

$$(X, O) = e^{-\frac{x^2}{2\sigma^2}}$$

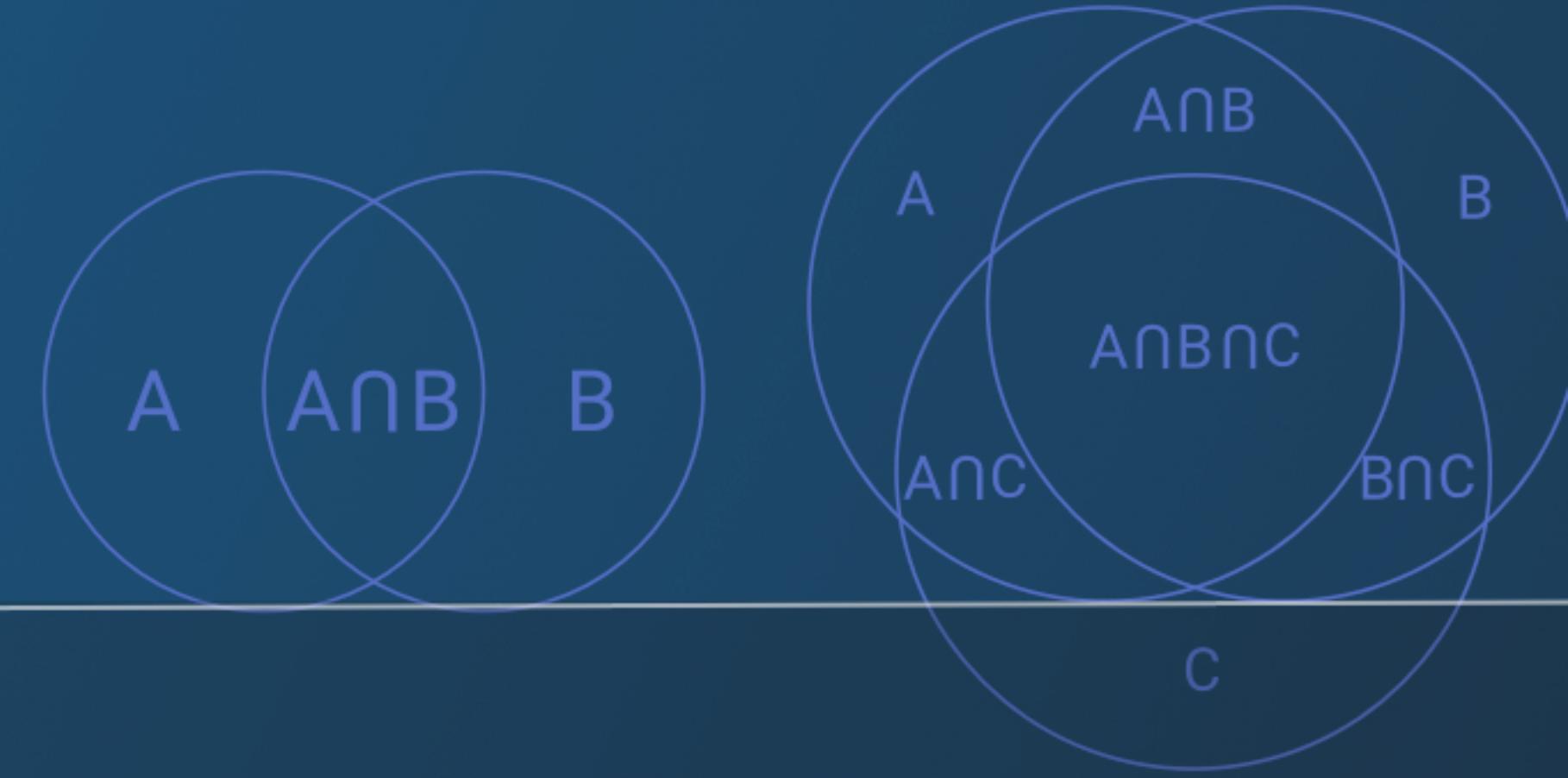
$$x(X, O) = -\frac{x}{\sigma^2} G(X, O) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

$$xx(X, O) = \frac{x^2 - \sigma^2}{\sigma^4} G(X, O) = \frac{x^2 - \sigma^2}{\sigma^4} e^{-\frac{x^2}{2\sigma^2}}$$

$$xxx(X, O) = -\frac{x^3 - x\sigma^2}{\sigma^6} G(X, O) = -\frac{x^3 - x\sigma^2}{\sigma^6} e^{-\frac{x^2}{2\sigma^2}}$$

