

01 視覺化的定義

2

思考資料視覺化的定義。

visualization 的中文為「**視覺化**」，意思是「讓人眼不可見的東西轉換成具體的形狀」，而讓數值這類資訊變得可見的方法就稱為視覺化，讓心中、腦中的想法、知識化為文章的過程也稱為視覺化。

而讓數值、文章或定位資訊這類資料視覺化就稱為資料的視覺化（以下簡稱（資料視覺化）（[圖 1.1](#)）。

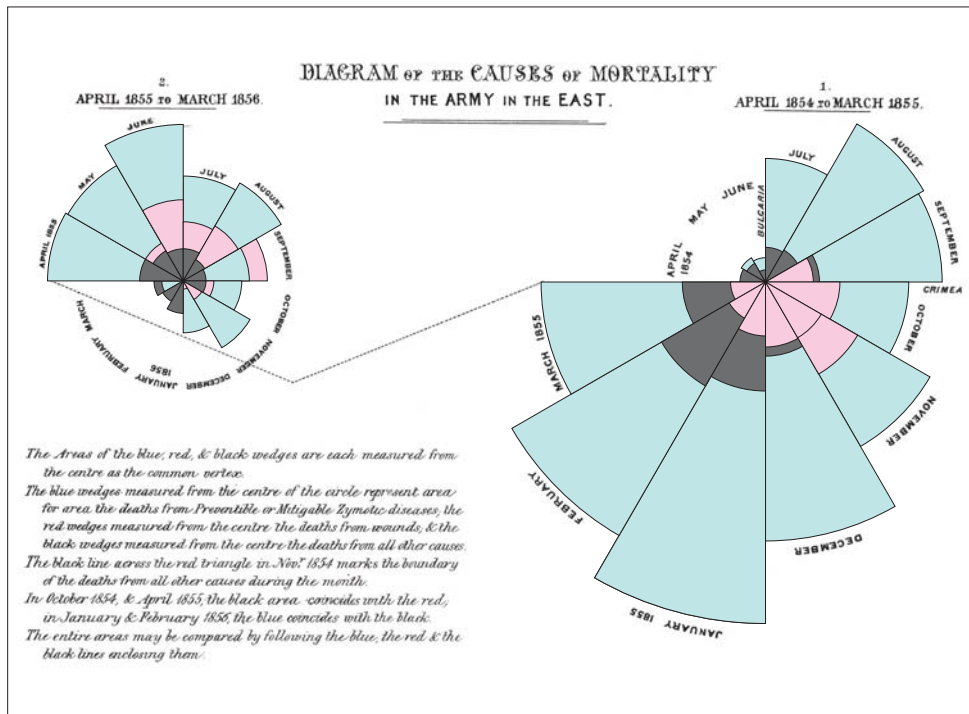


圖 1.1 南丁格爾玫瑰圖

出處：根據：『Diagram of the causes of mortality in the army in the East" by Florence Nightingale』製作

URL https://en.wikipedia.org/wiki/Florence_Nightingale#/media/File:Nightingale-mortality.jpg

02 | 視覺化的歷史

視覺化時至今日仍繼續發展。

02

視覺化的歷史

接著為大家簡單地介紹視覺化的歷史（圖 1.2）。

一般認為，人類視覺化的歷史是從在洞窟的牆壁畫下用棍子狩獵的情況（洞窟壁畫）開始，之後，資訊的視覺化就在各種場景出現，而我們現在使用的各種圖表則是從產業革命爆發的 1700 年代開始。

