

# econ-viz 套件：個體經濟學圖形

Pin Yue Sung<sup>\*†</sup>      Ling Tak Douglas Chung<sup>\*</sup>

2026-09-26 · v1.12.0 ·  · 

## 目錄

|                          |           |                                                |           |
|--------------------------|-----------|------------------------------------------------|-----------|
| <b>1 簡介</b> .....        | <b>2</b>  | <b>9 主題與設定</b> .....                           | <b>31</b> |
| 1.1 功能範圍 .....           | 2         | 9.1 內建主題 .....                                 | 31        |
| 1.2 閱讀指引 .....           | 2         | 9.2 自訂主題 .....                                 | 31        |
| <b>2 安裝</b> .....        | <b>3</b>  | 9.3 設定檔 .....                                  | 33        |
| 2.1 系統需求 .....           | 3         | <b>10 匯出</b> .....                             | <b>34</b> |
| 2.2 安裝 uv .....          | 3         | 10.1 輸出格式 .....                                | 34        |
| 2.3 安裝 econ-viz .....    | 3         | 10.2 GIF 動畫 .....                              | 34        |
| 2.4 選用依賴 .....           | 3         | 10.3 互動視窗 .....                                | 34        |
| 2.5 安裝命令列工具 .....        | 4         | <b>11 命令列工具</b> .....                          | <b>34</b> |
| 2.6 開發環境設定 .....         | 4         | 11.1 說明與模型 .....                               | 35        |
| 2.7 驗證安裝 .....           | 4         | 11.2 繪圖 .....                                  | 35        |
| <b>3 快速開始</b> .....      | <b>4</b>  | 11.3 建立設定檔 .....                               | 38        |
| 3.1 基本範例 .....           | 4         | 11.4 需求公式 .....                                | 38        |
| 3.2 逐步說明 .....           | 5         | <b>12 動畫</b> .....                             | <b>39</b> |
| <b>4 Canvas 畫布</b> ..... | <b>6</b>  | <b>13 互動元件</b> .....                           | <b>40</b> |
| 4.1 建構函式 .....           | 7         | 13.1 筆記本設定 .....                               | 41        |
| 4.2 繪圖方法 .....           | 8         | 13.2 選擇互動元件或 GIF .....                         | 41        |
| 4.3 樣式 .....             | 12        | <b>14 L<sup>A</sup>T<sub>E</sub>X 解析</b> ..... | <b>41</b> |
| 4.4 串接呼叫 .....           | 14        | 14.1 支援的形式 .....                               | 41        |
| <b>5 多面板圖與需求圖</b> .....  | <b>14</b> | 14.2 錯誤處理 .....                                | 42        |
| 5.1 多面板圖 .....           | 14        | 14.3 在命令列工具中使用 .....                           | 43        |
| 5.2 路徑 .....             | 15        | <b>更新紀錄</b> .....                              | <b>44</b> |
| 5.3 需求圖 .....            | 16        | <b>指令索引</b> .....                              | <b>47</b> |
| 5.4 價格效果分解 .....         | 17        | <b>參考文獻</b> .....                              | <b>49</b> |
| 5.5 Edgeworth 箱形圖 .....  | 19        |                                                |           |
| <b>6 核心模型</b> .....      | <b>20</b> |                                                |           |
| 6.1 平滑偏好 .....           | 20        |                                                |           |
| 6.2 拗折點與角解 .....         | 22        |                                                |           |
| 6.3 所得與參考點 .....         | 24        |                                                |           |
| <b>7 進階模型</b> .....      | <b>27</b> |                                                |           |
| <b>8 分析</b> .....        | <b>28</b> |                                                |           |
| 8.1 比較靜態 .....           | 28        |                                                |           |
| 8.2 Slutsky 矩陣 .....     | 29        |                                                |           |
| 8.3 齊次性 .....            | 29        |                                                |           |
| 8.4 規模報酬 .....           | 30        |                                                |           |

<sup>\*</sup>國立政治大學國際經營與貿易學系，臺北，臺灣。

<sup>†</sup>通訊作者。電子郵件：contact@econ-viz.org

## 第 1 節 簡介

econ-viz 是個體經濟學繪圖套件，涵蓋效用模型、最適消費組合求解與視覺化。Python API 與命令列工具均可產生無異曲線、預算線、消費者均衡、需求曲線及 Edgeworth 箱形圖。

圖形預設顯示第一象限，座標軸以箭頭表示並隱藏數字刻度，標籤支援 TeX 數學語法。輸出格式包括 PNG、PDF、SVG、TikZ 與 GIF。

### 1.1 功能範圍

econ-viz 的功能分為效用模型、均衡求解、效用水準、圖層與匯出格式五個部分。模型與求解結果可分開使用；同一模型也可交由畫布、多面板圖、需求圖或分析 API 處理。

### 1.2 閱讀指引

本手冊依主題分成四個部分（參見表 1）：

表 1  
章節主題

| 主題     | 子主題              | 說明         | 章節     |
|--------|------------------|------------|--------|
| 繪圖與匯出  | 畫布與圖層            | 座標軸、曲線與均衡點 | 第 4 節  |
|        | 多面板圖與需求圖         | 並排比較與需求曲線  | 第 5 節  |
|        | 主題與設定檔           | 配色、樣式與設定檔  | 第 9 節  |
|        | 輸出格式             | 匯出圖檔與 TikZ | 第 10 節 |
| 模型與分析  | 內建模型             | 常見效用函數     | 第 6 節  |
|        | 自訂與多商品模型         | 自訂函數與多商品   | 第 7 節  |
|        | 比較靜態與 Slutsky 矩陣 | 需求導數與齊次性   | 第 8 節  |
| 動畫與互動  | GIF 動畫           | 參數變動的 GIF  | 第 12 節 |
|        | Jupyter 互動元件     | 筆記本滑桿調參數   | 第 13 節 |
| 其他輸入方式 | 命令列工具            | 使用命令列產圖    | 第 11 節 |
|        | LaTeX 效用函數解析     | 從算式建立模型    | 第 14 節 |

初次使用時，依序閱讀第 3 節與第 4 節。查詢特定 API 時，可直接前往對應章節或指令索引。

<sup>1</sup>Python 官方網站提供各作業系統的安裝程式：<https://www.python.org/downloads/>。

<sup>2</sup>uv 是 Astral 開發的 Python 套件與專案管理工具，速度快且可一併管理 Python 版本；安裝方式與完整說明見官方文件：<https://docs.astral.sh/uv/>。

## 第 2 節 安裝

### 2.1 系統需求

econ-viz 需要 Python 3.10 以上版本<sup>1</sup>。本章以 uv<sup>2</sup> 管理套件與專案，並附上對應的 pip 指令。

安裝套件時會一併安裝 NumPy、SciPy、matplotlib 與 SymPy，分別用於陣列運算、數值求解、繪圖與符號運算<sup>3</sup>。

### 2.2 安裝 uv

本手冊的指令以 uv 為準。既有專案仍可使用 pip、pipx 或 Poetry；套件 API 不受管理工具影響。

依作業系統執行下列安裝指令。

#### macOS、Linux

```
curl -Lsf https://astral.sh/uv/install.sh | sh
```

#### Windows

```
powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"
```

### 2.3 安裝 econ-viz

建立專案並加入 econ-viz：

```
uv init my-diagrams
cd my-diagrams
uv add econ-viz
```

使用 uv run 在專案環境中執行程式：

```
uv run python main.py
```

將第 3 節的基本範例存為專案目錄下的 main.py，再執行上述指令。uv add 記錄專案依賴，uv run 使用該專案的 Python 環境。安裝與執行必須使用同一個環境。

若要固定本手冊使用的版本，可在加入依賴時指定版本號：

```
uv add "econ-viz==1.12.0"
```

若使用既有的 Python 虛擬環境，可透過 pip 安裝：

```
python -m pip install -U econ-viz
```

套件名稱是 econ-viz，Python 匯入名稱是 econ\_viz：

```
from econ_viz import Canvas, solve
```

import 陳述式不得使用連字號。若發生 ModuleNotFoundError，檢查執行程式的直譯器是否與安裝套件時使用的環境相同。

### 2.4 選用依賴

動畫與 Jupyter 互動元件需要選用依賴。依用途安裝其中一組：

---

<sup>3</sup>一般 Python 繪圖不需要另外安裝 L<sup>A</sup>T<sub>E</sub>X；只有要編譯匯出的 TikZ 原始碼時，才需要 L<sup>A</sup>T<sub>E</sub>X 環境。

animation 供 Animator 寫出 GIF（詳見第 12 節）；安裝 Pillow。  
 interactive 供 WidgetViewer 在 Jupyter 顯示控制項（詳見第 13 節）；安裝 ipywidgets 與 IPython。  
 all 全部選用依賴。

```
uv add "econ-viz[animation]" # GIF 匯出 (Pillow)
uv add "econ-viz[interactive]" # 筆記本互動元件
uv add "econ-viz[all]" # 全部選用依賴
```

安裝選用依賴不會自動執行動畫或開啟筆記本，仍需依各章範例呼叫對應的 API。

## 2.5 安裝命令列工具

僅使用命令列介面時，可將 econ-viz 安裝為獨立工具（詳見第 11 節）：

```
uv tool install econ-viz
```

此方式將命令列工具安裝在獨立環境。需要在 Python 程式中匯入套件時，仍須在該專案執行 `uv add econ-viz`。在專案內以 `uv run econ-viz` 呼叫工具，可讓命令列與 Python 程式使用同一版本。

## 2.6 開發環境設定

```
git clone https://github.com/EconViz/econ-viz.git
cd econ-viz
uv sync --all-extras
```

其中 `uv sync --all-extras` 會安裝開發依賴與所有選用依賴。完成後執行測試：

```
uv run pytest
```

## 2.7 驗證安裝

```
uv run econ-viz --version # econ-viz 1.12.0
uv run econ-viz help
```

註解中的版本號僅為範例，實際輸出依安裝版本而定。也可用以下指令確認 Python 能匯入繪圖與求解介面：

```
uv run python -c "from econ_viz import Canvas, solve; print('OK')"
```

在伺服器或其他沒有圖形介面的環境中，請以 `save()` 或命令列的 `--output` 輸出檔案。`show()` 需要可用的互動式繪圖後端，視窗打不開不代表安裝失敗。

# 第 3 節 快速開始

## 3.1 基本範例

本章以 Cobb-Douglas 消費者問題說明完整流程：

$$\begin{aligned} \max_{x,y} u(x,y) &= x^{\frac{1}{2}} y^{\frac{1}{2}} \\ \text{s.t. } 2x + 3y &= 30 \end{aligned}$$

程式先求解最適消費組合，再繪製無異曲線、預算集合與均衡點（參見圖 1）。

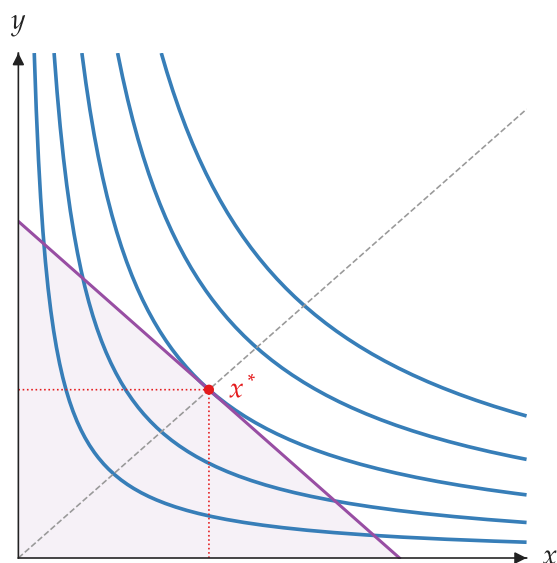


圖 1  
消費者均衡。

### 例 1

```
from econ_viz import Canvas, levels, solve
from econ_viz.models import CobbDouglas

model = CobbDouglas(alpha=0.5, beta=0.5)
eq = solve(model, px=2.0, py=3.0, income=30.0)
lvls = levels.around(eq.utility, n=5)

cvs = Canvas(
    x_max=20, y_max=15,
    x_label="x", y_label="y",
    title=r"Cobb-Douglas  $x^{0.5} y^{0.5}$ "
)
cvs.add_utility(model, levels=lvls)
cvs.add_budget(2.0, 3.0, 30.0, fill=True)
cvs.add_equilibrium(eq, show_ray=True)
cvs.save("cobb_douglas.png")
```

執行主程式 main.py 後，會在目前工作目錄建立 cobb\_douglas.png。本例的最適消費量為  $x^* = 7.5$ 、 $y^* = 5$ ，支出為  $2 \times 7.5 + 3 \times 5 = 30$ ，恰好等於所得。預算線的兩個軸截距則為  $\frac{I}{p_x} = 15$  與  $\frac{I}{p_y} = 10$ 。

## 3.2 逐步說明

繪圖流程依序為：建立模型、求解均衡、選取效用水準、建立畫布、加入圖層與匯出。

```
econ_viz.models from econ_viz.models import CobbDouglas
model = CobbDouglas(alpha=0.5, beta=0.5)
```

建立效用模型。內建模型詳見第 6 節；自訂函數使用 CustomUtility，詳見第 7 節。

```
solve solve(<模型>, px=<浮點數>, py=<浮點數>, income=<浮點數>)
```

依價格與所得求解，回傳不可變的 Equilibrium 物件。

x 最適消費量  $x^*$ 。

float

y 最適消費量  $y^*$ 。

float

- `utility` 模型在最適點的效用值。
- `float`
- `bundle_type` 解的類型："interior" 為內部解、"boundary" 為數值解落在數量下界（例如 Stone-Geary 的最低消費限制）、"kink" 為完全互補的拗折解、"corner" 為完全替代的角解。
- `str`
- ```
from econ_viz import solve
eq = solve(model, px=2.0, py=3.0, income=30.0)
print(eq.x, eq.y, round(eq.utility, 3))

# 7.5 5.0 6.124
```
- `levels.around` `levels.around(<效用值>, n=<整數>)`  
以指定效用值為中心，產生  $n$  個效用水準。圖 1 使用  $n=5$ ，因此繪製五條無異曲線。  
`levels.around()` 會保留指定的效用值，均衡點會落在其中一條無異曲線上。若直接傳入 `levels=5`，畫布會依取樣到的效用範圍選取水準，不保證包含均衡效用。要比較多張圖上的同一組偏好，可先計算一次 `lvls`，再讓各張圖使用相同水準。
- `Canvas` `Canvas(x_max=<浮點數>, y_max=<浮點數>, ...)`  
建立畫布並設定座標範圍，其餘參數詳見第 4 節。  
座標範圍只控制顯示區域，不參與 `solve()` 的計算。均衡點未出現在圖中時，檢查 `eq.x` 與 `eq.y` 是否超出範圍，再調整座標上限。
- `Canvas.add_*` 在同一張畫布上加入無異曲線、預算線或均衡點。這些方法可以串接呼叫（詳見第 4.4 節）。`add_equilibrium()` 使用既有的求解結果，不會重新求解；若之後修改價格、所得或模型，需重新呼叫 `solve()`，並建立對應的新圖。
- `Canvas.save` `save()` 依副檔名匯出檔案（詳見第 10 節）；`show()` 開啟 `matplotlib` 互動視窗。輸出路徑以執行程式時的工作目錄為準，若需指定子目錄，請事先建立目錄。
- `Canvas.show`