Julia 程式語言學習馬拉松

Day 23



$$\ln(x + \sqrt{1 + x^2}) + x - \frac{1}{x + \sqrt{1 + x^2}} \left(1 + \frac{x}{\sqrt{1 + x^2}} \right)$$

$$\ln(x + \sqrt{1 + x^2}) + x - \frac{1}{x + \sqrt{1 + x^2}} \left(\frac{\sqrt{1 + x^2} + x}{\sqrt{1 + x^2}} \right)$$



cupay 陪跑專家 : Andy Tu

Julia 基礎圖表繪製

Plots 套件





重要知識點



- 認識不同後端繪圖引擎
- 認識 Plots 能提供的圖表及 3D 繪圖與動畫



Plots 套件



- Plots 是一個提供全面的繪圖功能的套件，其包含多種後端引擎，供使用者挑選而不必修改程式。
- Plots 提供一個強大、直覺而精簡，並保有彈性的介面。有多樣的繪圖參數供使用者調整，並有適當的預設值。



後端引擎



- Plots 支援各種後端引擎，各引擎有各種特色。
- PyPlot：就是 Python Matplotlib 函式庫，功能完整，支援 3D，有獨立視窗。
- Plotly：互動性好，成熟套件，較難客製化
- GR：快速，繪圖品質普通，缺乏互動性
- UnicodePlots：可以在終端機上繪圖

```
using Plots  
gr() # 選擇後端引擎
```