從那時到現在，人們開始重視以視覺效果呈現數值的技術，這項技術也在這兩百年間急速進化。

- 1700 年代後期

這個時期出現了折線圖、長條圖（圖 1.3）這類我們耳熟能詳的基本圖表。雖然這也是目前常見的視覺化手法，但在當時只畫在紙上。

- 1800 年代

一如著名的南丁格爾玫瑰圖（圖 1.1），這個時代出現了能一口氣清楚呈現多種資訊的視覺化手法。

在 1800 年代，這種視覺化手法成為傳遞社會事件的主流，使用頻率甚至高過在商場使用。

- 1900 年代

自 1900 年代電腦問世後，視覺化就跟著急速發展。

1900 年代前期，首本於商業應用的視覺化專書問世，1970 年代也能看到利用電腦製作的視覺化作品。

- 2000 年代～

到了現代，在商場使用試算表製作圖表的視覺化手法已相當普及。近年來，BI 工具（商業智慧工具）的資料視覺化已慢慢地於企業之間普及，也有越

03 | 身邊常見的視覺化

我們的日常生活有很多視覺化的資訊。

03

身邊常見的視覺化

觸手可及的視覺化資訊

讓我們先來看看日常生活裡的視覺化吧！比方說，我們很常利用智慧型手機瀏覽氣溫的折線圖，確認當天的氣溫，我們也會在電視或新聞報導確認股價的波動，有些電視節目也會利用圓餅圖說明問卷調查的結果。

可見在我們身邊有許多資訊都被**視覺化**為簡單易懂的內容。

大家常收看的天氣預報也常以太陽、雲朵、雨傘這類圖示說明降雨機率，而這些都是視覺化手法的一種（圖 1.4）。

點開智慧型手機的軟體，也能看到許多被視覺化之後的資訊（圖 1.5）。

可見我們的生活裡，真的充斥著許多被視覺化的資訊。

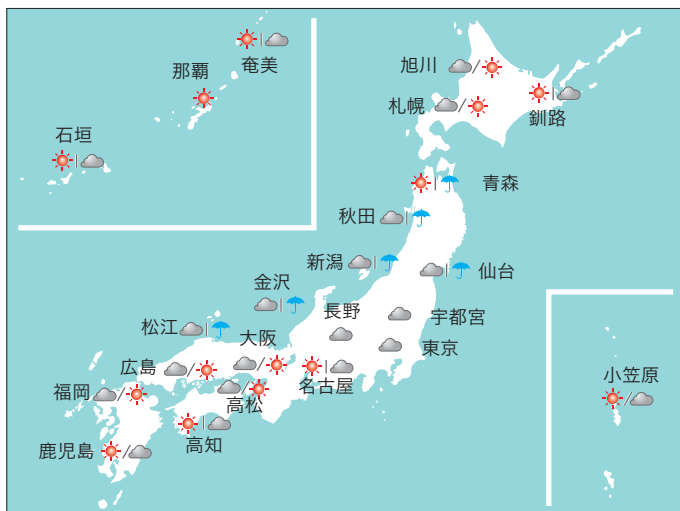


圖 1.4 天氣預報的範例

出處：根據氣象廳「天氣預報」製作

URL <https://www.jma.go.jp/jp/yoho/>

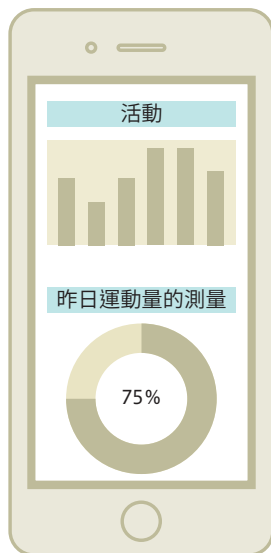


圖 1.5 智慧型手機 App 的範例

02 | 資料的種類與視覺化手法

資料視覺化手法會隨著資料的種類改變。

02

資料的種類與視覺化手法

特性不同的資料適合以不同的視覺化手法呈現，接下來就讓我們看看有哪些視覺化手法。

質化資料與量化資料

資料大致可分成質化與量化兩種，若以尺度區分，還可以分成四種尺度（表 2.1）。

質化資料被分類為「名目尺度」與「次序尺度」的資料，量化資料則被分類為「等距尺度」與「等比尺度」的資料。越接近表格下方的尺度越是經過嚴格測量的資料，也通常包含能以上方尺度呈現的資料。

