BONUS

B

案例演練：使用 Seaborn 發現辛普森悖論

在進行資料分析時，可能會得到錯誤的結果，而**辛普森悖論**（Simpson's Paradox）是最常見的現象之一。何謂辛普森悖論？當我們合併資料時，某一組的結果比另一組來得高，然而當我們拆分資料時，卻顯示出相反的結果。這一個現象就叫做辛普森悖論。例如：假設現在有學生 A 和學生 B，他們兩人都回答了 100 個問題。其中，學生 A 正確回答了 50% 的問題，而學生 B 正確回答了 80%。顯然，學生 B 的分數是更高的：

Student	Raw Score	Percent Correct
A	50/100	50
B	80/100	80

假設這兩位學生的考題難易程度完全相反。其中，學生 A 的考題中有 95% 是困難的，剩餘 5% 是簡單的；而學生 B 的考題中有 5% 是困難的，剩餘 95% 則是簡單的。下一頁的表格顯示了兩位學生在不同難度的題目上，分別答對的題數及百分比：

Student	Difficult	Easy	Difficult Percent	Easy Percent	Percent
A	45/95	5/5	47	100	50
B	2/5	78/95	40	82	80

從上表可見，學生 A 在困難題和簡單題的答對率（Difficult Percent 和 Easy Percent）都比學生 B 來得高，但整體的答對率（Percent）卻要低得多，這就是辛普森悖論的典型例子。

接下來，我們會分析鑽石資料集。在一開始的結果中，我們會得到『高品質的鑽石比低品質的鑽石來的便宜』的困惑結果。然後，我們便要來揭露辛普森悖論，並對資料進行更深入的檢視。

動手做

01 讀入鑽石資料集：

In

```
dia = pd.read_csv('data/diamonds.csv')
dia
```

Out

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

02 在開始分析之前，讓我們將 cut、color 和 clarity 欄位改為分類欄位，並排序其中的資料值。排序在越後面的分類，代表品質就越高：

In

```
cut_cats = ['Fair', 'Good', 'Very Good', 'Premium', 'Ideal']
color_cats = ['J', 'I', 'H', 'G', 'F', 'E', 'D']
clarity_cats = ['I1', 'SI2', 'SI1', 'VS2', 'VS1', 'VVS2', 'VVS1', 'IF']
```

最後面的 Ideal、D 和 IF 代表最高品質；最前面的 Fair、J 和 I1 代表最低品質

```
dia2 = (dia
    .assign(cut=pd.Categorical(dia['cut'],
                categories=cut_cats,
                ordered=True),
            color=pd.Categorical(dia['color'],
                  categories=color_cats,
                  ordered=True),
            clarity=pd.Categorical(dia['clarity'],
                    categories=clarity_cats,
                    ordered=True)))
dia2
```

Out

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

03 Seaborn 會根據分類變數的順序作為圖表 x 軸的標籤。讓我們根據 cut、color 和 clarity 欄位中，每個分類的平均價格來繪製長條圖：

In

```
import seaborn as sns
import matplotlib.pyplot as plt
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(14,4))
sns.barplot(x='color', y='price', data=dia2, ax=ax1)
sns.barplot(x='cut', y='price', data=dia2, ax=ax2)
sns.barplot(x='clarity', y='price', data=dia2, ax=ax3)
fig.suptitle('Price Decreasing with Increasing Quality?')
```

圖 B.1 使用 Seaborn 繪製不同欄位中，每個分類平均價格的長條圖。

04 根據目前的 color 欄位分類，價格似乎有下降的趨勢（**編註**：代表 color 分類的品質越高，對應的平均價格越低）。在 cut 欄位和 clarity 欄位中品質最高的分類（Ideal 和 IF），對應到的平均價格也很低。換句話說，鑽石的質量越高，價格卻越低，這就是辛普森悖論出現的地方。為什麼會這樣呢？讓我們更深入地分析資料，為每個 clarity 分類（存有鑽石的純淨度資訊）繪製不同 color 分類的價格圖表：

In

```
grid = sns.catplot(x='color', y='price', col='clarity',
                   col_wrap=4, data=dia2, kind='bar')
```