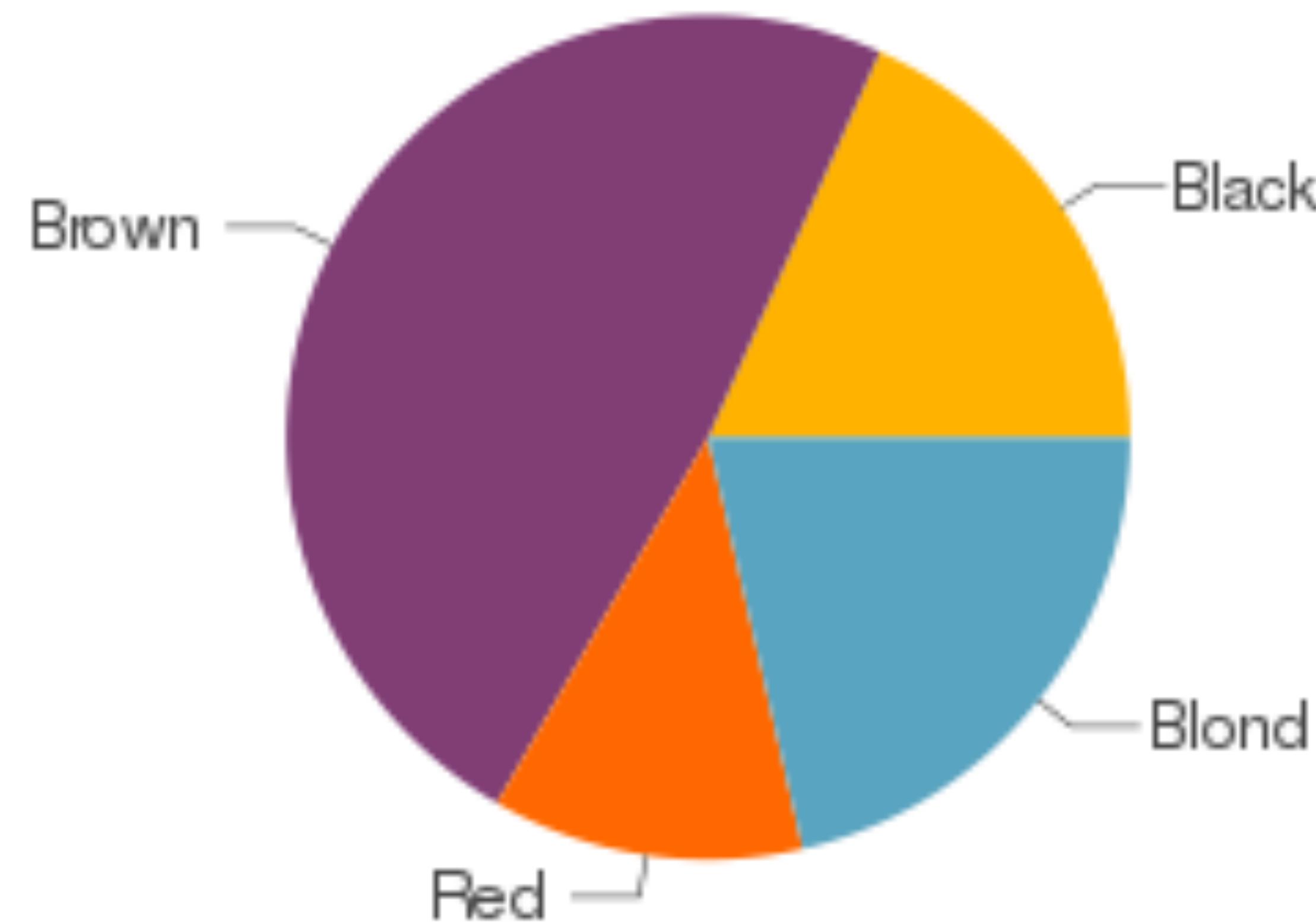


圓餅圖

- 圓餅圖 (pie chart) 可以呈現不同類別之間所佔的比例。
- 圓餅圖的使用需要兩個參數，第一個參數要給定類別欄位，第二個參數要給定相對類別的出現頻率。

```
haireyecolor = dataset("datasets",
"HairEyeColor")
haircolor = by(haireyecolor, :Hair,
Freq=:Freq => sum)
pie(haircolor[:Hair],
haircolor[:Freq])
```