適用的資料視覺化手法會隨著資料的種類與尺度不同。由於呈現的方式不同，所以有時會以顏色呈現該資料，有時則會改以形狀呈現，換言之在「顏色」與「形狀」這類組成元素的部分會不同。

如果使用了不適當的呈現手法很容易造成誤解，所以必須先了解資料的種類，才能著手視覺化。

資料視覺化的組成元素將於下一節解說。

表 2.1 資料種類與視覺化手法

資料種類	尺度	範例	呈現手法
質化資料 (質化變數)	名目尺度	性別、部門	突顯與其他資料的不同
	次序尺度	排名	突顯與其他資料之間的大小關係
量化資料 (量化變數)	等距尺度	溫度、時間	呈現與其他資料之間的落差
	等比尺度	業績、體重	呈現大小、比例

在同一張圖表裡繪製多個折線圖

接著讓我們將東京、大阪、那霸、函館的折線圖畫成同一張圖表。

第一步要先建立資料框架。年月欄位在 `weather_sample.csv` 之中，是第一個欄位，所以當我們利用 `pd.read_csv` 函數載入 CSV 檔案時，將 `index_col` 指定為 `0`（第一個欄位，也就是年月欄位），就能建立以日期資料的年月欄位為索引值的資料框架。

接著再從這個資料框架篩選出繪製折線圖所需的欄位，就能建立繪製折線圖所需的資料框架（**程式 5.62**）。

程式 5.62 東京、大阪、那霸、函館的平均氣溫資料

In

```
weather_index = pd.read_csv("weather_sample.csv", header=0,
                             parse_dates=["年月"], index_col=0)
tmp_ave = weather_index[["東京 - 平均氣溫 (°C)", "大阪 - 平均氣溫 (°C)",
                          "那霸 - 平均氣溫 (°C)", "函館 - 平均氣溫 (°C)"]]
tmp_ave
```

Out

	東京 - 平均氣溫 (°C)	大阪 - 平均氣溫 (°C)	那霸 - 平均氣溫 (°C)	函館 - 平均氣溫 (°C)
年月				
2015-01-01	5.8	6.1	16.6	-0.9
2015-02-01	5.7	6.9	16.8	0.1
2015-03-01	10.3	10.2	19.0	4.3
2015-04-01	14.5	15.9	22.2	8.3
2015-05-01	21.1	21.5	24.9	13.2
2015-06-01	22.1	22.9	28.7	16.6
(...略...)				
2018-11-01	14.0	14.6	23.1	7.2
2018-12-01	8.3	9.4	20.4	-0.3

到目前為止，我們已經建立了以年月欄位為索引值的各都市平均氣溫資料集，所以讓我們利用 `sns.lineplot` 函數繪製折線圖吧（**程式 5.63**）。執行程式之後，可看到所有的折線圖都集中在一個圖表之內。

此外，這個方法最多可在單一圖表塞進六張折線圖，不過塞太多張折線圖會變得很難閱讀，所以要比較的项目太多時，建議將折線圖拆開來繪製。

01 | 定位資訊的視覺化手法

本章要介紹定位資訊的視覺化手法。

6

定位資訊視覺化手法

對於零售業或觀光業來說，門市、店面的地理資訊非常重要，尤其現在大部分的人都會透過智慧型手機使用定位資訊服務，所以不難想像，各種產業在今後也將大量使用與定位資訊有關的資訊。

除了民間企業之外，政府機關也開始思考如何活用現有的開放資料。

開放資料的經緯度資料通常都是公開的，但只有經緯度的資料是很難有所用處，所以通常會先視覺化經緯度資料，讓這類資料變得有價值。

地圖視覺化的種類

地圖的視覺化手法有很多，例如以點、以面、以線條呈現地點，都是其中之一。

以點呈現的手法就是在地圖標出明顯的某一點，或是在地圖的某個區塊標示較密集的点，而以面呈現的手法就是框出某塊區域，以線條呈現的手法則可說明兩個地點之間的關係（圖 6.1）。

