

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

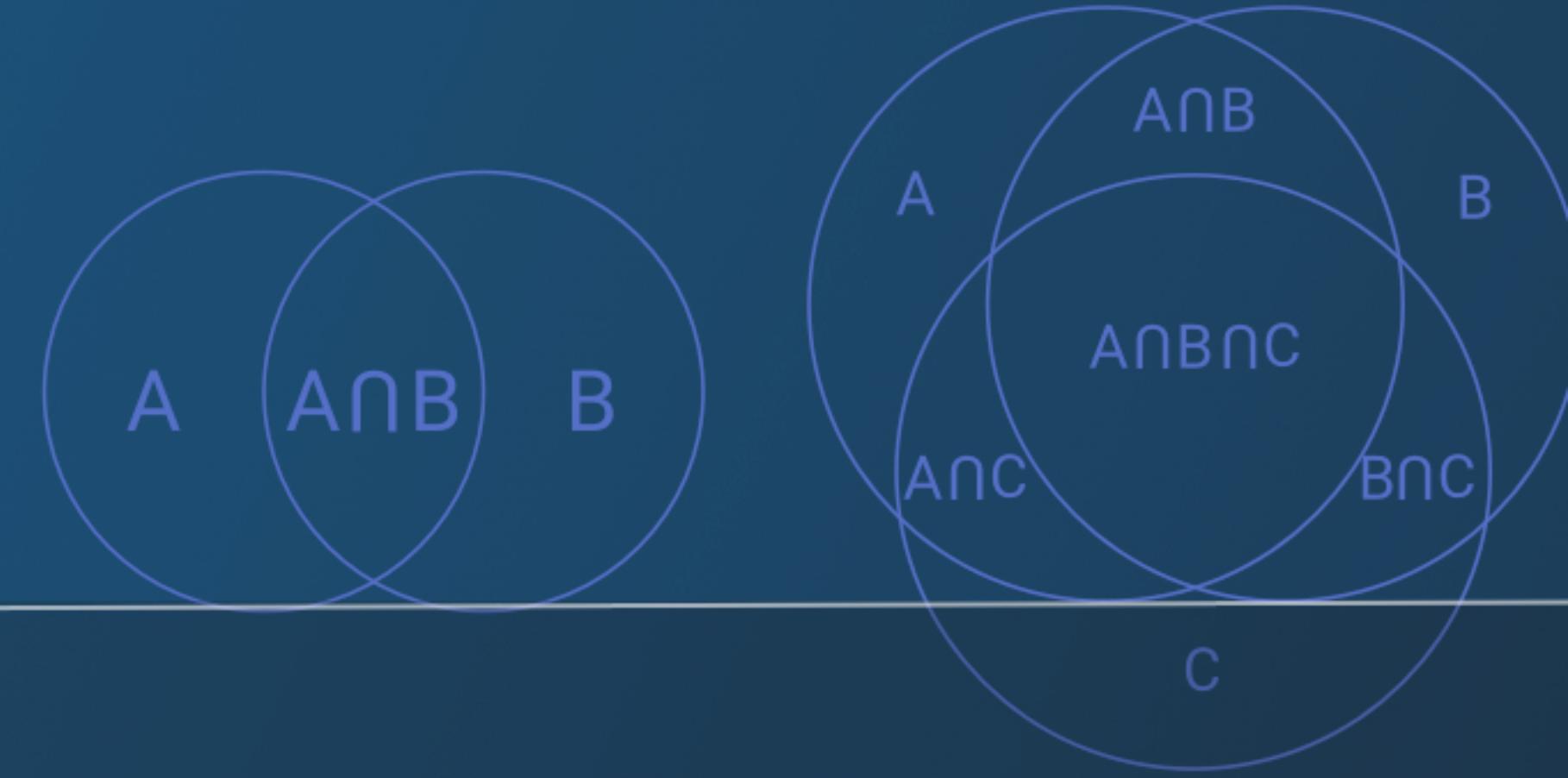
$$x(X, O) = -\frac{x}{\sigma^2} G(X, O) = -\frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}$$