圓餅圖





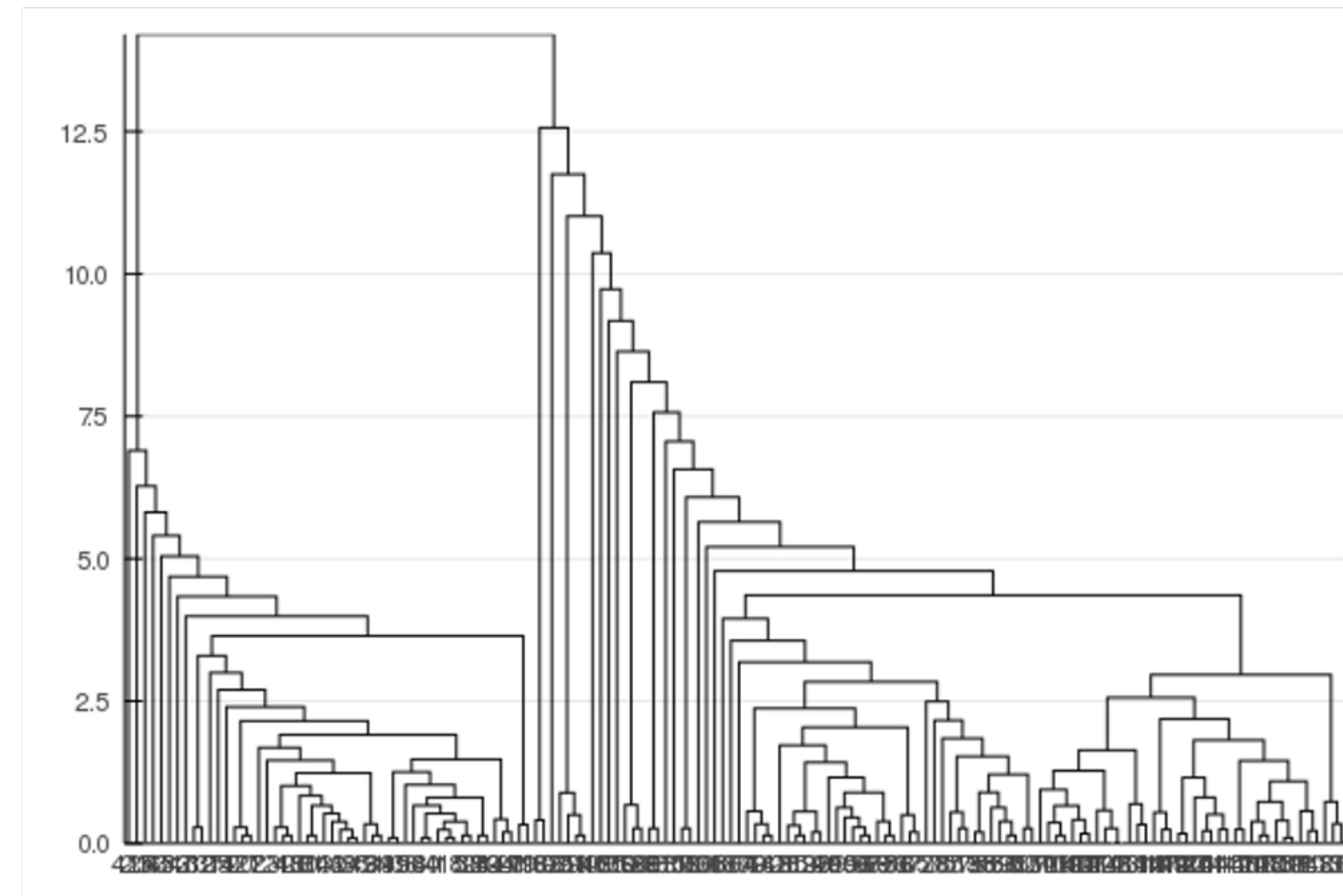
樹狀圖

- 樹狀圖 (dendrogram) 是呈現階層式分群法結果的好工具。
- 在使用 Clustering 中的 hcluster 完成階層式分群法結果，便可以使用延伸套件 StatsPlots 進行繪製，plot 即可。

```
result = hclust(D, linkage=:single)  
using StatsPlots  
plot(result)
```



樹狀圖





Marginal Histogram

- Marginal histogram 是結合了密度的呈現及直方圖的一種圖表，它可以一次閱覽兩個連續型變數的全貌。
- 使用 DataFrame 儲存資料的話需要在前方加上 @df 將 DataFrame 轉換成可以繪圖的格式。iris 為一 DataFrame。

```
using StatsPlots  
@df iris marginalhist(:PetalLength,  
:PetalWidth)
```