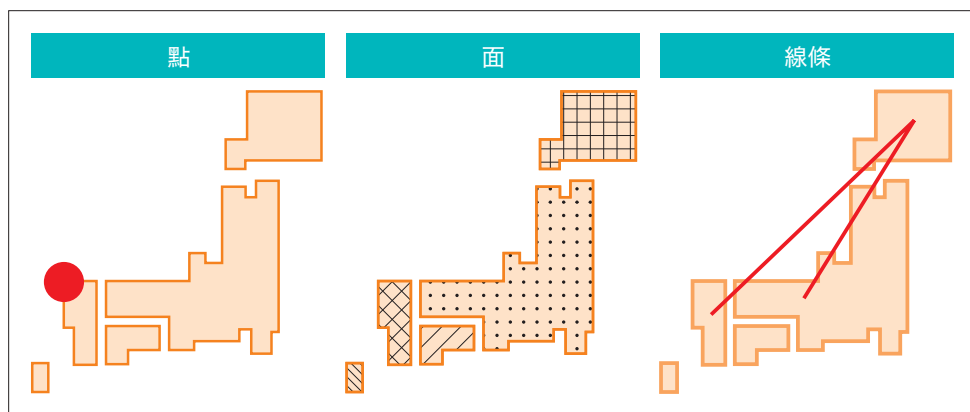


圖 6.1 以點、面、線條呈現的範例

02 | 地圖資訊的視覺化函式庫

介紹本章會用到的視覺化函式庫。

plotly 與 folium

plotly

第五章用來繪製各種圖表的 plotly 也可用來為世界地圖填色，所以可在比較各國資訊時使用。

不過這個函式庫沒有日本各都道府縣的資訊，所以不適合用來呈現日本的地圖資訊。

folium

若要呈現日本與其他國家的地圖資訊，folium 函式庫會比較適合，因為能快速製作分級著色圖，也能在地圖配置大頭針。

要替日本地圖填色就必須先取得日本都道府縣的行政區域資料。執筆之際，找不到可下載日本行政地區資料的網站，所以筆者自己製作了行政區域資料。

• 日本都道府縣的行政區域資訊

URL <https://github.com/kokubonatsumi/Japanmap>

作者將這份行政區域資料放在 GitHub，下載^{*1}之後可於視覺化日本行政地區時使用。請將上面這些檔案與 Jupyter Notebook 檔案放在一起。

讓我們先載入第六章會用到的函式庫（**程式 6.1**）。

程式 6.1 載入函式庫

```
In import plotly.express as px
import folium
import json
import pandas as pd
from branca.colormap import linear
from folium.plugins import HeatMap
```

*1 要下載 GitHub 的檔案時，請從「Code」點選「Download ZIP」，接著解壓縮下載的檔案。本書使用的是「prefs_064」資料夾裡的檔案。

世界地圖的分區標色會以 `folium` 進行（[程式 6.3](#)）。原始的地圖會以 `folium.Map` 函數指定。

`folium.Choropleth` 函數定義了各國疆界的視覺化資訊，而各國疆界的資訊則會從剛剛下載的 `countries.geo.json` 檔案取得。

參數 `data` 指定的是視覺化所需的資料，參數 `columns` 則指定了用於標色的欄位。

參數 `key_on` 指定了 `countries.geo.json` 檔案的各國資訊，參數 `fill_color` 則指定了填色所需的調色盤。

程式 6.3 根據世界各國人均 GDP 標色的範例

In

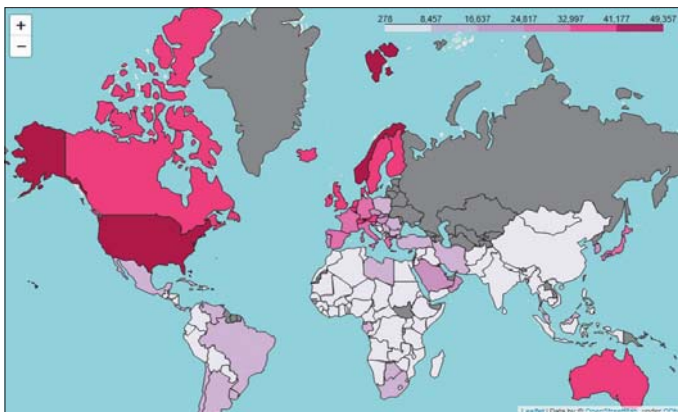
```
base_map = folium.Map(location=[50, 0], zoom_start=1.8)

# 新增 Choropleth
folium.Choropleth(
    geo_data=json.load(open("countries.geo.json", "r")),
    data=gapminder,
    fill_opacity=1,
    line_color = "black",
    nan_fill_color=" #888888",
    columns = ["iso_alpha", "gdpPercap"],
    key_on = "feature.id",
    fill_color = "PuRd",
).add_to(base_map)

base_map
```