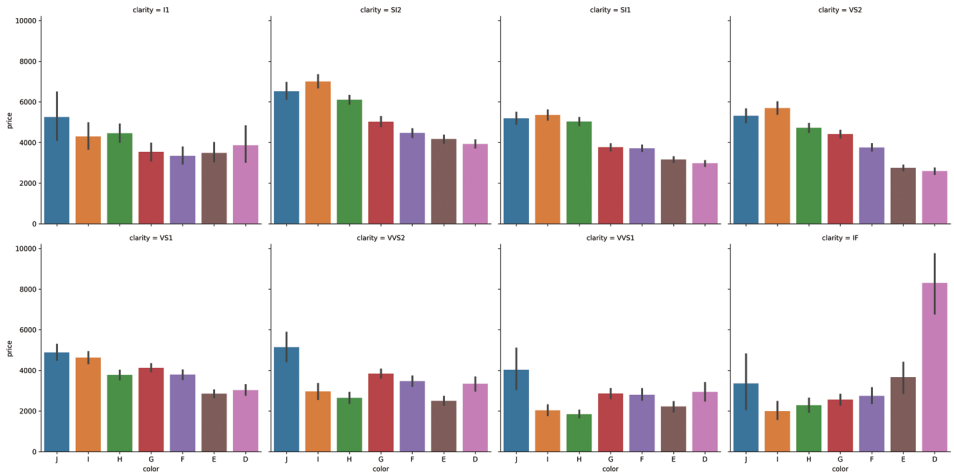

圖 B.2 為每個 clarity 分類繪製不同 color 分類的價格圖表。

05 雖然價格似乎隨著顏色質量的提高而下降，但當純淨度處於最高等級時，價格不但沒有下降，反而是大幅上漲。目前，我們只觀察鑽石的價格而忽略了鑽石的大小。讓我們重新繪製步驟 3 中的圖表，不過改用鑽石的克拉數（carat 欄位）來代替 y 軸的平均價格（**編註**：步驟 4 和步驟 5 分別是用 Figure 層次的 catplot() 的 Axes 層次的 barplot() 來繪製長條圖，可以從程式碼長度看出前者較為簡潔）。

In

```
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(14,4))
sns.barplot(x='color', y='carat', data=dia2, ax=ax1)
sns.barplot(x='cut', y='carat', data=dia2, ax=ax2)
sns.barplot(x='clarity', y='carat', data=dia2, ax=ax3)
fig.suptitle('Diamond size decreases with quality')
```

← 把 y 參數的值改為 carat

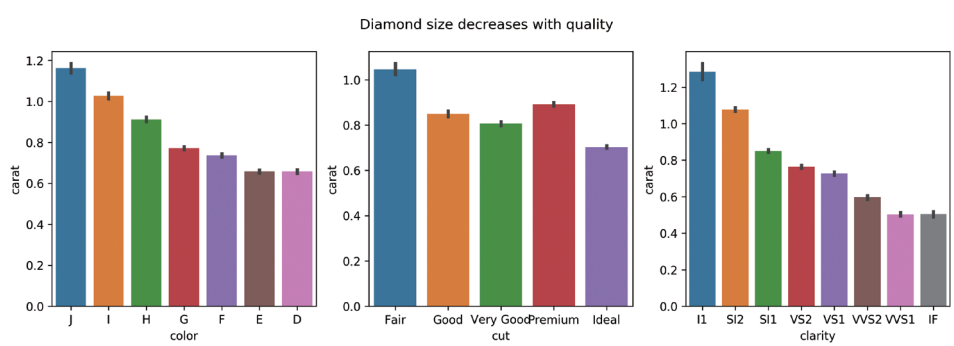

圖 B.3 使用 Seaborn 視覺化鑽石大小與不同欄位分類的關係。

06 目前得到的資訊為：質量更高的鑽石會更小，這在直覺上是合理的。讓我們建立一個新變數，將 carat 欄位中的值分成 5 個大小相等的區間，然後繪製一個**點線圖** (point plot)。從輸出的圖表可見，當根據大小進行分組後，質量更高的鑽石確實價格更高：