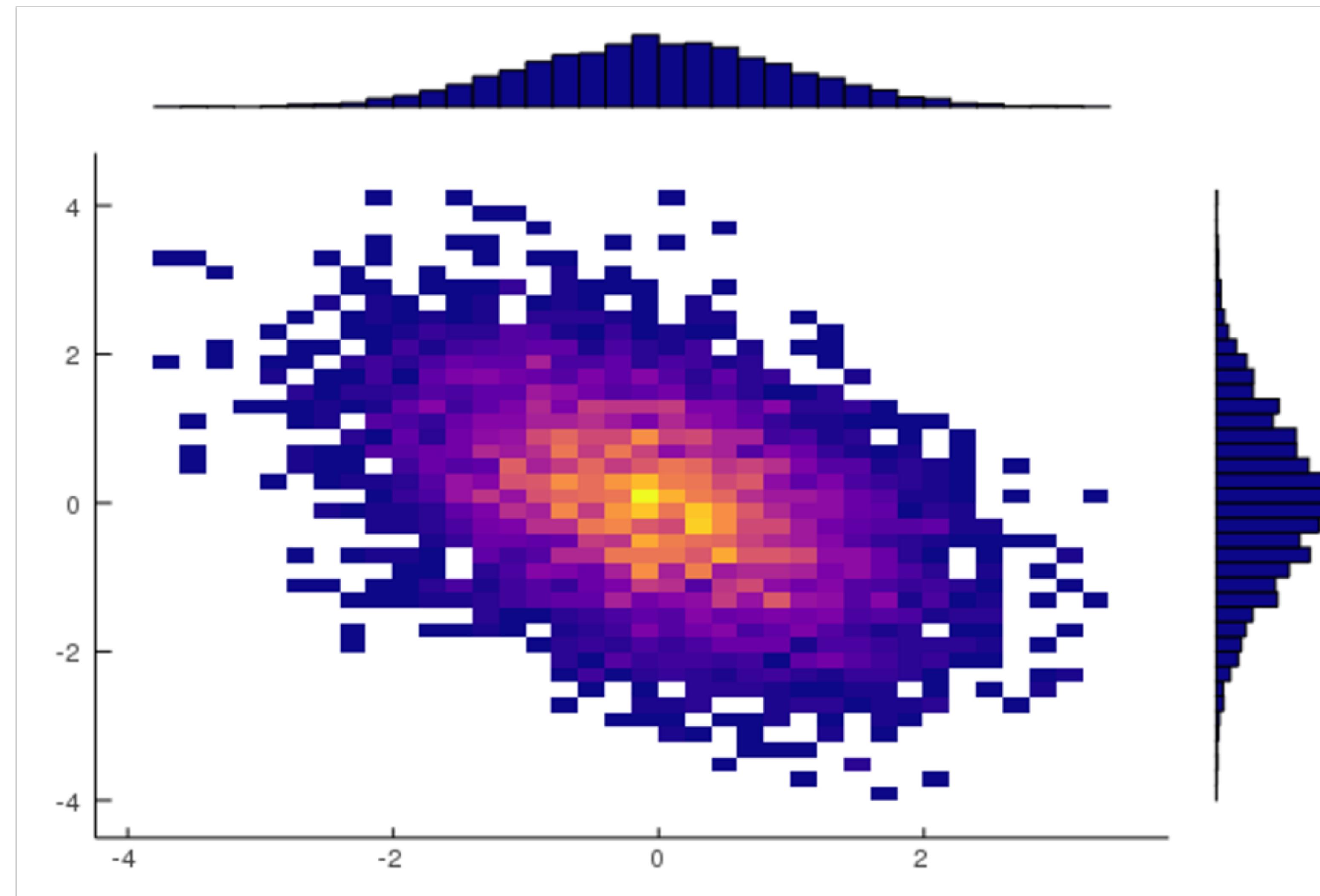


Marginal histogram

```
n = 5000
x = randn(n)
y = -0.5x + randn(n)
marginalhist(x, y, fc=:plasma, bins=40)
```