### 檢查輸入與求解失敗

價格與所得必須為正；輸入不符合條件時，`solve()` 拋出 `InvalidParameterError`<sup>4</sup>。

求解未收斂時，`solve()` 拋出 `OptimizationError`，不會回傳無效的均衡點。

```
from econ_viz import InvalidParameterError, OptimizationError, solve

try:
    eq = solve(model, px=2.0, py=3.0, income=30.0)
except (InvalidParameterError, OptimizationError) as error:
    print(error)
else:
    print(round(eq.x, 3), round(eq.y, 3), eq.bundle_type)
```

數值解可能有微小誤差，例如結果接近但不恰好等於 7.5。驗證時應使用適當容許誤差；顯示結果時再用 `round()` 格式化，避免過早四捨五入影響後續計算。

## 第 4 節 Canvas 畫布

`Canvas` 以 `matplotlib` 的 `Figure` 與 `Axes` 為基礎，提供無異曲線、預算線與均衡點等圖層。

<sup>4</sup>一般平滑模型使用序列最小平方法 (sequential least squares programming, SLSQP) 求解。

4.1 建構函式

```
Canvas from econ_viz import ArrowStyle, Canvas, Stroke, themes

cvs = Canvas(
    x_max=20,
    y_max=15,
    x_label="x",
    y_label="y",
    title=r"Cobb-Douglas  $x^{0.5} y^{0.5}$ ",
    dpi=300,
    x_label_pos="right", # "top"、"right" 或 "bottom"
    y_label_pos="top", # "left"、"top" 或 "right"
    font="DejaVu Sans",
    math_font="stix",
    axis_stroke=Stroke(width=1.0, arrow=ArrowStyle.TRIANGLE),
    theme=themes.default,
)
```

更新：v1.7.0 建立畫布，並設定座標範圍、軸標籤、字型、線條樣式與主題。未明確傳入的視覺設定取自目前的 Config 與 Theme。

x\_max 橫軸上限。

float · 預設 10

y\_max 縱軸上限。

float · 預設 10

x\_label 橫軸末端的標籤。

str · 預設 "x"

y\_label 縱軸末端的標籤。

str · 預設 "y"

title 圖形標題。

str | None · 預設 None

dpi 點陣圖匯出解析度，限制在 1-1200 之間。

int · 預設 300

x\_label\_pos 橫軸標籤相對箭頭的位置，也接受 LabelPosition（參見表 2）。

str · 預設 "right"

表 2

| 值        | 位置   |
|----------|------|
| "top"    | 箭頭上方 |
| "right"  | 箭頭右側 |
| "bottom" | 箭頭下方 |

y\_label\_pos 縱軸標籤相對箭頭的位置（參見表 3）。

str · 預設 "top"

表 3

| 值       | 位置   |
|---------|------|
| "left"  | 箭頭左側 |
| "top"   | 箭頭上方 |
| "right" | 箭頭右側 |

`font` 所有文字使用的字型，或候選字型清單。此項設定僅作用於指定畫布，不會更動 `matplotlib` 的全域設定。

`math_font` `matplotlib` 的數學字型：`dejavusans`、`dejavuserif`、`cm`、`stix` 或 `stixsans` 等。

`str` · 預設 `None`

`axis_stroke` 同時設定兩軸的粗細、線條樣式、顏色與箭頭（詳見第 4.3 節）。

`Stroke` · 預設 `theme`

`x_axis_stroke` 單獨覆寫橫軸。

`Stroke` · 預設 `None`

`y_axis_stroke` 單獨覆寫縱軸。

`Stroke` · 預設 `None`

`theme` 配色與樣式主題（詳見第 9 節）。

`Theme` · 預設 `themes.default`

## 4.2 繪圖方法

`add_*` 方法在目前的畫布加入圖層，並回傳同一個 `Canvas`，因此可串接呼叫（詳見第 4.4 節）。

```
add_utility    cvs.add_utility(
                func,
                levels=3,           # 曲線條數或效用值清單
                color=None,         # 預設 theme.ic_color
                linewidth=None,     # 預設 theme.ic_linewidth
                show_rays=False,
                show_kinks=False,
                kink_radius=1.0,
                show_bliss=True,    # 標出極樂點 (Satiation)
                show_ic_labels=False,
                stroke=None,
                ray_stroke=None,
                ic_label=None,
                highlight_level=None,
                secondary_stroke=None,
                label_style="numeric",
            )
```

繪製無異曲線。

`levels` 曲線條數或效用值清單；例如 `levels.around(eq.utility, n=5)` 會在均衡效用附近選

`int` | `list` · 預設 3 取五個水準。

`highlight_level` 焦點效用值。實際繪製時選取最接近的曲線，以主要樣式顯示。

`float` | `None` · 預設 `None`

`secondary_stroke` 其他曲線的樣式；省略時使用 `theme.secondary_ic_stroke`。

`Stroke` | `None` · 預設 `None`

`show_ic_labels` 顯示曲線標籤。標籤沿曲線切線旋轉，並保留在可見繪圖區域內（參見圖 3）。

`bool` · 預設 `False`

`label_style` "numeric" 顯示效用值；"ordinal" 依效用遞增順序顯示  $u_1, u_2, \dots$ 。

`str` · 預設 "numeric"

```
lvls = levels.around(eq.utility, n=5)
cvs.add_utility(
    model,
    levels=lvls,
```

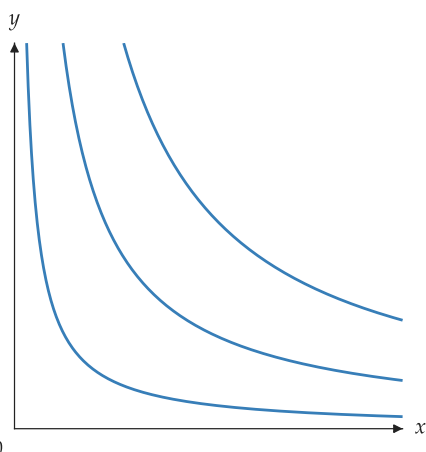


圖 2  
三條無異曲線。

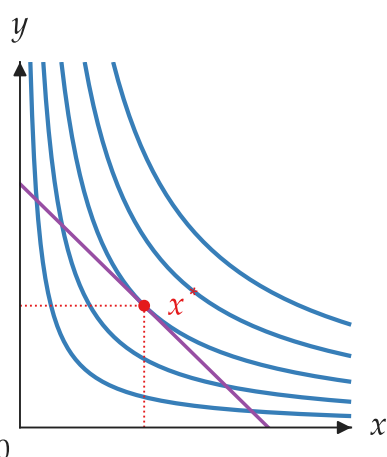
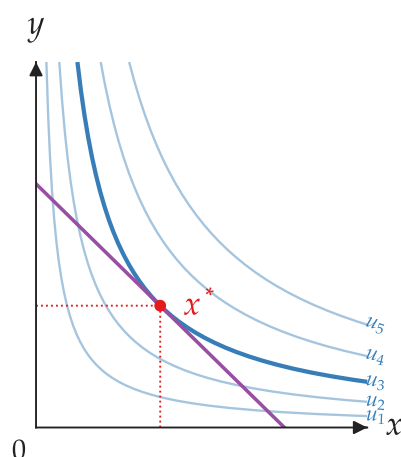


圖 3  
無異曲線主次層級。



```
highlight_level=eq.utility,
show_ic_labels=True,
label_style="ordinal",
)
```

```
add_budget cvs.add_budget(
    px, py, income,
    color=None,
    linewidth=None,
    linestyle="--",
    label=None,          # 圖例標籤 (LaTeX)
    fill=False,          # 可行集合陰影
    fill_alpha=None,     # 預設 theme.budget_fill_alpha
    stroke=None,
)
```

繪製預算線  $p_x x + p_y y = I$ 。

`fill` True 使用主題的陰影設定；傳入 `Fill` 可覆寫陰影顏色與透明度。

`bool` | `Fill` · 預設 False

`stroke` 統一設定預算線的顏色、線寬、線型與透明度，優先於 `color`、`linewidth`、`linestyle`

`Stroke` | `None` · 預設 None 等簡寫參數（詳見第 4.3 節）。

`label` 圖例文字；呼叫 `show_legend()` 後才會顯示圖例。

`str` | `None` · 預設 None

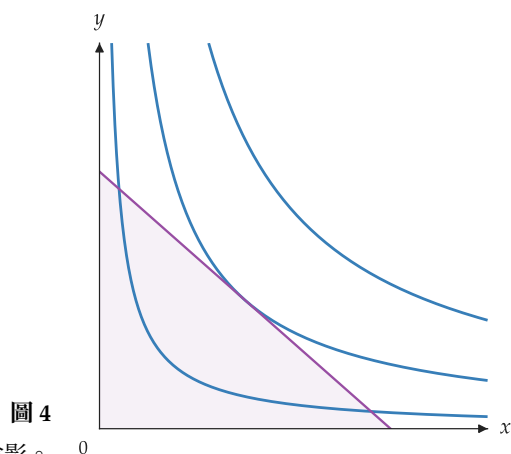


圖 4  
預算線與陰影。

```
add_equilibrium    cvs.add_equilibrium(
                    eq,                                # solve() 的回傳值
                    color=None,
                    markersize=None,
                    label="x*",
                    drop_dashes=True,                  # 到兩軸的虛線
                    show_ray=False,                   # 擴張路徑
                    drop_stroke=None,
                    ray_stroke=None,
                    marker=None,
                )
```

標記 solve() 回傳的最適消費組合。

drop\_dashes 顯示至兩軸的投影線。

bool · 預設 True

show\_ray 顯示通過原點與均衡點的射線。

bool · 預設 False

marker 均衡點的標記樣式（詳見第 4.3 節）。

Marker | None · 預設 None

```
add_ray            cvs.add_ray(
                    slope,                             # dy/dx
                    color=None,
                    linewidth=None,
                    stroke=None,
                )
```

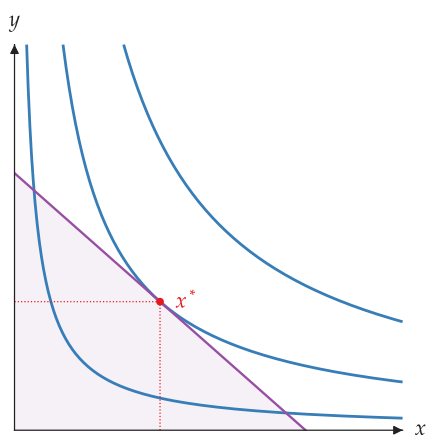


圖 5  
均衡點與虛線。

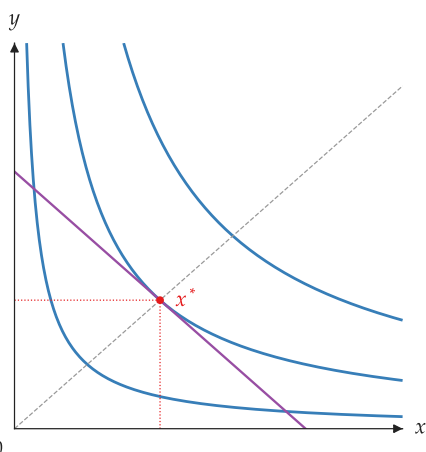


圖 6  
擴張路徑射線。

繪製通過原點、斜率為  $\text{slope}$  ( $dy/dx$ ) 的射線，用於標記擴張路徑或固定比例。

```
add_point    cvs.add_point(
              x, y,
              label=None,
              color=None,
              markersize=None,
              offset=None,      # 標籤位移 (pt)
              marker=None,
            )
```

在指定座標標記點。

label 標記文字，可傳入字串或 Label。

str | Label | None · 預設 None

offset 文字相對標記點的位移，單位為 pt。

tuple | None · 預設 None

marker 標記樣式。

Marker | None · 預設 None

```
show    cvs.show()          # 互動視窗
save    cvs.save("figure.png") # .png / .pdf / .svg / .tex
```

show() 開啟互動視窗；save() 依副檔名匯出檔案（詳見第 10 節），並釋放對應的 matplotlib 圖形，因此應最後呼叫。

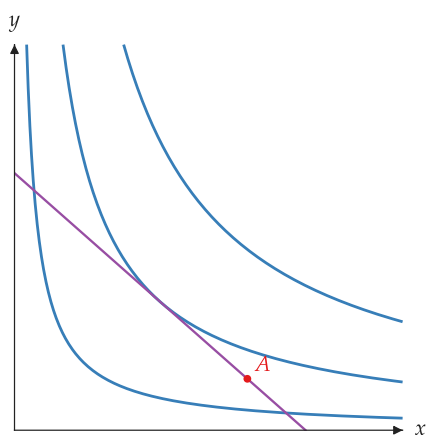


圖 7  
標記點 A。

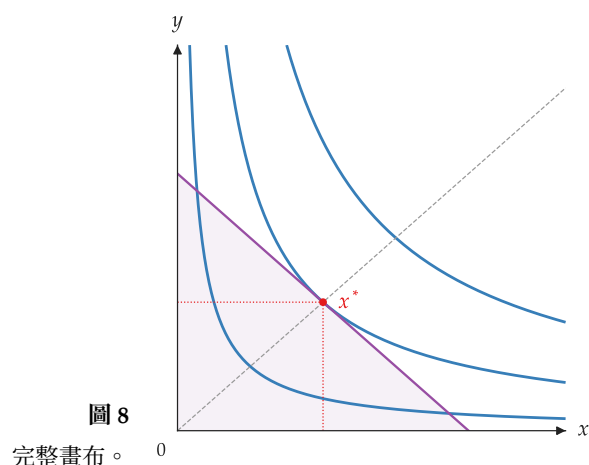


圖 8

完整畫布。

### 4.3 樣式

`LineStyle` 定義線型，`ArrowStyle` 定義箭頭樣式，`Stroke` 統一設定線寬、線型、顏色與箭頭，可套用於座標軸、預算線、路徑與投影線。

#### 線條樣式

`LineStyle` `LineStyle.SOLID`、`DASHED`、`DOTTED`、`DASHDOT`

新增：v1.7.0 提供實線、虛線、點線與點畫線（參見圖 9）。座標軸接受列舉值或對應字串；其他線條透過 `Stroke(style=...)` 設定。

```
import matplotlib.pyplot as plt

from econ_viz import Canvas, LineStyle

styles = [
    LineStyle.SOLID,
    LineStyle.DASHED,
    LineStyle.DOTTED,
    LineStyle.DASHDOT,
]

fig, axes = plt.subplots(1, 4, figsize=(8.5, 2.2))
for ax, style in zip(axes, styles):
    Canvas(
        title=style.value,
        x_line_style=style,
        y_line_style=style,
        fig=fig,
        ax=ax,
    )

# 軸標籤會放在箭頭外側；四張圖並排時要多留一點水平間距。
fig.subplots_adjust(wspace=0.62)
fig.savefig("line_styles.png", dpi=160, transparent=True)
```

