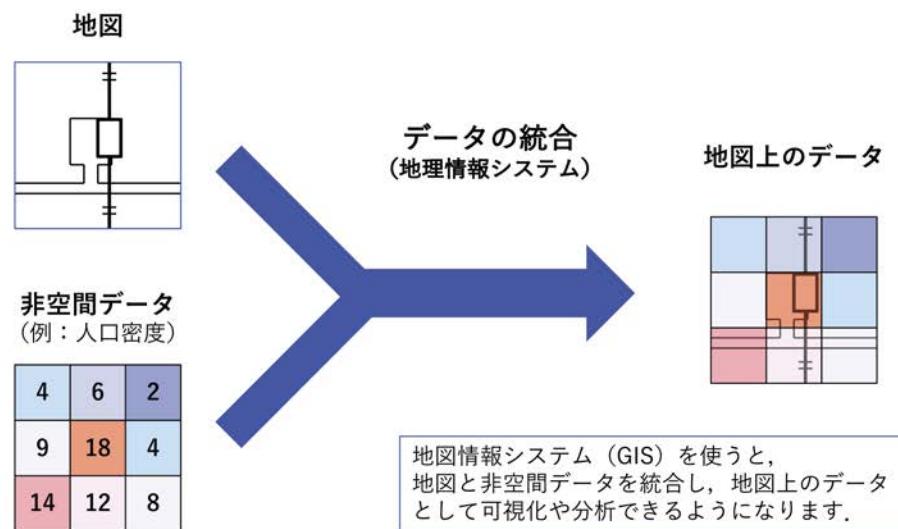
30

(http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf#page=30)

- Evelina Gabasova | StarWars social networks (<http://evelinag.com/blog/2016/01-25-social-network-force-awakens/>)

3.3 地図データ、GIS情報との融合

6. 地図データの可視化（1）



参考文献

- GIS基礎教材：[リンク](#)
- GISソフト (QGIS)：[リンク](#)

6. 地図データの可視化（2）

さらに、直近1時間のデータを非空間データとして用いることにより、地図上の可視化をリアルタイムで行うこともできます。

また、1時間毎の混雑度合い（モバイル空間統計、Yahoo地図）、雨雲が動く様子（Yahoo地図）、Covid-19感染者数の日毎の変化のリアルタイム可視化を行うサービスもあります。

詳しくは以下の例を参照。

例

- ・モバイル空間統計（ドコモ）：[リンク](#)
- ・Yahoo地図（Yahoo）：[リンク](#)
- ・CoVid-19 感染者数（インフォマティクス）：[リンク](#)

実データ

- ・関東圏の人の動きのデータ：[リンク](#)

東京大学 数理・情報教育研究センター 小林亮太 2021 CC BY-NC-SA

34

(http://www.mi.u-tokyo.ac.jp/pdf/1-5_data_visualization.pdf#page=34)

- ・モバイル空間統計（ドコモ）(<https://mobaku.jp/>)
- ・雨雲レーダー（Yahoo）(<https://weather.yahoo.co.jp/weather/zoomradar/>)
- ・CoVid-19 都道府県別・日別に地図とグラフで可視化 (<https://www.informatix.co.jp/covid-19/#/>)
- ・An open dataset for typical people mass movement in urban areas
(<https://github.com/sekilab/OpenPFLOW>)

memo ┃