Marginal histogram



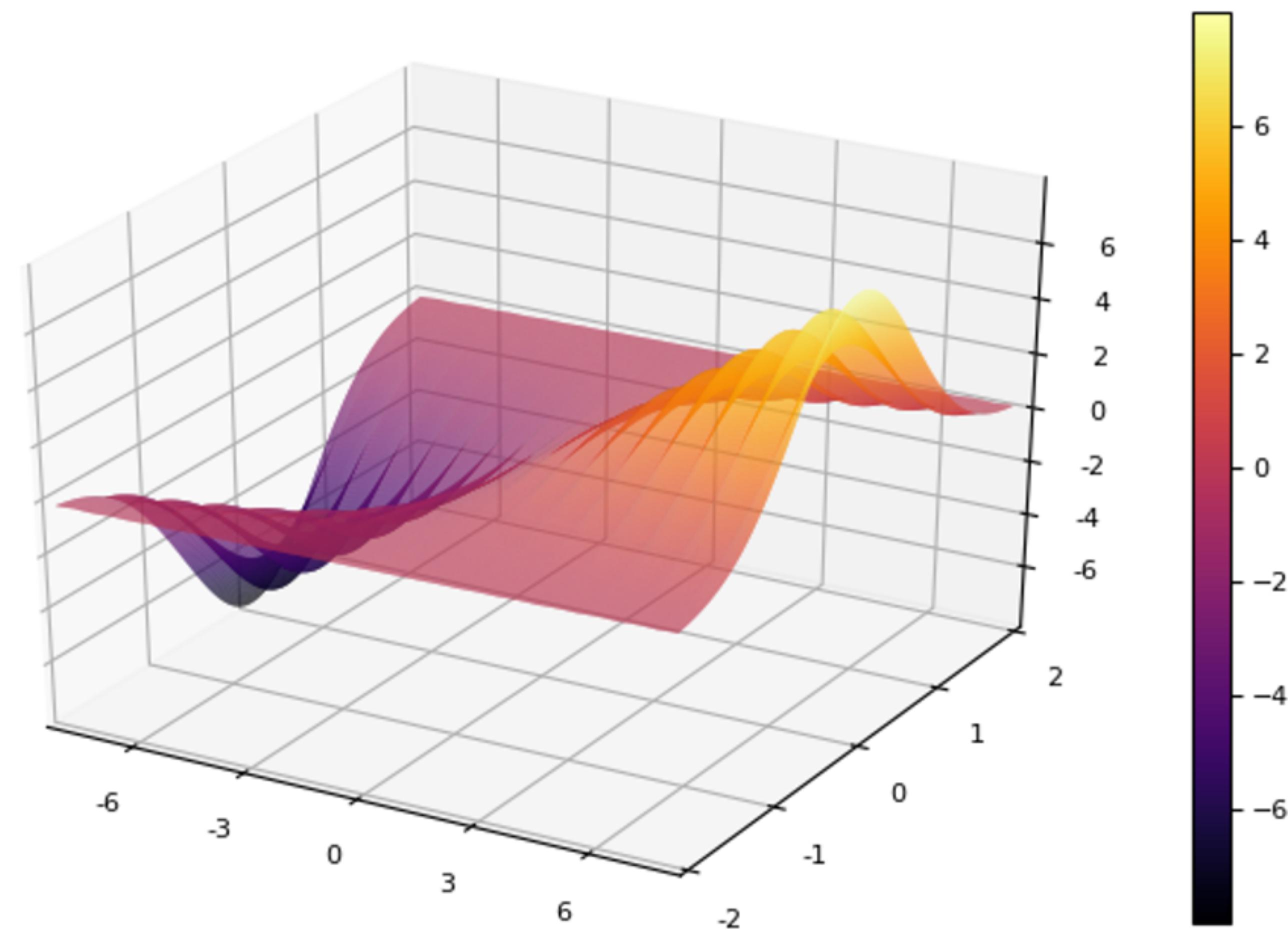


3D 繪圖

- Plots 有提供繪製 3D 繪圖的功能。
- 在 plot 依序給定 x、y、z 三軸的數值或是函數，並設定 st=:surface 即可。

```
n = 500
x = range(-8., stop=8., length=n)
y = range(-2., stop=2., length=n)
f(x,y) = x*exp(-(x-round(Int, x))^2-
y^2)
p = plot(x, y, f, st=:surface)
```