#### 箭頭樣式

`ArrowStyle` `ArrowStyle.SIMPLE`、`TRIANGLE`、`FANCY`、`WEDGE`

新增：v1.7.0 提供四種箭頭樣式（參見圖 10）。座標軸透過 `x_arrow_style`、`y_arrow_style` 設定；其他線條透過 `Stroke(arrow=...)` 設定。

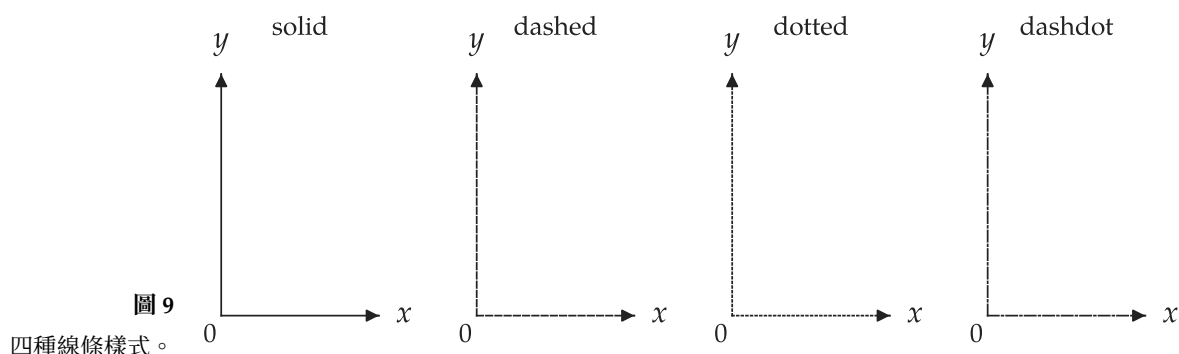


圖 9  
四種線條樣式。

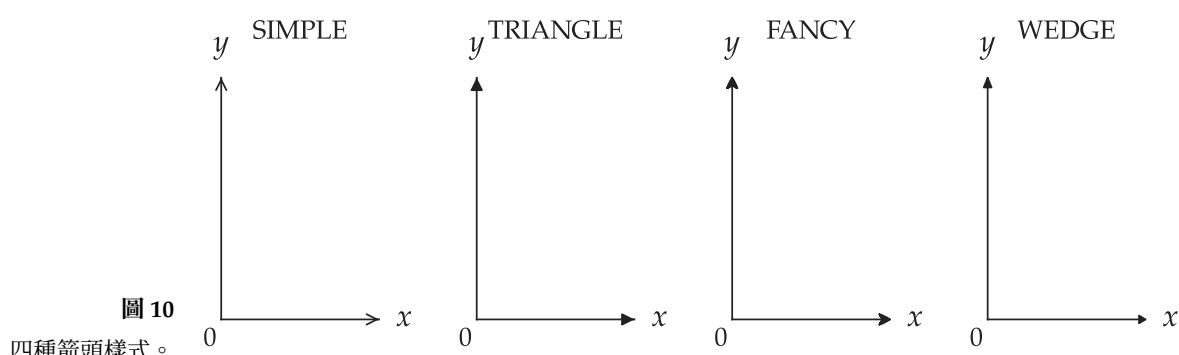


圖 10  
四種箭頭樣式。

### Stroke

Stroke Stroke(width=⟨浮點數⟩, style=⟨線條樣式⟩, color=⟨字串⟩, arrow=⟨箭頭樣式⟩, opacity=⟨浮點數⟩)

新增：v1.7.0 設定線寬、線型、顏色、箭頭與透明度。未指定的欄位沿用目前主題。opacity 的有效更新：v1.9.0 範圍為 0 至 1。

### Axis

Axis Axis(label=None, label\_position=None, stroke=None)

新增：v1.8.0 集中設定單一座標軸的標籤、標籤位置與線條。Canvas、Figure、DemandDiagram 與 EdgeworthBox 均接受 x\_axis 與 y\_axis。若同時傳入簡寫參數，Axis 中已設定的欄位優先。

```
from econ_viz import Axis, Canvas, Label, Stroke

cvs = Canvas(
    x_axis=Axis(
        label=Label(text="x_1", fontsize=16),
        label_position="bottom",
        stroke=Stroke(width=1.2),
    ),
    y_axis=Axis(label="x_2"),
)
```

### Marker 與 Label

Marker Marker(color=None, size=None, shape=None, opacity=None)

新增：v1.8.0 設定標記點的顏色、大小、形狀與透明度。shape 接受 matplotlib 的標記字串，例如更新：v1.9.0 "o"、"s"、"^"、"D" 與 "\*"。

Label Label(text=None, position=None, offset=None, color=None, fontsize=None, visible=None, opacity=None)

新增：v1.8.0 設定文字、位置、位移、顏色、字級、可見性與透明度。位置可為上、下、左、右或四

更新：v1.9.0 個角落；Label(visible=False) 隱藏對應文字。

```
from econ_viz import Label, Marker

cvs.add_equilibrium(
    eq,
    marker=Marker(shape="s", size=8),
    label=Label(position="bottom-left", offset=8),
)
cvs.add_point(
    12, 2,
    marker=Marker(color="black", shape="D"),
    label=Label(text="A", position="left", fontsize=14),
)
```

### Fill

Fill Fill(color=None, opacity=None)

新增：v1.8.0 設定陰影區域的顏色與透明度。alpha 是 opacity 的相容簡寫；同一次呼叫不得傳入不

更新：v1.9.0 同的 alpha 與 opacity。

```
from econ_viz import Fill

cvs.add_budget(
    2, 3, 30,
    fill=Fill(color="lightgrey", opacity=0.4),
)
```

## 4.4 串接呼叫

add\_\* 方法支援串接呼叫，最後以 save() 匯出：

```
Canvas(x_max=20, y_max=15) \
    .add_utility(model, levels=lvls) \
    .add_budget(2.0, 3.0, 30.0, fill=True) \
    .add_equilibrium(eq, show_ray=True) \
    .save("figure.png")
```

## 第 5 節 多面板圖與需求圖

### 5.1 多面板圖

Figure Figure(<版面>, x\_max=<浮點數>, y\_max=<浮點數>, ..., shared\_x=False, shared\_y=False)

新增：v1.2.0 建立多面板圖，座標範圍與樣式參數與 Canvas 相同。使用 fig[idx] 取得面板，再呼叫 add\_utility()、add\_budget() 等繪圖方法。

## 例 2

```

from econ_viz import Figure, Layout, Legend, levels, solve
from econ_viz.models import CobbDouglas

fig = Figure(
    Layout.SIDE_BY_SIDE,
    x_max=20,
    y_max=15,
    x_label="x",
    y_label="y",
    title="Before / After Price Change",
    shared_y=True,
)

cases = [
    (CobbDouglas(alpha=0.5, beta=0.5), 2.0, 3.0, 30.0, r"Before: $p_x=2$"),
    (CobbDouglas(alpha=0.3, beta=0.7), 4.0, 3.0, 30.0, r"After: $p_x=4$"),
]

for idx, (model, px, py, income, title) in enumerate(cases):
    eq = solve(model, px=px, py=py, income=income)
    panel = fig[idx]
    panel.ax.set_title(title)
    panel.add_utility(model, levels=levels.around(eq.utility, n=5), label="IC")
    panel.add_budget(px, py, income, fill=True, label="BC")
    panel.add_equilibrium(eq, show_ray=True)

fig[0].show_legend(legend=Legend(position="upper right"))
fig.save("figure_side_by_side.png")

```

Layout 內建版面有 SINGLE、STACKED、SIDE\_BY\_SIDE、TOP\_TWO\_BOTTOM\_ONE、TOP\_ONE\_BOTTOM\_TWO、GRID\_2X2 和 GRID\_3X3。

## 5.2 路徑

PricePath `from econ_viz import IncomePath, LinearBudget, PricePath`  
IncomePath `from econ_viz.models import CobbDouglas`

```

model = CobbDouglas(alpha=0.5, beta=0.5)
budget = LinearBudget(px=2.0, py=2.0, income=40.0)

price_path = PricePath(
    model,
    budget=budget,
    price="px",
    price_range=(0.8, 6.0),
    n=40,
)

```

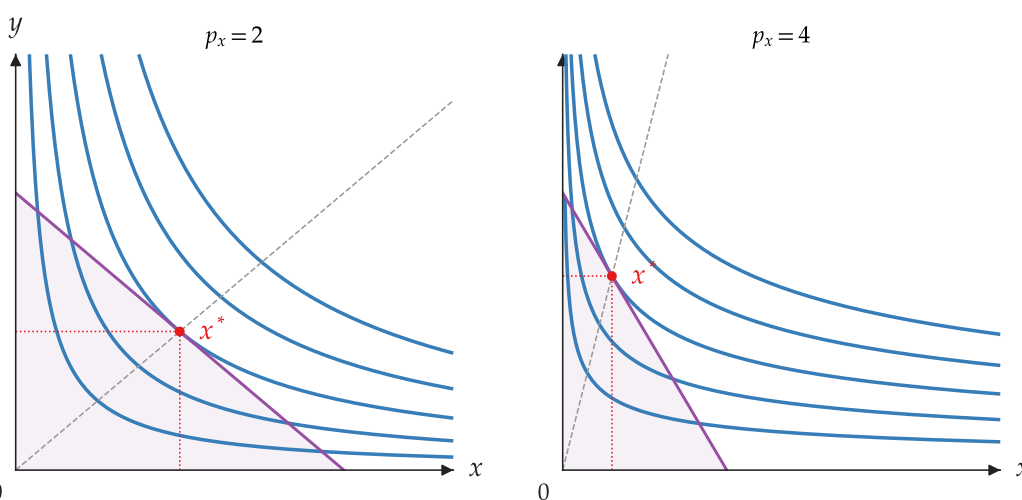


圖 11

價格變動前後比較。

```
income_path = IncomePath(
    model,
    budget=budget,
    income_range=(20.0, 80.0),
    n=30,
)
```

新增：v1.2.0 PricePath 與 IncomePath 分別計算價格、所得變動時的最適消費組合。使用 Canvas.add\_path() 繪製價格消費曲線（PCC）或所得消費曲線（ICC）；PricePath 也可用於需求圖。

```
Canvas.add_path from econ_viz import Canvas, levels

eq = price_path.equilibria[len(price_path.equilibria) // 2]
lvls = levels.around(eq.utility, n=5)

Canvas(x_max=25, y_max=20) \
    .add_utility(model, levels=lvls) \
    .add_path(price_path, label="PCC") \
    .save("price_path.png")
```

新增：v1.2.0 將 PricePath 或 IncomePath 加入畫布。圖 12 中，Cobb-Douglas 模型在  $p_x$  變動時， $y$  的需求量不變，因此 PCC 為水平線。

### 5.3 需求圖

```
DemandDiagram DemandDiagram(<價格路徑>, title=None, ...)

.add_marshallian_panel(price_markers=None, show_pcc=False,
    show_demand_guides=True)
```

新增：v1.2.0 依 PricePath 繪製兩個面板：上方為預算線與最適消費組合，下方為對應的 Marshall 需求曲線（參見圖 13）。目前只接受 PricePath，並分別處理平滑、拗折與角解。show\_pcc=True 在上方面板顯示價格消費曲線。

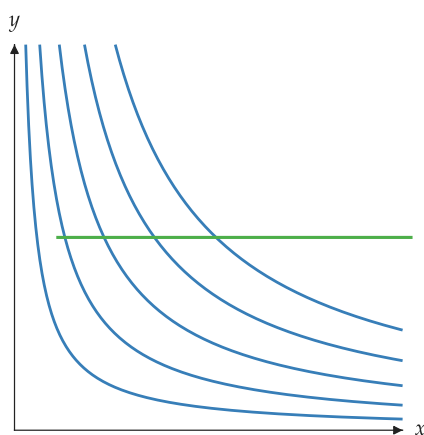


圖 12  
價格消費曲線。

## 例 3

```

from econ_viz import DemandDiagram, LinearBudget, PricePath
from econ_viz.models import CobbDouglas

model = CobbDouglas(alpha=0.5, beta=0.5)
budget = LinearBudget(px=2.0, py=2.0, income=40.0)
path = PricePath(
    model,
    budget=budget,
    price="px",
    price_range=(0.8, 6.0),
    n=40,
)

fig = DemandDiagram(path, title="Demand: Cobb-Douglas")
fig.add_marshallian_panel(
    price_markers=[1.5, 4.0],
    show_pcc=False,
    show_demand_guides=True,
)
fig.save("demand_cobb_douglas.png")

```

## 5.4 價格效果分解

`decompose_price_effect` `decompose_price_effect(<模型>, px=<舊值>, <新值>), py=<浮點數>, income=<浮點數>, method=<方法>)`

新增：v1.5.0 將價格效果分解為替代效果與所得效果。Hicks 補償 (Hicks, 1939) 維持原效用水準；Slutsky 補償 (Slutsky, 1915) 則使原消費組合在新價格下恰好可負擔。回傳值包含下列欄位：

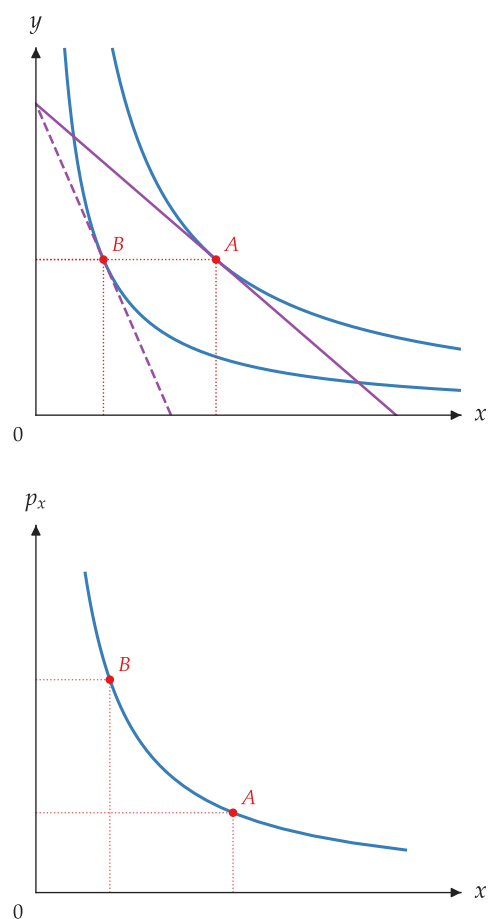


圖 13

需求曲線最適點。

|                         |                                                                                                                                   |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| A, B, C                 | 原始、補償後與最終的消費組合。                                                                                                                   |
| substitution_effect     | 替代效果。                                                                                                                             |
| income_effect           | 所得效果。                                                                                                                             |
| total_effect            | 總效果。                                                                                                                              |
| compensated_income      | 補償後的所得。                                                                                                                           |
| Canvas.                 | 在畫布上標示 A、B、C 三個消費組合、預算線、無異曲線與效果箭頭。Hicks 分解繪製通過 A 與 C 的曲線；Slutsky 分解另繪製通過 B 的曲線。                                                   |
| add_decomposition       |                                                                                                                                   |
| 新增：v1.5.0               |                                                                                                                                   |
| 更新：v1.9.0               |                                                                                                                                   |
| show_curves             | 繪製分解用的無異曲線；已有自訂無異曲線時設為 False，避免重複。                                                                                                |
| bool · 預設 True          |                                                                                                                                   |
| curve_stroke            | 無異曲線的樣式。                                                                                                                          |
| Stroke   None · 預設 None |                                                                                                                                   |
| curve_label             | 顯示並設定 $U_0$ 、 $U_1$ 與 $U_B$ 標籤。                                                                                                   |
| Label   None · 預設 None  |                                                                                                                                   |
| substitution, income    | 替代效果與所得效果的樣式（參見下方 Effect）。                                                                                                        |
| Effect   None · 預設 None |                                                                                                                                   |
| point_marker            | 消費組合的標記。                                                                                                                          |
| Marker   None · 預設 None |                                                                                                                                   |
| point_label             | 消費組合的文字。                                                                                                                          |
| Label   None · 預設 None  |                                                                                                                                   |
| legend                  | 圖例位置與樣式（參見下方 Legend）。                                                                                                             |
| Legend   None · 預設 None |                                                                                                                                   |
| Effect                  | Effect(color=None, y=None, label=None, label_position="right", label_offset=4, opacity=None)                                      |
| 新增：v1.8.0               | 設定替代效果或所得效果的顏色、橫軸下方箭頭高度、標籤及透明度。線條本身仍可由                                                                                            |
| 更新：v1.9.0               | substitution_stroke 或 income_stroke 覆寫。                                                                                           |
| Legend                  | Legend(position="auto", fontsize=None, frame=None, columns=None, visible=None, opacity=None)                                      |
| 新增：v1.9.0               | "auto" 選擇遮蔽圖形最少的內側角落；四個角落皆與圖形重疊時，圖例移到右側。也可指定 "upper left" 等內側角落，或 "top"、"bottom"、"left"、"right" 等外側位置。Legend(visible=False) 隱藏圖例。 |