使用的資料
填色的透明度
邊界顏色
遺漏值的填色
填色所需的 Key 與欄位名稱
與資料對應的 geo.json 的 Key
與資料對應的 geo.json 的 Key

Out



04 | 中文文章的視覺化手法

介紹繪製中文文字雲常見的問題。

本節要直接將以 wordcloud 函式庫繪製中文文章的文字雲。

這次使用的中文文章為青空文章的《跑吧！美樂斯》（太宰治）。

7

文字資訊的視覺化手法

- 青空文庫

URL <https://www.aozora.gr.jp/>

- 青空文庫：跑吧！美樂斯（太宰治）

URL https://www.aozora.gr.jp/cards/000035/files/1567_14913.html

只要依照繪製英文文字元的步驟處理中文文章，也能繪製中文文字雲，但有時會出現**程式 7.3** 這種只顯示一個句子，無法順利轉換成「文字雲」的狀態。

會出現這個現象是因為中文文章的格式與英文文章不同，不會以空白字元間隔單字。

wordcloud 預設是將英文文章這種以空白字元間隔單字的文章轉換成文字雲，但中文文章的單字通常是連在一起的，直到段落結束之前，都是一個句子，所以直接將中文文章指定給 wordcloud 的話，只會顯示整篇文章，而不會顯示文字雲。

程式 7.3 繪製中文文字雲的範例（錯誤示範）

In

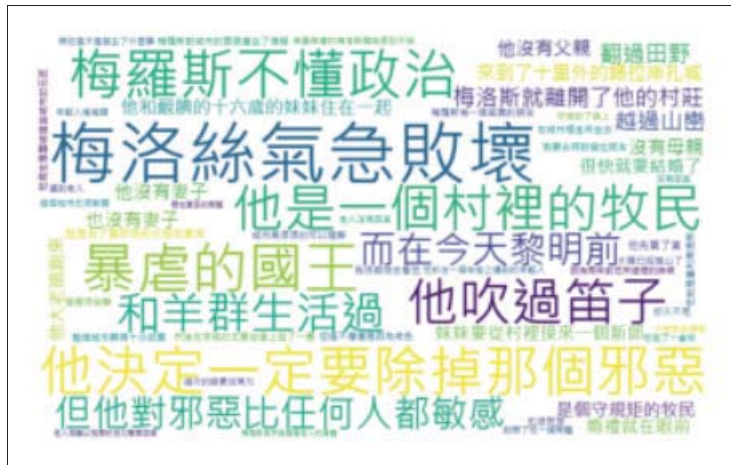
利用半形空白字元切割的中文文章

```
text_tw = " " 梅洛絲氣急敗壞。 他決定一定要除掉那個邪惡、暴虐的國王。 梅羅斯不懂 ➡  
政治。 他是一個村裡的牧民。 他吹過笛子，和羊群生活過。 但他對邪惡比任何人都敏感。 而在 ➡  
今天黎明前，梅洛斯就離開了他的村莊，翻過田野，越過山巒，來到了十里外的錫拉庫扎城。 他沒 ➡  
有父親，沒有母親，也沒有妻子。 他沒有妻子。 他和靦腆的十六歲的妹妹住在一起。 妹妹要從村 ➡  
裡接來一個新郎，是個守規矩的牧民，很快就要結婚了。 婚禮就在眼前。 他大老遠跑來，就是為了 ➡  
買新娘的衣服和宴席。 他先買了貨，然後在京城的主要街道上逛了一圈。 梅羅斯有一個高驕的朋友。  
他就是塞利南提斯。 他現在是雪城這座城市的石匠。 我要去拜訪這位朋友。 好久不見，我很期待 ➡  
去看他。 在城市裡走來走去，梅羅斯對城市的面貌產生了懷疑。 這裡很安靜，也很荒涼。 太陽已 ➡  
經落山了，城市黑漆漆的可以理解，但這不僅僅是因為夜色，整個城市顯得十分寂寞。 無憂無慮的 ➡  
梅洛斯開始感到不安。 他抓住一個在街上遇到的年輕人，問他是不是發生了什麼事，因為兩年前他 ➡  
來這裡的時候，大家都在唱歌，即使到了晚上，這個城市也很熱鬧。 年輕人搖搖頭，沒有回答。 ➡  
他走了一會兒，遇到老人，就問了他一個問題，這次的話更加有力。 老人沒有回答。 梅羅斯用手 ➡  
搖晃著老人的身體，問他更多的問題。 老人用難以捉摸的低沉聲音回答。" " "
```

```
wc = wordcloud.WordCloud(width=1000,  
                           height=600,  
                           background_color=" white",  
                           font_path=r"C:\Windows\Fonts\msjrh.ttf")  
  
wc.generate(text_tw)  
plt.imshow(wc)  
plt.axis("off")
```