3D 繪圖





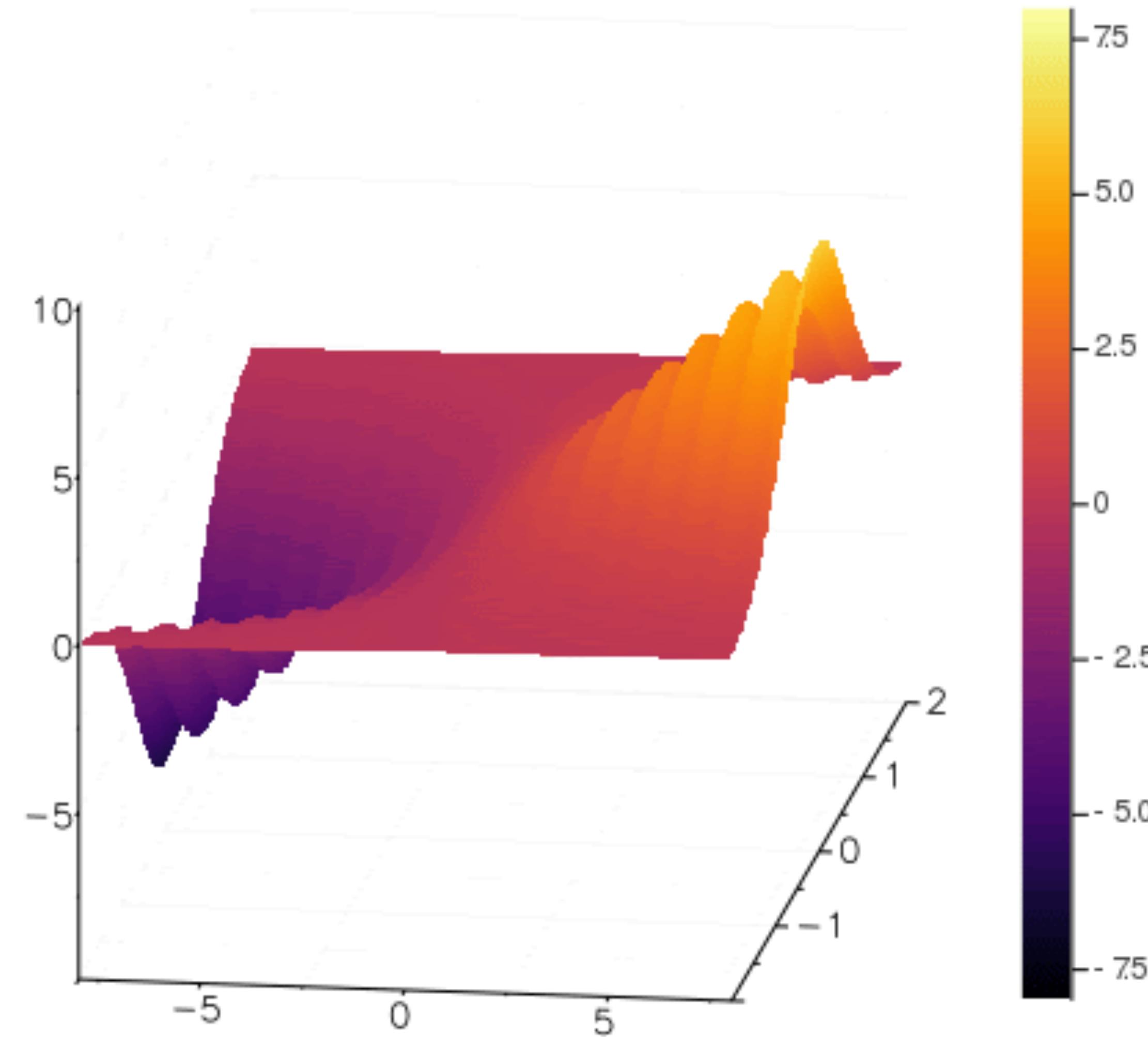
3D 動畫



- Plots 有提供繪製 3D 動畫的功能。
- 承接上面的程式碼，使用 @gif 及 for 迴圈即可以繪製動畫。這邊動畫是旋轉視角，將每次繪製完成的 p 用 plot! 繪製到畫布上，並用 camera 參數更改視角。

```
@gif for i in range(0, stop=2π,  
length=100)  
    p = plot(x, y, f, st=:surface)  
    plot!(p, camera=(15*cos(i), 40))  
end
```

3D 動畫



[請點此參閱範例中3D圖片Gif檔](#)



動畫輸出成 gif

- 繪製完成的動畫可以以 gif 輸出或是以 mp4 輸出。
- 要繪製 gif 影片可以參考以下程式碼。

```
p = plot(1)
ani = @animate for x=0:0.1:2π
    push!(p, 1, sin(x))
end
gif(ani, "animate.gif")
```



動畫輸出成 gif

- 要繪製 mp4 影片可以參考以下程式碼。

```
p = plot(1)
ani = @animate for x=0:0.1:2π
    push!(p, 1, sin(x))
end
mp4(ani, "animate.mp4", fps = 30)
```

知識點 回顧

- 選擇不同後端引擎
- 繪製：
 - 圓餅圖
 - 樹狀圖
 - Marginal histogram
 - 3D 繪圖
 - 3D 動畫



推薦閱讀

- Plots 官方文件





解題時間

請跳出 PDF 至官網 Sample Code
& 作業開始解題