## 例 4

```

from econ_viz import Canvas, DecompositionMethod, Effect, Label, Legend
from econ_viz.models import CobbDouglas
from econ_viz.optimizer import decompose_price_effect

model = CobbDouglas(alpha=0.5, beta=0.5)
result = decompose_price_effect(
    model,
    px=(2.0, 4.0),
    py=3.0,
    income=60.0,
    method=DecompositionMethod.HICKS,
)

(
    Canvas(x_max=25, y_max=25, title="Hicks decomposition")
    .add_decomposition(
        result,
        show_arrows=True,
        label_effects=True,
        show_x_projections=True,
        substitution=Effect(label="SE"),
        income=Effect(label="IE", opacity=0.8),
        curve_label=Label(position="right"),
        legend=Legend(position="bottom"),
    )
    .save("hicks.png")
)

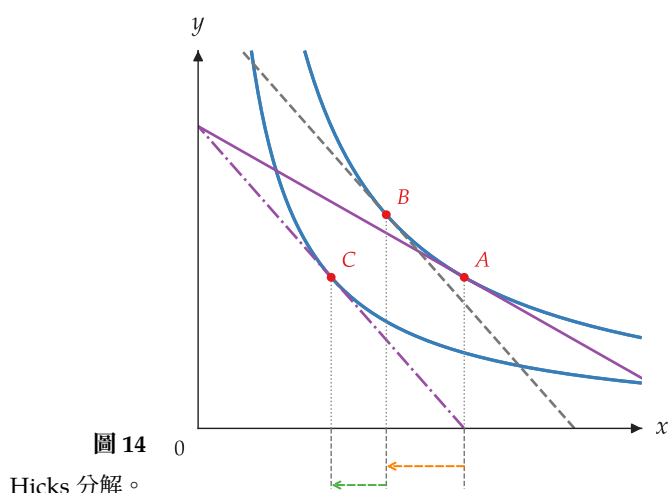
```

## 5.5 Edgeworth 箱形圖

EdgeworthBox EdgeworthBox(<效用函數 A>, <效用函數 B>, total\_x=<浮點數>, total\_y=<浮點數>, ...)

新增：v1.3.0 建立兩人交換經濟的 Edgeworth 箱形圖 (Edgeworth, 1881)。A 的原點在左下，B 的原點在右上。使用 add\_\* 方法加入稟賦、契約曲線、核、價格線與 Walras 均衡。

平滑偏好的契約曲線使用 method="mrs"；有拗折點、角解或自訂分段效用函數時，使用 method="pareto"。



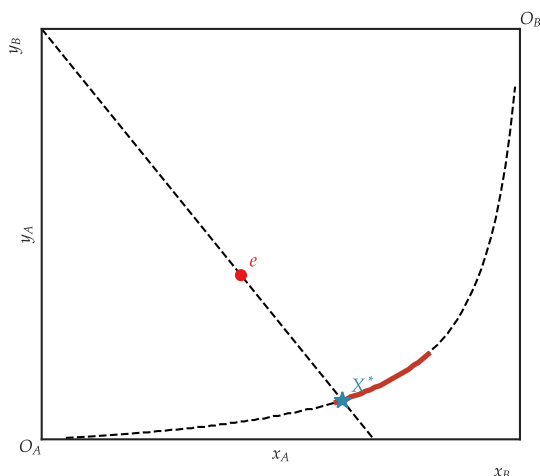


圖 15  
契約曲線與均衡。

### 例 5

```
from econ_viz import EdgeworthBox
from econ_viz.models import CobbDouglas

box = EdgeworthBox(
    CobbDouglas(alpha=0.8, beta=0.2),
    CobbDouglas(alpha=0.2, beta=0.8),
    total_x=12.0,
    total_y=10.0,
    title="Asymmetric Cobb-Douglas",
)

(
    box.add_endowment(5.0, 4.0)
    .add_contract_curve(n=100, method="mrs")
    .add_core()
    .add_price_line(px=1.2, py=1.0)
    .add_walrasian_equilibrium(px=1.2, py=1.0)
    .show_legend(loc="center left", bbox_to_anchor=(1.02, 0.5))
    .save("edgeworth.png")
)
```

## 第 6 節 核心模型

內建模型位於 `econ_viz.models`，可傳入 `solve()` 與 `Canvas.add_utility()`。本章依偏好特性分組，列出各模型的效用函數、參數、特有行為與圖形。

### 6.1 平滑偏好

Cobb-Douglas、CES 與 Translog 的無異曲線為平滑曲線；最適解取決於模型參數、價格與所得。

#### Cobb-Douglas

CobbDouglas CobbDouglas(alpha=0.5, beta=0.5)

Cobb-Douglas 效用函數 (Cobb & Douglas, 1928)：

$$u(x, y) = x^\alpha y^\beta$$

正指數決定兩種商品的相對權重，無異曲線以座標軸為漸近線。

alpha 商品  $x$  的權重。

預設 0.5

beta 商品  $y$  的權重。

預設 0.5

### CES

CES CES(alpha=0.5, beta=0.5, rho=0.5)

CES (Arrow et al., 1961) 透過 rho 調整商品間的替代彈性：

$$u(x, y) = (\alpha x^\rho + \beta y^\rho)^{\frac{1}{\rho}}.$$

替代彈性為  $\sigma = 1/(1 - \rho)$ 。極限形式分別為 Cobb-Douglas ( $\rho \rightarrow 0$ )、完全互補 ( $\rho \rightarrow -\infty$ ) 與完全替代 ( $\rho \rightarrow 1$ )。

alpha 商品  $x$  的權重。

預設 0.5

beta 商品  $y$  的權重。

預設 0.5

rho 替代參數，且  $\rho \neq 1$ 。

預設 0.5

### Translog

Translog Translog(alpha\_0=0.0, alpha\_x=0.5, alpha\_y=0.5, beta\_xx=0.0, beta\_yy=0.0, beta\_xy=0.0)

新增：v1.1.0 Translog (Christensen et al., 1975) 以對數的一次項、二次項與交叉項表示效用：

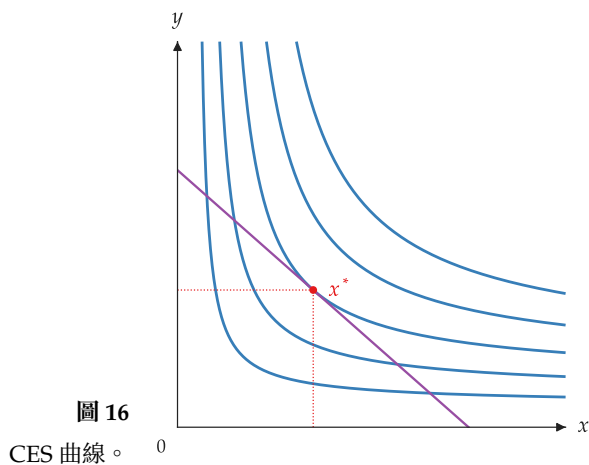
$$\ln u(x, y) = \alpha_0 + \alpha_x \ln x + \alpha_y \ln y$$

$$+ \frac{1}{2} \beta_{xx} (\ln x)^2 + \frac{1}{2} \beta_{yy} (\ln y)^2 + \beta_{xy} \ln x \ln y.$$

二次項與交叉項控制曲率。所有  $\beta$  係數為零時，退化為 Cobb-Douglas 型的對數線性形式。

alpha\_0 對數效用截距。

預設 0.0



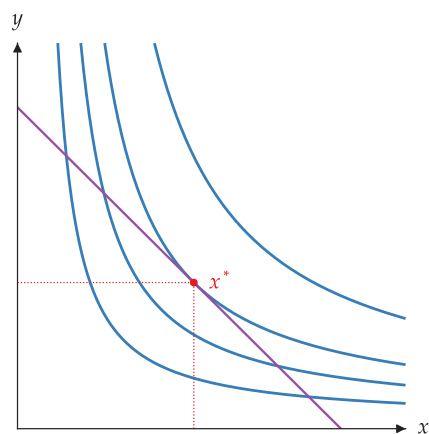


圖 17  
Translog 範例。

`alpha_x`  $\ln x$  的一次項權重。

預設 0.5

`alpha_y`  $\ln y$  的一次項權重。

預設 0.5

`beta_xx`  $x$  方向的曲率。

預設 0.0

`beta_yy`  $y$  方向的曲率。

預設 0.0

`beta_xy`  $x$  與  $y$  的交叉效果。

預設 0.0

## 6.2 拗折點與角解

無異曲線有拗折或為直線時，最適解可能位於拗折點或座標軸，不能僅以相切條件判定。

### 完全互補

`Leontief` `Leontief(a=1.0, b=1.0)`

完全互補效用函數：

$$u(x, y) = \min\{ax, by\}$$

無異曲線呈 L 型。權重與價格為正時，最適點滿足固定比例  $ax = by$ 。

a 商品  $x$  的權重。

預設 1.0

b 商品  $y$  的權重。

預設 1.0

### 完全替代

`PerfectSubstitutes` `PerfectSubstitutes(a=1.0, b=1.0)`

完全替代效用函數：

$$u(x, y) = ax + by.$$

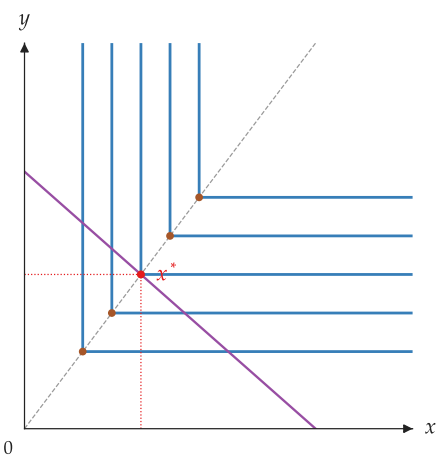


圖 18  
完全互補射線。

無異曲線為直線。每元邊際效用不同時，最適解為僅購買較高者的角解；兩者相等時，預算線上的消費組合皆為最適解。

a 商品  $x$  的邊際效用。

預設 1.0

b 商品  $y$  的邊際效用。

預設 1.0

最大值效用

maximum CustomUtility(func=lambda x, y: np.maximum(a \* x, b \* y))  
以兩個加權數量的最大值定義效用：

$$u(x,y) = \max\{ax,by\}$$

此偏好不具凸性。在正權重與正價格下，最適解位於效用較高的預算線軸截距。此模型以 CustomUtility 建立（詳見第 7 節）。

a 商品  $x$  的權重。

預設 1.0

b 商品  $y$  的權重。

預設 1.0

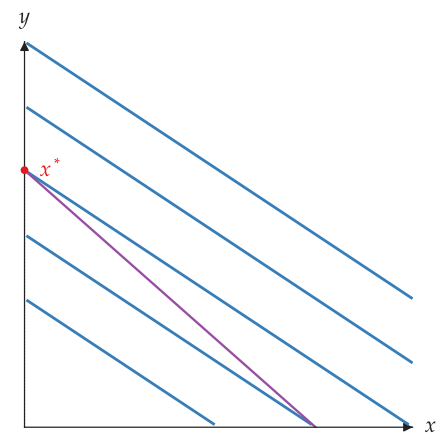


圖 19  
完全替代角解。

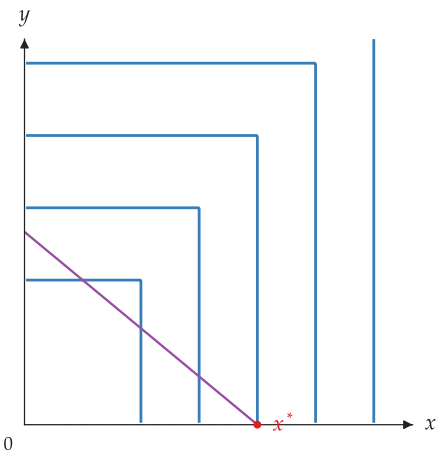


圖 20  
最大值效用。

### 6.3 所得與參考點

以下模型分別描述準線性偏好、最低消費需求與具有極樂點的飽和偏好。

#### 準線性

QuasiLinear `QuasiLinear(v_func=numpy.log, linear_in="y")`  
準線性效用中，一種商品以線性項表示：

$$u(x,y) = f(x) + y.$$

在內部解範圍內，非線性商品的需求不受所得影響。`linear_in` 指定線性項對應的商品。

`v_func` 遞增且凹的函數  $f$ 。  
預設 `numpy.log`  
`linear_in` 以線性方式進入的商品。  
預設 `"y"`

#### Haagsma

Haagsma `Haagsma(alpha_x=1.0, alpha_y=2.0, gamma_x=2.0, gamma_y=27.0)`  
新增：*v1.9.0* `Haagsma` (2012) 效用函數為

$$u(x,y) = \alpha_x \ln(x - \gamma_x) - \alpha_y \ln(\gamma_y - y).$$

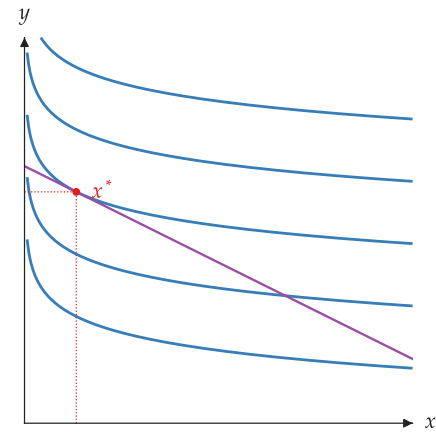


圖 21  
準線性偏好。

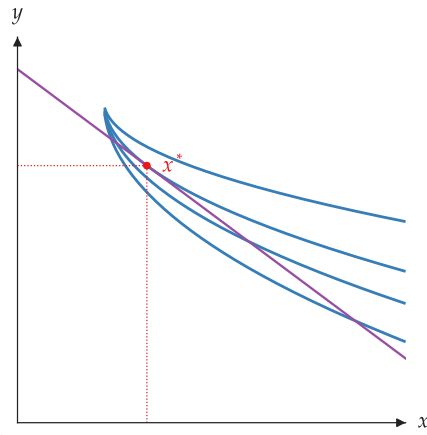


圖 22

$\gamma_y = 10$  的 Haagsma。

定義域為  $x > \gamma_x$  且  $0 \leq y < \gamma_y$ ，並要求  $0 < \alpha_x < \alpha_y$ 。商品  $x$  的 Marshall 需求隨所得下降；當  $\gamma_y p_y < I < \gamma_y p_y + \gamma_x p_x$  時，商品  $x$  為季芬財。