Out

(-0.5, 999.5, 599.5, -0.5)



04

中文文章的視覺化手法

02 | 象形圖

介紹常於資訊圖表使用的「象形圖」。

何謂象形圖

在製作資訊圖表時，有一項非常方便實用的工具，那就是「象形圖」。

所謂的象形圖就是具有特定意義的圖案文字，最大的特徵在於以簡單的圖案傳遞意義，而且大部分都是單色的（圖 8.1、圖 8.2）。

在日本常見的象形圖都符合「JIS 規格」，但就算是不符合這項規格的象形圖，只要能一眼了解意思，就適合用來製作資訊圖表。



圖 8.1 象形圖範例①

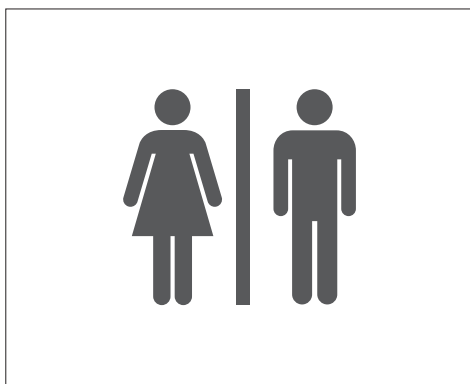


圖 8.2 象形圖範例②

MEMO JIS 規格與 ISO 規格的象形圖

在日本，較常見的象形圖為 JIS 規格，而 ISO 規格的象形圖則是國際標準的象形圖。此外，許多 JIS 規格的象形圖同時也是 ISO 規格的象形圖。

07 | 利用圖片強調比例

介紹以圖片強調比例的方法。

調整圖片的部分顏色是很常用來強調比例的視覺化手法。

利用單一圖片的填色強調比例

要利用單一圖片強調比例時，通常會將圖片塗成不同的顏色。

第一步，讓我們準備單色圖片。

在此使用的是背景透明的黑色人形圖片「woman.png」。

本書使用的是從 ICOOON MONO ([URL https://icooon-mono.com/](https://icooon-mono.com/)) 下載的裙裝人形圖示（參考本章 02 節的 MEMO）。

下載完畢後，請將檔案名稱變更為「woman.png」再與 Jupyter Notebook 的 notebook 檔案放在同一個資料夾。

圖片就緒後，利用 `Image.open` 函數載入圖片（[程式 8.6](#)）。

程式 8.6 載入圖片

In

```
from PIL import Image, ImageOps
from IPython.display import display

im = Image.open("original_icon\woman.png")
display(im)
```