In

```
dia2 = (dia2
    .assign(carat_category=pd.qcut(dia2.carat, 5))
)
from matplotlib.cm import Greys
greys = Greys(np.arange(50,250,40))
grid = sns.catplot(x='clarity', y='price', data=dia2,
                   hue='carat_category', col='color',
                   col_wrap=4, kind='point')
grid.fig.suptitle('Diamond price by size, color and clarity',
                  y=1.02, size=20)
```

新增一個分類欄位，用來表示鑽石的大小

將 carat 欄位中的值分成 5 個相同大小的區間

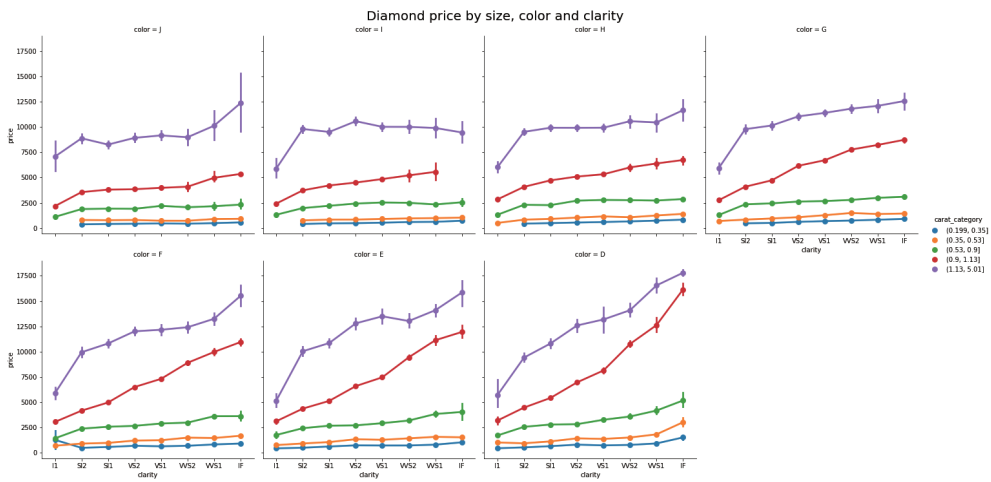

圖 B.4 使用 Seaborn 繪製點線圖。

了解更多

步驟 3 和 5 中的長條圖可以使用更進階的 **PairGrid 類別**，直接繪製出雙變量關係。使用 PairGrid 具有兩個步驟：第一個步驟是呼叫建構子並告知它哪些變數是 x，哪些是 y；第二個步驟呼叫 map()，將圖表應用於 x 和 y 欄位的所有組合：

In

```
g = sns.PairGrid(dia2, height=5,
                 x_vars=['color', 'cut', 'clarity'],
                 y_vars=['price'])
g.map(sns.barplot)
g.fig.suptitle('Replication of Step 3 with PairGrid', y=1.02)
```

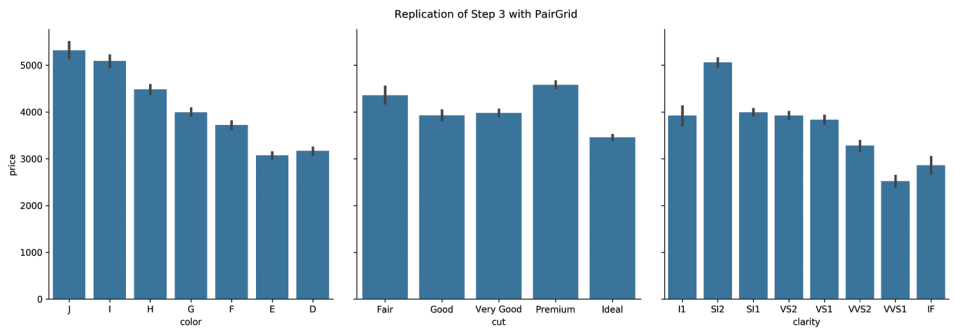

圖 B.5 使用更進階的 PairGrid 類別來繪製長條圖。