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

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

Julia 程式語言學習馬拉松

Day 03



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



cupay 陪跑專家 : James Huang

Julia 套件管理 (Package Management)





重要知識點



- Julia 內建套件管理工具 (Package Manager): Pkg.jl，可以用來安裝、更新、移除套件。
- 可以在兩種不同的模式下進行套件的管理：
 - REPL Mode
 - Functional API Mode
- 在今天的內容中，主要會以在 Jupyter Notebook 中使用 Functional API Mode 做為示範。



REPL Mode – 使用 REPL 套件管理



- 開啟 Julia REPL，輸入 “]”即可進入套件管理。當看到 (v1.2) Pkg > 提示字元時，就是已成功進入。
- 輸入 st 或 status 指令，顯示所有自行安裝的套件。
- 按 BackSpace 鍵或是 Ctrl+C 都可以退出套件管理命令提示。

```
Documentation: https://docs.julialang.org
Type "?" for help, "]?" for Pkg help.

Version 1.1.1 (2019-05-16)
Official https://julialang.org/ release

(v1.1) pkg> st
Status `C:\Users\yellowcc\.julia\environments\v1.1\Project.toml`
(empty environment)

(v1.1) pkg>
```



REPL Mode – 安裝套件



- 在套件管理提示字元，輸入 add IJulia。
- 螢幕顯示類似畫面訊息，進行套件下載及安裝。相依的套件也會一併下載安裝。
- 在 Julia 裡面，大小寫是有分別的，安裝套件時請留意套件名稱的大小寫。

```
(v1.1) pkg> add IJulia
Cloning default registries into `C:\Users\yellowcc\.julia`
Cloning registry from "https://github.com/JuliaRegistries/General.git"
  Added registry `General` to `C:\Users\yellowcc\.julia\registries\General`
Resolving package versions...
Installed VersionParsing ━━━━━━ v1.1.3
Installed Compat ━━━━━━ v2.1.0
Installed SoftGlobalScope ━━━━ v1.0.10
Installed ZMQ ━━━━ v1.0.0
Installed IJulia ━━━━ v1.18.1
Installed MbedTLS ━━━━ v0.6.8
Installed BinaryProvider ━━━━ v0.5.4
Installed JSON ━━━━ v0.20.0
Installed Conda ━━━━ v1.2.0
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Project.toml`
[7073ff75] + IJulia v1.18.1
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Manifest.toml`
[b99e7846] + BinaryProvider v0.5.4
[34da2185] + Compat v2.1.0
[8f4d0f93] + Conda v1.2.0
[7073ff75] + IJulia v1.18.1
Documentation: https://docs.julialang.org
Type "?" for help, "]?" for Pkg help.
Version 1.1.1 (2019-05-16)
Official https://julialang.org/ release
```



從 Julia 開啟 Jupyter Notebook



- 回到 REPL，輸入 `using IJulia` 及 `notebook()` 開啟 Jupyter Notebook。第一次會詢問是否安裝 miniconda 3，按 y 繼續。
- 安裝成功後，會開啟 Jupyter Notebook。

```
julia> using IJulia
[ Info: Precompiling IJulia [7073ff75-c697-5162-941a-fcdaad2a7d2a]

julia> notebook()
install Jupyter via Conda, y/n? [y]: y
[ Info: Downloading miniconda installer ...
[ Info: Installing miniconda ...
[ Info: Running `conda config --add channels defaults --file 'C:\Users\...
e` in root environment
Warning: 'defaults' already in 'channels' list, moving to the top
[ Info: Running `conda update -y conda` in root environment
Solving environment: -
```



新增 Julia Notebook



jupyter

Files Running Clusters

Select items to perform actions on them.