所得達到  $\gamma_y p_y + \gamma_x p_x$  時，效用在預算線上沒有有限最大值，模型會拋出 `Invalid ParameterError`。

`alpha_x` 商品  $x$  的權重，必須小於 `alpha_y`。

預設 1.0

`alpha_y` 商品  $y$  的權重。

預設 2.0

`gamma_x` 商品  $x$  的數量下界。

預設 2.0

`gamma_y` 商品  $y$  的數量上界。

預設 27.0

`demand(px, py, income)` 回傳閉式內部解。

`is_giffen(px, py, income)` 檢查指定預算下商品  $x$  是否為季芬財。

### Stone-Geary

`StoneGeary` `StoneGeary(alpha=0.5, beta=0.5, bar_x=1.0, bar_y=1.0)`

Stone-Geary (Geary, 1950; Stone, 1954) 在 Cobb-Douglas 中加入最低消費量：

$$u(x, y) = (x - \bar{x})^\alpha (y - \bar{y})^\beta.$$

滿足基本需求量  $\bar{x}$  與  $\bar{y}$  後，剩餘所得依權重分配。

`alpha` 超額  $x$  的權重。

預設 0.5

`beta` 超額  $y$  的權重。

預設 0.5

`bar_x` 基本需求量  $\bar{x}$ 。

預設 1.0

`bar_y` 基本需求量  $\bar{y}$ 。

預設 1.0

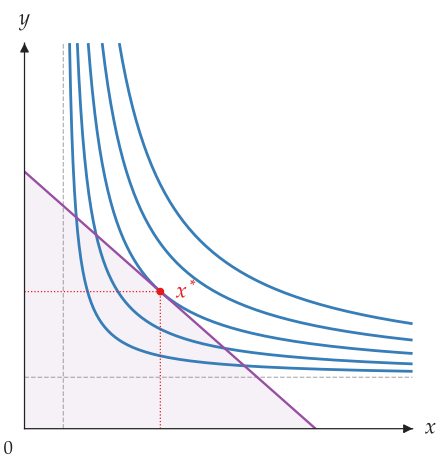


圖 23  
Stone-Geary ◦

飽和偏好

```
Satiation Satiation(bliss_x=5.0, bliss_y=5.0, a=1.0, b=1.0)
```

飽和偏好以  $(x^*, y^*)$  為效用最高的極樂點：

$$u(x, y) = -a(x - x^*)^2 - b(y - y^*)^2.$$

當  $a, b > 0$ ，偏離極樂點會使效用下降，無異曲線為封閉橢圓，偏好不具單調性。

bliss\_x 極樂點座標  $x^*$  ◦

預設 5.0

bliss\_y 極樂點座標  $y^*$  ◦

預設 5.0

a  $x$  軸方向的曲率 ◦

預設 1.0

b  $y$  軸方向的曲率 ◦

預設 1.0

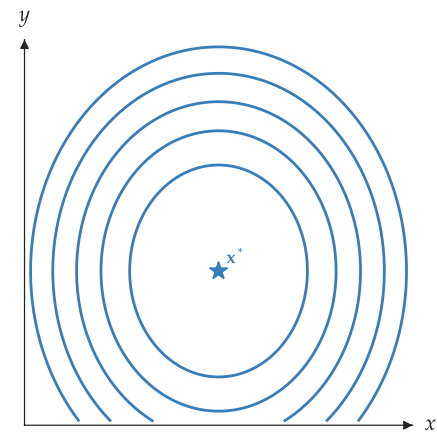


圖 24  
極樂點飽和偏好 ◦

## 第 7 節 進階模型

CustomUtility 定義二變數效用函數。MultiGoodCD 表示多商品 Cobb-Douglas；固定其他商品的數量後，可將模型投影為二維圖。

### 自訂效用函數

CustomUtility CustomUtility(func=<可呼叫物件>, name=<字串>)  
接受向量化的 Python 函數，回傳的模型可用於 solve() 與 Canvas。函數須接受兩個 NumPy 陣列，並回傳相同形狀的效用陣列。以下以對數效用為例：

$$u(x, y) = \ln x + \ln y.$$

func 以  $x$  與  $y$  為變數的向量化效用函數。

name 模型的顯示名稱。

```
import numpy as np
from econ_viz import Canvas, levels, solve
from econ_viz.models import CustomUtility

model = CustomUtility(
    func=lambda x, y: np.log(x) + np.log(y),
    name="log+log",
)
eq = solve(model, px=2.0, py=3.0, income=30.0)

(
    Canvas(x_max=20, y_max=15, title="Custom Utility")
    .add_utility(model, levels=levels.around(eq.utility, n=5))
    .add_budget(2.0, 3.0, 30.0)
    .add_equilibrium(eq)
    .save("custom.png")
)
```

### 多商品 Cobb-Douglas

MultiGoodCD MultiGoodCD(<權重>)

MultiGoodCD.freeze .freeze(<商品>=<數量>, ...)

MultiGoodCD 表示  $N$  種商品的 Cobb-Douglas：

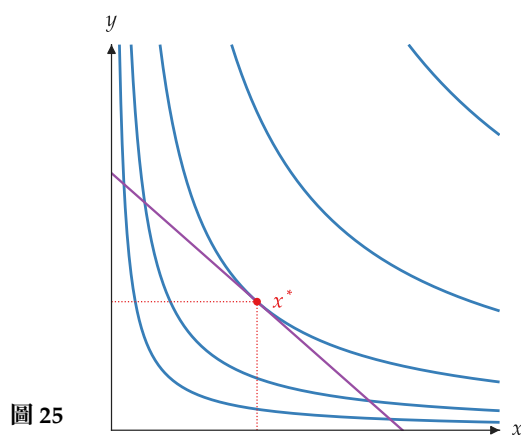


圖 25  
對數自訂模型。

$$u(x_1, \dots, x_N) = \prod_{i=1}^N x_i^{\alpha_i}.$$

使用 `freeze()` 固定  $x$ 、 $y$  以外的商品數量，回傳可用於二維 Canvas 的 `CustomUtility`。

```
shares 商品名稱與指數  $\alpha_i$  的對應。
freeze(...)  $x$ 、 $y$  以外商品的固定數量。

from econ_viz import Canvas, levels, solve
from econ_viz.models import MultiGoodCD

model = MultiGoodCD({"x": 0.3, "y": 0.3, "z": 0.4})
two_good_model = model.freeze(z=10.0)
eq = solve(two_good_model, px=2.0, py=3.0, income=30.0)

(
    Canvas(x_max=20, y_max=15, title="Multi-Good Cobb-Douglas")
    .add_utility(
        two_good_model,
        levels=levels.around(eq.utility, n=5),
    )
    .add_budget(2.0, 3.0, 30.0, fill=True)
    .add_equilibrium(eq)
    .save("multigood.png")
)
```

## 第 8 節 分析

分析 API 提供 Marshall 需求的比較靜態、Slutsky 矩陣，以及效用函數的齊次性與位似性檢查。

### 8.1 比較靜態

`comparative_statics` `comparative_statics(<模型>, px=<浮點數>, py=<浮點數>, income=<浮點數>)`

新增：v1.1.0 以有限差分估計兩商品 Marshall 需求對  $p_x$ 、 $p_y$  與所得的六個偏導數，回傳 `Comparative Statics`。

- 在 `solve(...)` 的結果附近使用中央有限差分，預設的相對步長是  $1e-3$ 。

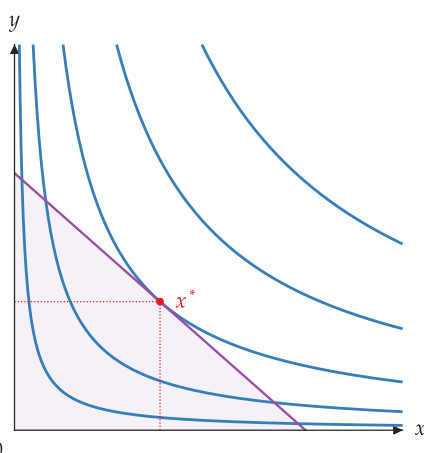


圖 26

三商品固定  $z = 10$ 。

- 偵測到自身價格導數為正或所得導數為負時會發出警告，分別對應季芬財與劣等財的需求特性。

```

dx_dpx   $\partial x^* / \partial p_x$  °
    float
dx_dpy   $\partial x^* / \partial p_y$  °
    float
dx_dI    $\partial x^* / \partial I$  °
    float
dy_dpx   $\partial y^* / \partial p_x$  °
    float
dy_dpy   $\partial y^* / \partial p_y$  °
    float
dy_dI    $\partial y^* / \partial I$  °
    float

from econ_viz.models import CobbDouglas
from econ_viz.optimizer import comparative_statics

model = CobbDouglas(alpha=0.4, beta=0.6)
cs = comparative_statics(model, px=2.0, py=3.0, income=60.0)

print(round(cs.dx_dpx, 1), round(cs.dx_dpy, 1), round(cs.dx_dI, 1))
print(round(cs.dy_dpx, 1), round(cs.dy_dpy, 1), round(cs.dy_dI, 1))

# -6.0 0.0 0.2
# 0.0 -4.0 0.2

```

## 8.2 Slutsky 矩陣

`slutsky_matrix` `slutsky_matrix(<模型>, px=<浮點數>, py=<浮點數>, income=<浮點數>)`  
 更新：v1.2.3 以 Marshall 需求導數與所得效果計算兩商品的 Slutsky 替代矩陣，並檢查對稱性、負半定性與齊次性；檢查未通過時發出警告。使用 `as_array()` 可轉為 NumPy 陣列。

```

from econ_viz import slutsky_matrix
from econ_viz.models import CobbDouglas

S = slutsky_matrix(
    CobbDouglas(alpha=0.4, beta=0.6),
    px=2.0, py=3.0, income=60.0,
)

print(round(S.s_xx, 1), round(S.s_xy, 1))
print(round(S.s_yx, 1), round(S.s_yy, 1))
print(S.as_array().round(1))

# -3.6 2.4
# 2.4 -1.6
# [[-3.6  2.4]
#  [ 2.4 -1.6]]

```

## 8.3 齊次性

`HomogeneityAnalyzer` `HomogeneityAnalyzer(<模型>)`  
 新增：v1.1.0 檢查效用函數的齊次性、位似性與需求的零次齊次性。

```

    degree() 估計齊次次數，詳見第 8.4 節。
    euler_check(x, y) 計算指定消費組合的 Euler 定理殘差。
    is_homothetic() 邊際替代率 (MRS) 在等比例縮放下是否不變。
    demand_degree_zero(px, py, income) Marshall 需求是否為零次齊次。

from econ_viz.analysis import HomogeneityAnalyzer
from econ_viz.models import CobbDouglas

analyzer = HomogeneityAnalyzer(CobbDouglas(alpha=0.4, beta=0.6))
result = analyzer.degree()

print(round(result.degree, 6))
print(result.returns_to_scale)
print(round(analyzer.euler_check(3.0, 4.0), 6))
print(analyzer.is_homothetic())
print(analyzer.demand_degree_zero(px=2.0, py=3.0, income=60.0))

# 1.0
# ReturnsToScale.CONSTANT
# 0.0
# True
# True

```

## 8.4 規模報酬

`degree()` 回傳 `HomogeneityResult`，包含下列欄位：

```

degree 估計的齊次次數。
    float
is_homogeneous 是否為齊次函數。
    bool
returns_to_scale 規模報酬的分類，取值如下。
    ReturnsToScale

```

Cobb-Douglas 的齊次次數為  $\alpha + \beta$ ：

$$u(\lambda x, \lambda y) = \lambda^{\alpha+\beta} u(x, y).$$

```

INCREASING  $\alpha + \beta > 1$ ，規模報酬遞增。
CONSTANT  $\alpha + \beta = 1$ ，規模報酬固定。
DECREASING  $\alpha + \beta < 1$ ，規模報酬遞減。
NOT_HOMOGENEOUS 沒有一致的齊次次數。

```

```

def shifted_utility(x, y):
    return x**0.4 * y**0.6 + 1.0

models = [
    CobbDouglas(alpha=0.7, beta=0.6),
    CobbDouglas(alpha=0.4, beta=0.6),
    CobbDouglas(alpha=0.2, beta=0.5),
    shifted_utility,
]

for model in models:
    result = HomogeneityAnalyzer(model).degree()
    degree = None if result.degree is None else round(result.degree, 1)
    print(degree, result.returns_to_scale.name)

# 1.3 INCREASING

```

```
# 1.0 CONSTANT
# 0.7 DECREASING
# None NOT_HOMOGENEOUS
```

## 第 9 節 主題與設定

Theme 定義圖形的預設外觀；Config 從 TOML 檔案載入主題覆寫與字型設定。Canvas、Figure、DemandDiagram、價格效果分解及 EdgeworthBox 共用這些樣式物件。

### 9.1 內建主題

```
themes from econ_viz import Canvas, themes
cvs = Canvas(theme=themes.paper)
```

更新：v1.11.0 Python API 提供下列主題。主題物件不可變；需要調整欄位時，可建立新的 Theme，或以 Config 覆寫指定欄位。

表 4  
內建主題

| 名稱           | 特性                |
|--------------|-------------------|
| default      | 色盲友善的預設配色         |
| colorblind   | 與 default 相同的配色別名 |
| nord         | 採用 Nord 色盤的冷色系主題  |
| paper        | 透明背景與出版用細線條       |
| monochrome   | 以線型與標記區分圖層的黑白主題   |
| presentation | 放大標籤、線條與標記，供投影使用  |
| dark         | 深色背景與相應的前景色       |

Python API 與設定檔的 base 可使用全部七個名稱<sup>5</sup>。

預設主題的顏色取自一組色盲友善的配色 (thriveth, 2014)。

themes.COLORBLIND\_CYCLE\_HEX 與 themes.COLORBLIND\_CYCLE\_RGB 提供這組配色的十六進位與 RGB 值。經濟學教學大量依賴圖形，因此不應只靠顏色傳達意義，以免視覺障礙的學生無法辨識 (Kugler & Andrews, 1996)。