0 /
3D Objects

Upload New Notebook:
Name: Julia 1.2.0
Python 3

Logout

jupyter Untitled1 Last Checkpoint: 幾秒前 (unsaved changes)

File Edit View Insert Cell Kernel Widgets Help

Kernel starting, please wait... Trusted Julia 1.2.0

In []:



Functional API Mode – 安裝套件 DataFrames



- 在 Notebook 中輸入 using Pkg 啟用套件管理。
- Pkg.add("DataFrames") 安裝套件。

```
In [1]: using Pkg

In [2]: Pkg.installed()

Out[2]: Dict{String,Union{Nothing, VersionNumber}} with 1 entry:
    "IJulia" => v"1.18.1"

In [3]: Pkg.add("DataFrames")

    Updating registry at `C:\Users\yellowcc\.julia\registries\General`
    Updating git-repo `https://github.com/JuliaRegistries/General.git`
    Resolving package versions...
    Installed SortingAlgorithms ━━━━━━ v0.3.1
    Installed PooledArrays ━━━━━━ v0.5.1
    Installed Missings ━━━━━━ v0.4.1
    Installed Requires ━━━━━━ v0.5.2
    Installed IteratorInterfaceExtensions ━ v1.0.0
    Installed OrderedCollections ━━━━ v1.1.0
    Installed DataFrames ━━━━ v0.18.3
    Installed StatsBase ━━━━ v0.30.0
    Installed TableTraits ━━━━ v1.0.0
    Installed Reexport ━━━━ v0.2.0
    Installed Tables ━━━━ v0.2.5
    Installed DataStructures ━━━━ v0.15.0
    Installed CategoricalArrays ━━━━ v0.5.2
    Updating `C:\Users\yellowcc\.julia\environments\v1.1\Project.toml`
    [a93c6f00] + DataFrames v0.18.3
```



Functional API Mode – 安裝套件特定版本



- 在 Notebook 中輸入 using Pkg 啟用套件管理。
- Pkg.add(PackageSpec(name="PyCall", version="1.91.0")) 安裝套件的特定版本。

```
In [4]: Pkg.add(PackageSpec(name="PyCall", version="1.91.0"))
```

```
Resolving package versions...
Installed MacroTools - v0.5.0
Installed Tokenize -- v0.5.3
Installed CSTParser -- v0.5.2
Installed PyCall ---- v1.91.0
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Project.toml`
[438e738f] + PyCall v1.91.0
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Manifest.toml`
[00ebfdb7] + CSTParser v0.5.2
[1914dd2f] + MacroTools v0.5.0
[438e738f] + PyCall v1.91.0
[0796e94c] + Tokenize v0.5.3
```



Functional API Mode – 更新套件



- `Pkg.update()` 即可更新所有套件。
- `Pkg.update("PyCall")` 更新單一套件 PyCall。

In [5]: `Pkg.update()`

```
Updating registry at `C:\Users\yellowcc\.julia\registries\General`
Updating git-repo `https://github.com/JuliaRegistries/General.git`
Resolving package versions...
Installed PyCall - v1.91.2
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Project.toml`
[438e738f] ↑ PyCall v1.91.0 ⇒ v1.91.2
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Manifest.toml`
[438e738f] ↑ PyCall v1.91.0 ⇒ v1.91.2
Building PyCall → `C:\Users\yellowcc\.julia\packages\PyCall\ttONZ\deps\build.log`
```



Functional API Mode – 刪除套件



- Pkg.rm("PyCall") 刪除 PyCall 套件。

```
In [6]: Pkg.rm("PyCall")
```

```
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Project.toml`  
[438e738f] - PyCall v1.91.2  
Updating `C:\Users\yellowcc\.julia\environments\v1.1\Manifest.toml`  
[00ebfdb7] - CSTParser v0.5.2  
[1914dd2f] - MacroTools v0.5.0  
[438e738f] - PyCall v1.91.2  
[0796e94c] - Tokenize v0.5.3
```

知識點 回顧

- 今天內容中介紹了 Julia 內建的套件管理工具 Pkg.jl，套件管理可以透過 2 種模式進行操作。
 - REPL Mode
 - Functional API Mode
- 除了 Julia 內建的模組功能之外，愈來愈多開發者提供功能強大的套件，透過內建的套件管理功能，可以新增、更新、刪除套件，協助我們進行更多的開發工作。



推薦閱讀

- 深入淺出 Julia 套件管理 ([Package Management](#))
- 官方文件：[Pkg](#)，內有 REPL 模式的指令範例。





解題時間

請跳出 PDF 至官網 Sample Code
& 作業開始解題