### 9.2 自訂主題

```
Theme from econ_viz import Theme

my_theme = Theme(
    name="custom",
    background_color="white",
    label_scale=1.0,
    axis_color="#333333",
    label_color="#333333",
    ic_color="#2563eb",
    secondary_ic_color="#93c5fd",
    secondary_ic_opacity=0.45,
```

<sup>5</sup> 命令列直接傳入 --theme 時，只接受 colorblind 別名以外的六個主題。

```

        budget_color="#dc2626",
        eq_color="#16a34a",
    )

```

更新：v1.12.0 Theme 保存圖形元素的顏色、線寬與樣式預設值。未指定的欄位使用類別預設值；繪圖方法明確收到的樣式參數優先於主題。

name 主題識別名稱。

str

background\_color 圖形與座標區域的背景色；None 保持透明。

str | None · 預設 None

label\_scale 一般標籤字級的倍率。

float · 預設 1.0

axis\_color 座標軸顏色。

str · 預設 "#222222"

label\_color 座標標籤與原點文字顏色。

str · 預設 "#222222"

ic\_color 無異曲線顏色。

str · 預設 "#377EB8"

ic\_linewidth 無異曲線線寬。

float · 預設 1.8

secondary\_ic\_color 非焦點無異曲線顏色；None 沿用 ic\_color。

str | None · 預設 None

secondary\_ic\_linewidth 非焦點無異曲線線寬。

float · 預設 1.0

secondary\_ic\_opacity 非焦點無異曲線透明度。

float · 預設 0.45

path\_color PCC 與 ICC 路徑顏色。

str · 預設 "#4DAF4A"

budget\_color 預算線顏色。

str · 預設 "#984EA3"

budget\_fill\_alpha 預算集合陰影透明度。

float · 預設 0.08

eq\_color 均衡點與投影線顏色。

str · 預設 "#E41A1C"

eq\_markersize 均衡點標記大小。

float · 預設 4.0

Theme 也包含射線、拗折點、補償預算線與價格效果箭頭的顏色和線寬。axis\_stroke、drop\_stroke、projection\_stroke、guide\_stroke 與 box\_stroke 保存其他線條的預設值；標記、標籤、填色及圖例則由 Marker、Label、Fill 與 Legend 屬性提供（詳見第 4.3 節）。

### 9.3 設定檔

```
Config.load    from econ_viz import Config
Config.use     Config.load("econ-viz.toml").use()
Config.reset   # 此後建立的圖形使用該設定
               Config.reset()
               # 恢復內建預設值
```

新增：v1.10.0 Config.load() 讀取 TOML 檔案並回傳 Config。use() 將其設為後續圖形的預設設定；reset() 恢復內建預設值。

#### 教學：以設定檔統一圖形樣式

同一份講義或論文的圖形，通常要共用相同的顏色與線條。與其在每個 Canvas 重複傳入參數，不如把樣式寫在專案根目錄的 econ-viz.toml，讓 Python 與命令列讀取同一份設定。

1. **建立範本**。在專案根目錄執行下列指令，會產生含註解的 econ-viz.toml，列出每個區段可用的名稱與欄位：

```
econ-viz init
```

範本中的設定預設都是註解；只保留需要修改的項目即可，其餘可以刪除。

2. **選擇基底主題**。base 指定起始的內建主題（表 4），其餘設定都在它之上覆寫：

```
base = "paper"
```

3. **覆寫顏色與樣式**。只寫出要改的欄位：

```
[color]
ic = "#1B4F72"
eq = "#C0392B"

[stroke.budget]
width = 1.6
style = "dashed"

[marker.eq]
size = 5

[label.point]
fontsize = 11
position = "right"
```

區段名稱對應主題屬性：[color] 的 ic 對應 ic\_color，[stroke.budget] 對應 theme.budget\_stroke；區段內的欄位就是 Stroke、Marker、Label、Fill、Legend 的參數（詳見第 4.3 節）。字型寫在 [font] 的 text 與 math。省略的欄位保留基底主題的值。

4. **在 Python 中套用**。在建立任何圖形之前載入一次：

```
from econ_viz import Config

Config.load().use() # 預設讀取目前目錄的 econ-viz.toml
```

此後建立的 Canvas、Figure、DemandDiagram 與 EdgeworthBox 都使用這組設定；呼叫 Config.reset() 可恢復內建預設值。

5. **在命令列套用**。以 --config 指定設定檔：

```
econ-viz plot --config econ-viz.toml --model cobb-douglas \
--px 2 --py 3 --income 30 --output figure.png
```

設定的優先順序由高到低為：直接傳入 Canvas 或繪圖方法的參數、設定檔、基底主題。

例如 `Canvas(theme=themes.default)` 會略過設定檔中的主題。

設定檔寫錯時，`Config.load()` 會拋出 `InvalidParameterError`，並列出可用的名稱。例如把 `[stroke.budget]` 誤寫成 `[stroke.budgt]`：

```
econ-viz.toml: [stroke.budgt]: no such setting (choose: axis, box,
budget, ...)
```

## 第 10 節 匯出

### 10.1 輸出格式

`Canvas.save` `cvs.save(<路徑>)`

依 `path` 的副檔名決定輸出格式。`Figure`、`DemandDiagram` 與 `EdgeworthBox` 使用相同介面。

- PNG：點陣圖，解析度由畫布的 `dpi` 設定，預設為 300。
- PDF、SVG：向量圖，可在列印或縮放時保留線條品質。
- TeX：TikZ 原始碼（.tex），可嵌入 L<sup>A</sup>T<sub>E</sub>X 文件。透過關鍵字參數 `tikz_scale` 與 `tikz_standalone` 調整輸出。

```
cvs.save("figure.png")    # PNG，依畫布 DPI（預設 300）
cvs.save("figure.pdf")    # PDF（向量）
cvs.save("figure.svg")    # SVG（向量）
cvs.save("figure.tex")    # TikZ
```

`dpi` 僅影響點陣圖。預覽時可降低解析度：

```
cvs = Canvas(x_max=20, y_max=15, dpi=150)  # 降低 DPI，加快預覽
```

### 10.2 GIF 動畫

使用 `Animator.save()` 匯出 GIF，完整範例詳見第 12 節。

```
from econ_viz.animation import Animator

Animator(draw_frame, frames=frames).save("animation.gif", fps=12,
dpi=120)
```

### 10.3 互動視窗

`cvs.show()` 開啟 `matplotlib` 互動視窗，不會儲存檔案。命令列的 `plot` 省略 `--output` 時也會開啟視窗（詳見第 11.2 節）：

```
econ-viz plot --model cobb-douglas --px 2 --py 3 --income 30
```

## 第 11 節 命令列工具

命令列工具可繪圖、批次處理及輸出需求公式。以 `uv tool install econ-viz` 安裝為獨立工具，或在 `uv` 專案中於指令前加上 `uv run`（詳見第 2.5 節）。

表 5  
命令列指令

| 指令                     | 說明                         |
|------------------------|----------------------------|
| econ-viz help [⟨指令⟩]   | 顯示命令列工具或特定指令的說明            |
| econ-viz models        | 列出所有支援的效用模型                |
| econ-viz plot ...      | 產生並匯出圖形                    |
| econ-viz solve-tex ... | 以 TeX 格式輸出 Marshall 需求的閉式解 |
| econ-viz init [path]   | 建立 econ-viz.toml 設定範本      |

11.1 說明與模型

econ-viz help econ-viz help [⟨指令⟩]  
未指定參數時列出所有指令；指定指令名稱時顯示該指令的選項。

```
econ-viz help      # 所有指令
econ-viz help plot # plot 選項
econ-viz help models # models 選項
```

econ-viz models 列出命令列介面支援的模型名稱與參數。--model 使用 kebab-case 名稱，與 Python 類別名稱不同（參見表 6）。參數選項詳見第 11.2 節。

表 6  
命令列模型

| 模型           | --model 值           | 必要參數            | 選用參數                                            |
|--------------|---------------------|-----------------|-------------------------------------------------|
| Cobb-Douglas | cobb-douglas        | alpha、beta      | —                                               |
| CES          | ces                 | rho             | alpha、beta                                      |
| 完全互補         | leontief            | a、b             | —                                               |
| 完全替代         | perfect-substitutes | a、b             | —                                               |
| 飽和偏好         | satiation           | bliss-x、bliss-y | a、b                                             |
| 準線性          | quasi-linear        | —               | v-func、linear-in                                |
| Stone-Geary  | stone-geary         | —               | alpha、beta、bar-x、bar-y                          |
| Translog     | translog            | —               | alpha-0、alpha-x、alpha-y、beta-xx、beta-yy、beta-xy |

11.2 繪圖

econ-viz plot econ-viz plot --model ⟨名稱⟩ ⟨選項⟩  
econ-viz plot --latex ⟨算式⟩ ⟨選項⟩  
使用 --model 指定模型，或使用 --latex 輸入效用函數，兩者擇一。指定 --output 時匯出檔案，省略時開啟互動視窗。

預算線與均衡點需要完整的價格與所得。--no-budget、--no-equilibrium、--no-curves 可分別省略對應圖層，且可併用（參見表 7）。

表 7  
圖層控制選項

| 情況                                                                 | 結果                   |
|--------------------------------------------------------------------|----------------------|
| 提供完整 <code>--px</code> 、 <code>--py</code> 、 <code>--income</code> | 繪製預算線並求解均衡           |
| 省略任一價格或所得                                                          | 僅繪製無異曲線，不畫預算線與均衡點    |
| 加上 <code>--no-equilibrium</code>                                   | 略過求解與均衡標示，即使提供了價格與所得 |
| 加上 <code>--no-budget</code>                                        | 隱藏預算線，即使提供了價格與所得     |
| <code>--no-curves</code> 搭配完整價格與所得                                 | 只呈現預算線與均衡點，不畫無異曲線    |

```
# 指定模型名稱
econ-viz plot --model cobb-douglas --alpha 0.5 --beta 0.5 ...

# LaTeX 算式
econ-viz plot --latex "x^{0.4} y^{0.6}" ...
```

## 例 6

```
# Cobb-Douglas，可行集合加陰影
econ-viz plot --model cobb-douglas --alpha 0.5 --beta 0.5 \
--px 2 --py 3 --income 30 \
--fill --output cobb_douglas.png

# LaTeX 輸入，Nord 主題，擴張路徑
econ-viz plot --latex "x^{0.4} y^{0.6}" \
--px 2 --py 3 --income 30 \
--theme nord --show-ray \
--output cd_latex.png

# 完全互補，加大畫布
econ-viz plot --model leontief --a 1 --b 2 \
--px 2 --py 3 --income 30 \
--x-max 20 --y-max 15 \
--output leontief.png

# CES，只畫無異曲線
econ-viz plot --model ces --rho -0.5 \
--x-max 20 --y-max 15 --n-curves 6 \
--no-budget --no-equilibrium \
--output ces.png

# 飽和（極樂點）
econ-viz plot --model satiation --bliss-x 6 --bliss-y 4 \
--x-max 12 --y-max 10 \
--no-budget --no-equilibrium \
--output satiation.png

# 不加 --output：開啟視窗
econ-viz plot --model cobb-douglas --px 2 --py 3 --income 30
```

## 選擇模型

`--model`, `-m` 模型名稱，完整清單可用 `econ-viz models` 查詢。  
`--latex`, `-l`  $\text{\LaTeX}$  算式，支援 Cobb-Douglas、完全互補、完全替代與 CES（詳見第 14 節）。

## 價格與所得

`--px` 商品  $x$  的價格。  
`--py` 商品  $y$  的價格。  
`--income` 消費者的所得。

## 模型參數

`--alpha` `alpha` 參數（Cobb-Douglas、CES）。  
預設 0.5

```

--beta beta 參數 (Cobb-Douglas、CES) 。
    預設 0.5
    --a 參數  $a$  (完全互補、完全替代、飽和) 。
    預設 1.0
    --b 參數  $b$  (完全互補、完全替代、飽和) 。
    預設 1.0
    --rho 替代參數 (CES) 。
    預設 0.5
--bliss-x 極樂點的  $x$  座標 (飽和) 。
    預設 5.0
--bliss-y 極樂點的  $y$  座標 (飽和) 。
    預設 5.0
    準線性、Stone-Geary 與 Translog 另有下列選項，可依模型使用：
--v-func 準線性的非線性函數，接受 log 或 sqrt 。
    預設 log
--linear-in 準線性的線性商品，接受 x 或 y 。
    預設 y
--bar-x Stone-Geary 的最低消費量  $\bar{x}$  。
    預設 1.0
--bar-y Stone-Geary 的最低消費量  $\bar{y}$  。
    預設 1.0
--alpha-0 Translog 的截距。
    預設 0.0
--alpha-x Translog 中  $\ln x$  的一次項權重。
    預設 0.5
--alpha-y Translog 中  $\ln y$  的一次項權重。
    預設 0.5
--beta-xx Translog 中  $x$  方向的曲率。
    預設 0.0
--beta-yy Translog 中  $y$  方向的曲率。
    預設 0.0
--beta-xy Translog 中  $x$  與  $y$  的交叉項。
    預設 0.0

econ-viz plot --model stone-geary --bar-x 2 --bar-y 2 \
              --px 2 --py 3 --income 30 \
              --x-max 20 --y-max 15 --output stone_geary.png

```

此例的最低消費支出為  $2 \times 2 + 3 \times 2 = 10$ ，所得 30 高於此限制。若將所得改為 10 或更低，求解會因不符合模型定義域而失敗。

### 畫布與圖層

```

--x-max 橫軸上限。
    預設 10
--y-max 縱軸上限。
    預設 10

```

`--x-label` 橫軸標籤。  
 預設 `x`  
`--y-label` 縱軸標籤。  
 預設 `y`  
`--title` 圖形標題。  
`--theme` 主題名稱：`default`、`nord`、`paper`、`monochrome`、`presentation` 或 `dark`（詳見第 9 節）。  
 預設 `default`  
`--config` TOML 設定檔路徑。命令列參數的優先順序高於設定檔。  
`--n-curves` 無異曲線的條數。  
 預設 `5`  
`--dpi` 點陣圖輸出解析度。  
 預設 `300`  
`--fill` 為預算線下方的可行集合加上陰影。  
`--show-ray` 畫出通過最適點的擴張路徑射線。  
`--no-budget` 不畫預算線。  
`--no-equilibrium` 不畫均衡點。  
`--no-curves` 不畫無異曲線。  
`--output, -o` 輸出檔案（`.png`、`.pdf`、`.svg`、`.tex`）；省略時開啟互動視窗。  
 批次產圖時，為每組參數指定不同檔名，並先建立輸出目錄。`--x-max` 與 `--y-max` 不會隨價格自動調整；軸截距或最適點超出範圍時，應同步增大畫布上限。

### 11.3 建立設定檔

`econ-viz init` `econ-viz init` [`<路徑>`] [`--force`]  
 新增：`v1.10.0` 建立含註解的 `econ-viz.toml` 範本。省略路徑時寫入目前目錄；檔案已存在時，必須使用 `--force` 才會覆寫。

```
econ-viz init
econ-viz init config/figures.toml
econ-viz plot --config econ-viz.toml --model cobb-douglas \
  --px 2 --py 3 --income 30 --output figure.png
```

### 11.4 需求公式

`econ-viz solve-tex` `econ-viz solve-tex` `--model` `<名稱>` `<選項>`  
 以 TeX 格式輸出 Marshall 需求的閉式解，可用於支援 TeX 數學式的文件。

```
# 數值參數
econ-viz solve-tex --model cobb-douglas --alpha 0.4 --beta 0.6

# 符號參數
econ-viz solve-tex --model cobb-douglas --symbolic-params

# 自訂價格與所得符號
econ-viz solve-tex --model leontief --a 2 --b 3 \
  --px-symbol p_1 --py-symbol p_2 --income-symbol M
```

支援 `cobb-douglas`、`leontief`、`perfect-substitutes` 及其  $\text{\LaTeX}$  表示式。此指令只輸出公式文字，不產生圖檔。

`--symbolic-params` 保留模型參數符號，例如  $\alpha$ 、 $\beta$ ；未使用時代入指定的參數值。

```
--px-symbol 商品  $x$  價格的符號。
           預設  $p_x$ 
--py-symbol 商品  $y$  價格的符號。
           預設  $p_y$ 
--income-symbol 所得的符號。
           預設  $I$ 
```

## 第 12 節 動畫

Animator 依指定的參數序列呼叫繪圖函數，將回傳的 Canvas 或 Figure 匯出為 GIF。使用前需安裝 animation 選用依賴（詳見第 2.4 節）。

Animator Animator(<繪圖函式>, frames=<數值>).save(<路徑>, fps=12, dpi=120, loop=0)  
 新增：v1.4.0 依序將 frames 的值傳入 draw，每次回傳的圖形對應一個影格。

- save() 使用 Pillow，不需要 ffmpeg。
- 匯出前將影格合成至白色背景，避免殘影。
- fps、dpi 與 loop 分別設定影格率、解析度與循環次數。

### 例 7

```
import numpy as np

from econ_viz import Canvas, levels, solve
from econ_viz.animation import Animator
from econ_viz.models import CobbDouglas

def draw(px: float) -> Canvas:
    model = CobbDouglas(alpha=0.5, beta=0.5)
    eq = solve(model, px=px, py=2.0, income=20.0)
    lvls = levels.around(eq.utility, n=5)

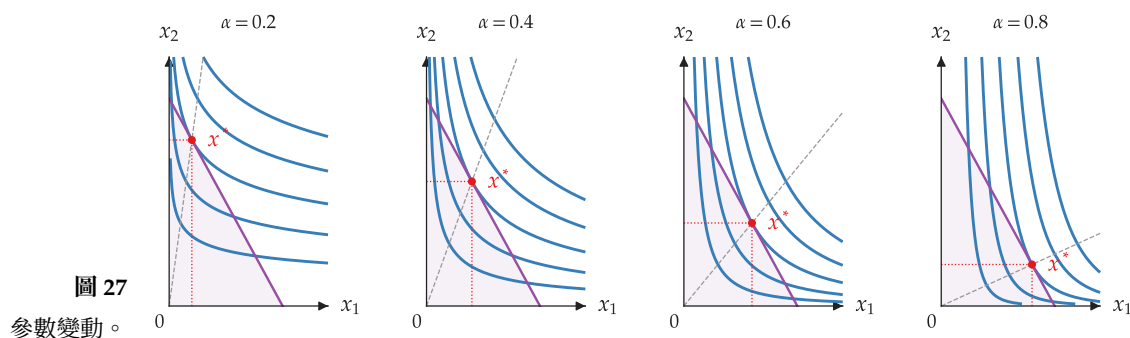
    return (
        Canvas(
            x_max=14, y_max=12,
            x_label="X_1", y_label="X_2",
            title="Price sweep"
        )
        .add_utility(model, levels=lvls)
        .add_budget(px=px, py=2.0, income=20.0, fill=True)
        .add_equilibrium(eq, show_ray=True, drop_dashes=True)
    )

Animator(draw, frames=np.linspace(1.0, 6.0, 45)).save(
    "price_sweep.gif",
    fps=12,
    dpi=120,
)
```

examples/scripts/animation.py 提供四種範例<sup>6</sup>：

- 參數變動：固定價格與所得，調整效用函數參數。
- 價格變動：固定效用函數與  $p_y$ ，調整  $p_x$ ，呈現預算線旋轉，寫法見上例。
- 所得變動：固定效用函數與價格，調整所得，呈現預算線平移。
- 預算線變動：僅顯示預算線，省略效用圖層。

<sup>6</sup>各範例均包含 Cobb-Douglas、CES、完全替代與完全互補模型。



四種範例的差異只在於 `draw()` 每次固定哪些值、調整哪個值。以參數變動為例，將上例的 `draw()` 改為接收 `alpha`，價格固定為  $p_x = 2$ ：

```
def draw(alpha: float) -> Canvas:
    model = CobbDouglas(alpha=alpha, beta=1.0 - alpha)
    eq = solve(model, px=2.0, py=2.0, income=20.0)
    ...

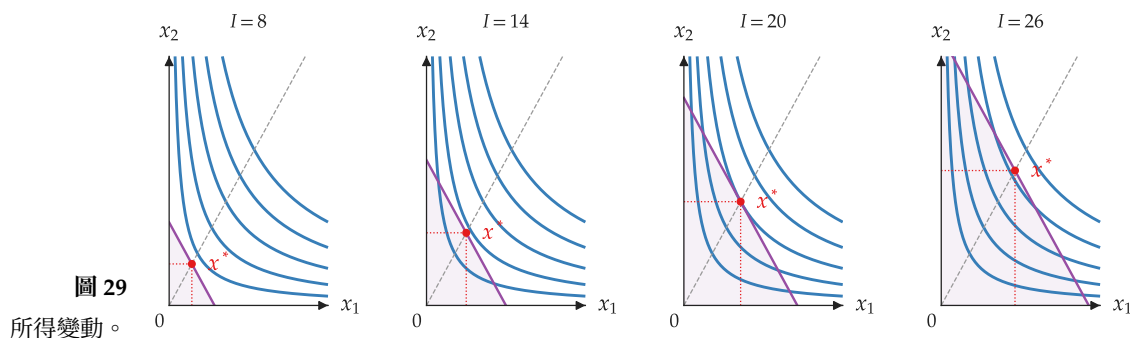
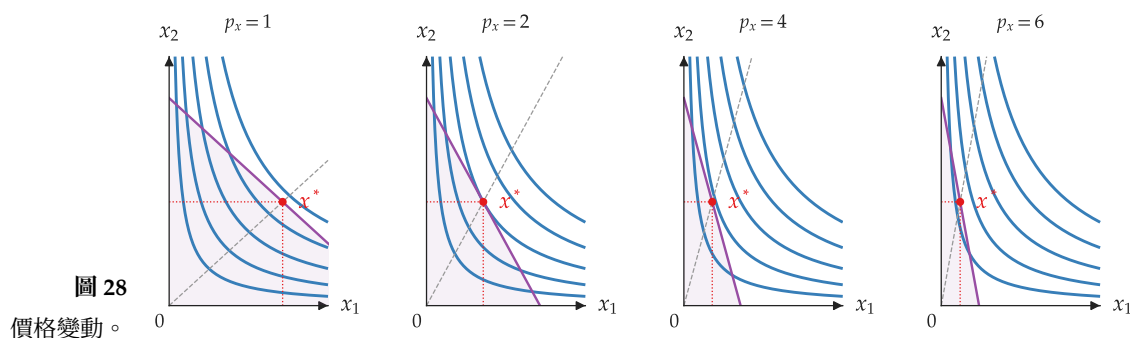
Animator(draw, frames=np.linspace(0.1, 0.9,
45)).save("parameter_sweep.gif")
```

所得變動將 `draw()` 的引數改為 `income`，並把該值傳入 `add_budget()` 與 `solve()`。只呈現預算線時，省略 `add_utility()` 與 `add_equilibrium()`。

圖 27 至圖 29 各節錄四個影格。

## 第 13 節 互動元件

`WidgetViewer` 在 Jupyter 中提供滑桿與數值輸入欄位，調整參數後會更新圖形。使用前需安裝 `interactive` 選用依賴（詳見第 2.4 節）。



WidgetViewer WidgetViewer((繪圖函式), (名稱)=((下限), (上限), (間距)), ...).show()

新增: v1.4.0 傳入繪圖函數 draw，並以 (最小值, 最大值, 步長) 指定各參數範圍。每個參數會產生同步的 FloatSlider 與 FloatText；數值變更後重新呼叫 draw，並取代儲存格中的舊圖。draw 的寫法與 Animator 相同（詳見第 12 節），只是可同時接收多個參數。以第 12 節的價格變動範例為基礎，讓 draw() 另外接收 alpha：

### 例 8

```
from econ_viz.interactive import WidgetViewer

def draw(alpha: float, px: float) -> Canvas:
    model = CobbDouglas(alpha=alpha, beta=1.0 - alpha)
    ... # 其餘與動畫範例相同

WidgetViewer(
    draw,
    alpha=(0.2, 0.8, 0.05),
    px=(1.0, 6.0, 0.25),
).show()
```

## 13.1 筆記本設定

在新的 Jupyter 或 Colab 環境中，依下列順序執行<sup>7</sup>：

1. 執行安裝儲存格。
2. 若安裝過程升級 ipywidgets、traitlets 或 IPython，重新啟動執行環境。
3. 重新啟動後，從匯入套件的儲存格繼續執行，無須再次安裝。

## 13.2 選擇互動元件或 GIF

需要由使用者調整參數時，使用 WidgetViewer；需要在簡報或網頁中播放固定的變動過程時，使用 Animator 匯出 GIF（詳見第 12 節）。

# 第 14 節 L<sup>A</sup>T<sub>E</sub>X 解析

parse\_latex parse\_latex((算式))

將 L<sup>A</sup>T<sub>E</sub>X 效用函數解析為模型，回傳值可用於 solve()、Canvas 或其他接受模型的 API。

```
from econ_viz import parse_latex

model = parse_latex(r"x^{0.4} y^{0.6}") # CobbDouglas(alpha=0.4,
beta=0.6)
model = parse_latex(r"\min(2x, 3y)") # Leontief(a=2.0, b=3.0)
model = parse_latex(r"2x + 3y") # PerfectSubstitutes(a=2.0,
b=3.0)
```

## 14.1 支援的形式

支援的輸入形式參見表 8。省略的係數或指數預設為 1。

<sup>7</sup> econ-viz repo 中的 notebook/econ-viz Playground.ipynb 已採用此流程，可在 Jupyter、VS Code 或 Colab 開啟；環境中已有 econ-viz 時，會自動略過安裝。

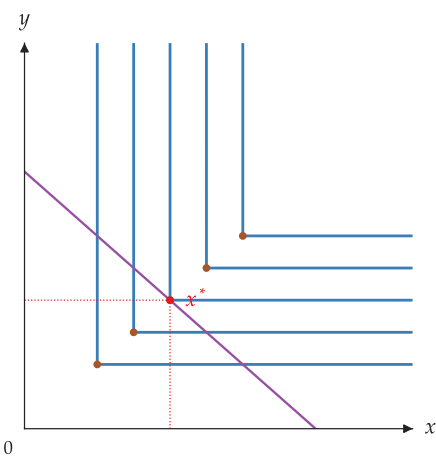


圖 30  
Leontief 範例。

表 8  
支援的效用函數。

| 類型           | 格式                                              | 範例                                 |
|--------------|-------------------------------------------------|------------------------------------|
| Cobb-Douglas | $x^{\alpha} y^{\beta}$ 或 $x^{\alpha} y^{\beta}$ | $x^{0.3} y^{0.7}$                  |
| 完全互補         | $\min(ax, by)$ 或 $\min(ax, by)$                 | $\min(2x, y)$                      |
| 完全替代         | $ax + by$                                       | $3x + 1.5y$                        |
| CES          | $(\alpha x^{\rho} + \beta y^{\rho})^{1/\rho}$   | $(0.4x^{-0.5} + 0.6y^{-0.5})^{-2}$ |

可省略開頭的  $U =$  或  $u(x, y) = :$

$U(x,y) = x^{0.5} y^{0.5}$   
 $U = x^{0.5} y^{0.5}$   
 $x^{0.5} y^{0.5}$

14.2 錯誤處理

無法解析時會拋出 `econ_viz.exceptions.ParseError`，錯誤訊息包含支援的格式：

```
from econ_viz import parse_latex
from econ_viz.exceptions import ParseError

try:
    model = parse_latex(r"x^2 + y^2")
except ParseError as e:
    print(e)
```

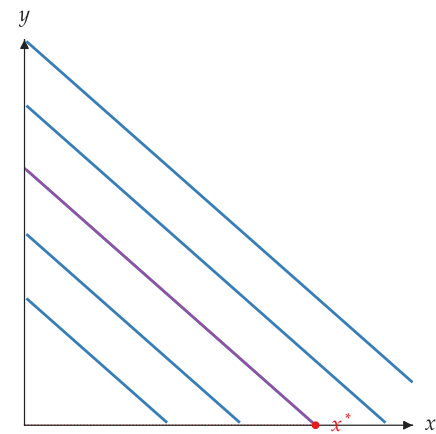


圖 31  
完全替代輸入範例。

```
# Unrecognised LaTeX utility function: 'x^2 + y^2'
# Supported forms:
#   Cobb-Douglas      : x^{\alpha} y^{\beta}
#   Leontief          : \min(ax, by)
#   Perfect Substitutes: ax + by
#   CES               : (\alpha x^{\rho} + \beta y^{\rho})^{1/\rho}
```

### 14.3 在命令列工具中使用

econ-viz plot --latex 使用相同的解析器，接受相同的輸入格式（詳見第 11.2 節）：

```
econ-viz plot --latex "x^{0.4} y^{0.6}" --px 2 --py 3 --income 30 -o
out.png
```

## 更新紀錄

本章列出影響套件使用的版本變更，不含僅修改文件的版本；各條目依 l3doc 慣例標註於它所描述的功能旁，頁碼即該功能的實際頁碼。

|                                                  |              |                                                              |              |
|--------------------------------------------------|--------------|--------------------------------------------------------------|--------------|
| <b>v1.12.0</b>                                   | (2026-09-26) | EdgeworthBox：未指定主題與字型時，使用目前的 Config                          | 19           |
| 一般：新增無異曲線主次層級、焦點曲線及數值或序數標籤                       | 2            | Config：新增 Config、econ-viz.toml、econ-viz init 與 plot --config | 33           |
| Canvas.add_utility：新增焦點效用層級、次要曲線樣式及序數標籤          | 7            | econ-viz init：新增設定檔範本指令與 plot --config                       | 34           |
| Theme：新增次要無異曲線的顏色、線寬與透明度設定                       | 31           | <b>v1.9.0</b>                                                | (2026-09-26) |
| <b>v1.11.0</b>                                   | (2026-09-26) | Canvas：座標軸、原點、標題與其他文字元素接受 Label                              | 7            |
| 一般：完成主題系統，統一畫布、多面板圖、需求圖、效果分解與 Edgeworth 箱形圖的視覺設定 | 2            | Stroke：新增 opacity 透明度                                        | 12           |
| Canvas：背景、標籤、線條與標記均由主題提供預設值                      | 7            | DemandDiagram：支援 Legend、Marker、Label 與各圖層的 Stroke            | 16           |
| Figure：面板背景、標籤、線條與標記完整套用主題                       | 14           | Canvas.add_decomposition：預設繪製 A、B、C 所在的無異曲線，並支援曲線標籤與 Legend  | 17           |
| EdgeworthBox：箱框、契約曲線、核、稟賦、價格線與 Walras 均衡完整套用主題   | 19           | 橫軸下方的效果箭頭依 A→B、B→C 方向繪製；效果為零時不繪製箭頭                           | 17           |
| Theme：完成主題系統，新增背景色、標籤比例及所有經濟圖層的預設樣式              | 31           | Haagsma：新增具有閉式解的 Haagsma 效用模型，用於劣等財與季芬財分析                    | 24           |
| themes：新增 paper、monochrome、presentation 與 dark   | 31           | <b>v1.8.0</b>                                                | (2026-09-26) |
| Config：設定檔可選用所有內建主題，並在各種圖形間保留相同的視覺語意             | 33           | Canvas：新增 Axis、Marker、Label 與 Fill 樣式物件                      | 7            |
| econ-viz plot：--theme 支援所有命令列內建主題                | 34           | Stroke：線條樣式統一由 Stroke 表示；個別樣式參數保留為簡寫                         | 12           |
| <b>v1.10.1</b>                                   | (2026-09-26) | Canvas.add_decomposition：新增 Effect、Marker 與 Label 樣式物件       | 17           |
| 一般：套件加入 py.typed，並在 CI 執行 Ruff 與 Mypy 檢查         | 2            | EdgeworthBox：標記點、文字與座標軸接受 Marker、Label 與 Axis                | 19           |
| <b>v1.10.0</b>                                   | (2026-09-26) | <b>v1.7.0</b>                                                | (2026-09-25) |
| econ-viz：Python 3.10 使用 tomli 讀取 TOML 設定檔        | 3            | 一般：CI 測試 Python 3.10–3.13，強制分支覆蓋率，並對可執行的範例做冒煙測試              | 2            |
| Figure：未指定主題與字型時，使用目前的 Config                    | 14           | 版本標籤須通過測試後才會觸發發布                                             | 2            |

|                                                                                         |    |                                                                                                                                                            |    |
|-----------------------------------------------------------------------------------------|----|------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| econ-viz：套件管理與建置改用 uv .....                                                             | 3  | econ-viz：Colab 上的筆記本安裝流程在重新<br>啟動後也能正常運作 .....                                                                                                             | 4  |
| 修正範例腳本，使其可在新簽出的原始碼目錄<br>中直接執行 .....                                                     | 4  | Figure：改善共用面板版面中座標軸標籤的位<br>置與可見度 .....                                                                                                                     | 14 |
| 新增 econ-viz --version，顯示已安裝的套件<br>版本 .....                                              | 34 | decompose_price_effect：價格效果分解：<br>decompose_price_effect()、PriceEffect<br>Decomposition 與 Canvas.add_<br>decomposition()，可畫出 A/B/C 消費組合以<br>及替代與所得效果 ..... | 17 |
| Canvas：支援個別設定各圖的文字與數學字<br>型，不影響 matplotlib 的全域設定 .....                                  | 7  | 效果分解的 API 可以直接從套件根目錄匯入 ...                                                                                                                                 | 17 |
| Canvas 與 Figure 支援分別設定座標軸標籤位<br>置、線條樣式與箭頭樣式 .....                                       | 12 | Canvas.save：純 TikZ 匯出後端；Canvas.<br>save() 與 Figure.save() 接受 .tex .....                                                                                    | 34 |
| Stroke：統一控制畫布、多面板圖、需求圖與<br>Edgeworth 箱形圖中線條的粗細、樣式、顏色<br>與箭頭 .....                       | 12 | <b>v1.4.0</b> (2026-04-09)                                                                                                                                 |    |
| CobbDouglas：修正效用模型的定義域、非對稱<br>CES 的擴張路徑、Cobb-Douglas 的極限情<br>況，以及所得未花完時的飽和最適解 .....     | 20 | 一般：新增 Animator 與 WidgetViewer 的測試 ·                                                                                                                        | 2  |
| 修正效用值接近零或為負時的等高線水準計算 ·                                                                  | 20 | econ-viz[extras]：新增選用依賴<br>animation、interactive 與 all .....                                                                                               | 3  |
| comparative_statics：修正價格、所得接近零<br>或基本需求限制時，比較靜態計算超出定義域<br>的問題 .....                     | 28 | Animator.save：匯出前將影格合成至不透明背<br>景，改善 GIF 匯出穩定性 .....                                                                                                        | 34 |
| <b>v1.6.0</b> (2026-04-24)                                                              |    | Animator：Animator 以 Pillow 產生 GIF 參數變<br>動動畫，不需要 ffmpeg .....                                                                                              | 39 |
| 一般：examples/output/ 的產出檔案不再納入<br>版本控制 .....                                             | 2  | 新增效用參數、價格、所得與僅顯示預算線的<br>GIF 範例 .....                                                                                                                       | 39 |
| Canvas：無異曲線的預設線寬從 2.0 降為 1.8 ·                                                          | 12 | WidgetViewer：WidgetViewer 在 Jupyter 筆記<br>本中提供滑桿控制 .....                                                                                                   | 40 |
| themes.default：預設配色改為色盲友善，並<br>提供 COLORBLIND_CYCLE_RGB 與 COLORBLIND_<br>CYCLE_HEX ..... | 31 | 筆記本互動元件新增數值輸入欄位 .....                                                                                                                                      | 40 |
| Canvas.save：TikZ 座標軸的回歸測試不再依<br>賴顏色索引 .....                                             | 34 | <b>v1.3.2</b> (2026-04-03)                                                                                                                                 |    |
| Animator：動畫範例移到 examples/scripts/<br>animation.py .....                                 | 39 | 一般：發布流程略過 PyPI 上已有的版本 .....                                                                                                                                | 2  |
| <b>v1.5.0</b> (2026-04-21)                                                              |    | EdgeworthBox：Edgeworth 契約曲線與價格線<br>預設為黑色虛線 .....                                                                                                           | 19 |
| 一般：新增效果分解、需求圖、PCC/ICC 路<br>徑、多面板版面與 TikZ 匯出的範例腳本 .....                                  | 2  | <b>v1.3.1</b> (2026-04-03)                                                                                                                                 |    |
| 新增 TikZ 匯出與價格效果分解的回歸測試 ·                                                                | 2  | 一般：套件根目錄的匯出改為延遲載入，公開<br>API 不變 .....                                                                                                                       | 2  |
|                                                                                         |    | Canvas：Canvas 的繪製拆成 canvas.<br>renderers 與 canvas.primitives .....                                                                                         | 12 |

|                                                                        |    |                                                                        |    |
|------------------------------------------------------------------------|----|------------------------------------------------------------------------|----|
| EdgeworthBox：Edgeworth 的內部實作拆成 compute、state 與 plotter 三個模組 .....      | 19 | PricePath：新增 PricePath、IncomePath 與 Canvas.add_path()，支援 PCC/ICC 繪圖 .. | 15 |
| models.registry：新增模型註冊表（models.registry），供命令列工具建立模型 .....              | 20 | DemandDiagram：新增 DemandDiagram，連動顯示商品空間與 Marshall 需求 .....             | 16 |
| CobbDouglas：等高線水準的規則集中在 contours.level_policies .....                  | 20 | <b>v1.1.0</b> (2026-03-30)                                             |    |
| Canvas.save：Canvas、Figure 與 Edgeworth Box 共用同一個圖形匯出器（io.exporter） ...  | 34 | Translog：Translog 模型，以及圖例與無異曲線標籤 .....                                 | 20 |
| econ-viz：命令列錯誤改為拋出 CliConfig Error，結束處理集中在 cli.main .....              | 34 | comparative_statics：comparative_statics 輔助函數 .....                     | 28 |
| <b>v1.3.0</b> (2026-04-03)                                             |    | HomogeneityAnalyzer：分析模組中的 HomogeneityAnalyzer 與 ReturnsToScale ...    | 29 |
| 一般：examples/edgeworth_box.py 涵蓋常見的效用函數組合 .....                         | 2  | <b>v1.0.2</b> (2026-03-29)                                             |    |
| 新增 Edgeworth 測試 .....                                                  | 2  | econ-viz：移除 NumPy 的版本上限，避免在 Colab 上發生衝突 .....                          | 3  |
| EdgeworthBox：新增 EdgeworthBox 與 EquilibriumFocusConfig，支援兩人交換經濟繪圖 ..... | 19 | Canvas：數學式的座標軸標籤不再被重複包裝 ...                                            | 12 |
| 契約曲線、核、Walras 均衡疊圖，以及聚焦於均衡的無異曲線 .....                                  | 19 | <b>v1.0.1</b> (2026-03-29)                                             |    |
| <b>v1.2.3</b> (2026-03-31)                                             |    | econ-viz：固定 numpy<2，避免 Colab 與本機安裝時 ABI 不相容 .....                      | 3  |
| slutsky_matrix：SlutskyMatrix 會檢查對稱性、負半定性與齊次性，條件不成立時發出警告 .....          | 29 | 放寬 Python 與 NumPy 的版本限制；pytest 固定在 8.x .....                           | 3  |
| econ-viz models：命令列工具支援 Quasi Linear、StoneGeary 與 Translog .....       | 34 | <b>v1.0.0</b> (2026-03-28)                                             |    |
| <b>v1.2.2</b> (2026-03-31)                                             |    | 一般：首次發布 .....                                                          | 44 |
| 一般：新增 PCC/ICC 範例產生器與測試 .....                                           | 2  |                                                                        |    |
| PricePath：PCC/ICC 路徑預設經過平滑處理，端點略微延長，除非指定否則不顯示標記 ...                    | 15 |                                                                        |    |
| PCC/ICC 路徑改用獨立配色，並增加需求圖商品空間的留白 .....                                   | 15 |                                                                        |    |
| <b>v1.2.0</b> (2026-03-30)                                             |    |                                                                        |    |
| Layout：多面板 Figure 版面與 Layout 列舉 ...                                    | 15 |                                                                        |    |

## 指令索引

### A

|                              |    |
|------------------------------|----|
| <code>add_budget</code>      | 9  |
| <code>add_equilibrium</code> | 10 |
| <code>add_point</code>       | 11 |
| <code>add_ray</code>         | 10 |
| <code>add_utility</code>     | 8  |
| <code>Animator</code>        | 39 |
| <code>ArrowStyle</code>      | 12 |
| <code>Axis</code>            | 13 |

### C

|                                       |       |
|---------------------------------------|-------|
| <code>Canvas</code>                   | 6, 7  |
| <code>Canvas.add_*</code>             | 6     |
| <code>Canvas.add_decomposition</code> | 18    |
| <code>Canvas.add_path</code>          | 16    |
| <code>Canvas.save</code>              | 6, 34 |
| <code>Canvas.show</code>              | 6     |
| <code>CES</code>                      | 21    |
| <code>CobbDouglas</code>              | 20    |
| <code>comparative_statics</code>      | 28    |
| <code>Config.load</code>              | 33    |
| <code>Config.reset</code>             | 33    |
| <code>Config.use</code>               | 33    |
| <code>CustomUtility</code>            | 27    |

### D

|                                     |    |
|-------------------------------------|----|
| <code>decompose_price_effect</code> | 17 |
| <code>DemandDiagram</code>          | 16 |

### E

|                                 |    |
|---------------------------------|----|
| <code>econ-viz help</code>      | 35 |
| <code>econ-viz init</code>      | 38 |
| <code>econ-viz models</code>    | 35 |
| <code>econ-viz plot</code>      | 35 |
| <code>econ-viz solve-tex</code> | 38 |
| <code>econ_viz.models</code>    | 5  |
| <code>EdgeworthBox</code>       | 19 |
| <code>Effect</code>             | 18 |

### F

|                     |    |
|---------------------|----|
| <code>Figure</code> | 14 |
| <code>Fill</code>   | 14 |

### H

|                                  |    |
|----------------------------------|----|
| <code>Haagsma</code>             | 24 |
| <code>HomogeneityAnalyzer</code> | 29 |

### I

|                         |    |
|-------------------------|----|
| <code>IncomePath</code> | 15 |
|-------------------------|----|

### L

|                            |    |
|----------------------------|----|
| <code>Label</code>         | 14 |
| <code>Layout</code>        | 15 |
| <code>Legend</code>        | 18 |
| <code>Leontief</code>      | 22 |
| <code>levels.around</code> | 6  |
| <code>LineStyle</code>     | 12 |

### M

|                                 |    |
|---------------------------------|----|
| <code>Marker</code>             | 13 |
| <code>maximum</code>            | 23 |
| <code>MultiGoodCD</code>        | 27 |
| <code>MultiGoodCD.freeze</code> | 27 |

### P

|                                 |    |
|---------------------------------|----|
| <code>parse_latex</code>        | 41 |
| <code>PerfectSubstitutes</code> | 22 |
| <code>PricePath</code>          | 15 |

### Q

|                          |    |
|--------------------------|----|
| <code>QuasiLinear</code> | 24 |
|--------------------------|----|

### S

|                             |    |
|-----------------------------|----|
| <code>Satiation</code>      | 26 |
| <code>save</code>           | 11 |
| <code>show</code>           | 11 |
| <code>slutsky_matrix</code> | 29 |
| <code>solve</code>          | 5  |
| <code>StoneGeary</code>     | 25 |
| <code>Stroke</code>         | 13 |

### T

Theme ..... 31

themes ..... 31

Translog ..... 21

## W

WidgetViewer ..... 41

## 參考文獻

- Arrow, K. J., Chenery, H. B., Minhas, B. S., & Solow, R. M. (1961). Capital-labor substitution and economic efficiency. *Review of Economics and Statistics*, 43(3), 225–250.
- Christensen, L. R., Jorgenson, D. W., & Lau, L. J. (1975). Transcendental logarithmic utility functions. *American Economic Review*, 65(3), 367–383.
- Cobb, C. W., & Douglas, P. H. (1928). A theory of production. *American Economic Review*, 18(1), 139–165.
- Edgeworth, F. Y. (1881). *Mathematical psychics*. C. Kegan Paul.
- Geary, R. C. (1950). A note on "A constant-utility index of the cost of living". *Review of Economic Studies*, 18(1), 65–66.
- Haagsma, R. (2012). A convenient utility function with Giffen behaviour. *ISRN Economics*, 2012, Article 608645. <https://doi.org/10.5402/2012/608645>
- Hicks, J. R. (1939). *Value and capital*. Clarendon Press.
- Kugler, P., & Andrews, K. (1996). Graphical analysis and the visually impaired in undergraduate economics courses. *The Journal of Economic Education*, 27(3), 224–228. <https://doi.org/10.1080/00220485.1996.10844911>
- Slutsky, E. (1915). Sulla teoria del bilancio del consumatore. *Giornale degli Economisti e Rivista di Statistica*, 51, 1–26.
- Stone, R. (1954). Linear expenditure systems and demand analysis: An application to the pattern of British demand. *Economic Journal*, 64(255), 511–527.
- thriveth. (2014, January 22). *A color blind/friendly color cycle for Matplotlib line plots* [Source code]. <https://gist.github.com/thriveth/8560036>