

# QuantLibXL 財工增益集操作手冊

## Excel Financial AddIn QuantLibXL Manual

2020 March 第二版

昀騰金融科技

技術長

董夢雲 博士

[dongmy@ms5.hinet.net](mailto:dongmy@ms5.hinet.net)

# 主 題

## Part I 基礎類別(10hrs)

- 一、QuantLibXL 的安裝與使用(4hrs)
- 二、日期與時間類別(4hrs)
- 三、通用與數學類別(3hrs)
- 四、統計與時間數列類別(3hrs)

## Part II 財務市場類別(12hrs)

- 五、Quote、Interest Rate 類別(2hrs)
- 六、Yield Term Structure 類別(2hrs)
- 七、Index、FRA 類別(2hrs)
- 八、Volatility Term Structure 類別(2hrs)
- 九、Bond 相關類別(4hrs)

## Part III 衍生金融工具類別(14hrs)

十、Option 相關類別(4hrs)

十一、Swap 類別(3hrs)

十二、Caps/Floors 類別(3hrs)

十三、Swaption 類別(2hrs)

十四、CMS 類別(2hrs)

## Part IV QuantLibXL 與 VBA 的使用(Optional)

十五、使用 VBA 呼叫 QuantLibXL

# Part I 基礎類別

## 主題一 QuantLibXL 的安裝與使用

# 昀騰金融科技股份有限公司

技術長  
金融博士、證券分析師

董夢雲 Andy Dong



ID:50917111

Line/WeChat:andydong3137

E:andydong1209@gmail.com

<https://github.com/andydong1209>

M:(T)0988-065-751 (C)1508-919-2872

10647 台北市大安區辛亥路一段 50 號 4 樓

**現職：**國立台灣大學財務金融研究所兼任教授級專家  
國立台灣科技大學財務金融研究所兼任助理教授  
台灣金融研訓院 2020 年菁英講座

**經歷：**中國信託商業銀行交易室研發科主管  
凱基證券風險管理部主管兼亞洲區風險管理主管  
中華開發金控、工業銀行風險管理處處長  
永豐金控、商業銀行風險管理處處長  
永豐商業銀行結構商品開發部副總經理

**學歷：**國立台灣大學電機工程學系學士  
國立中央大學財務管理學研究所博士

**專業：**證券暨投資分析人員合格(1996)

**專長：**風險管理理論與實務，資本配置與額度規劃、資產負債管理實務  
外匯與利率結構商品評價實務，股權與債權及衍生商品評價實務  
GPU 平行運算與結構商品系統開發，CUDA、OpenCL  
CPU 平行運算與 ALM 系統開發，C#/C++/C、.Net Framework、SQL  
人工智慧(Deep Learning)交易策略開發，Python、Keras、TensorFlow

# 一、QuantLib 程式庫的由來

## ◆ QuantLib 是一個開放源碼軟體程式庫(Open-Source Software Library)

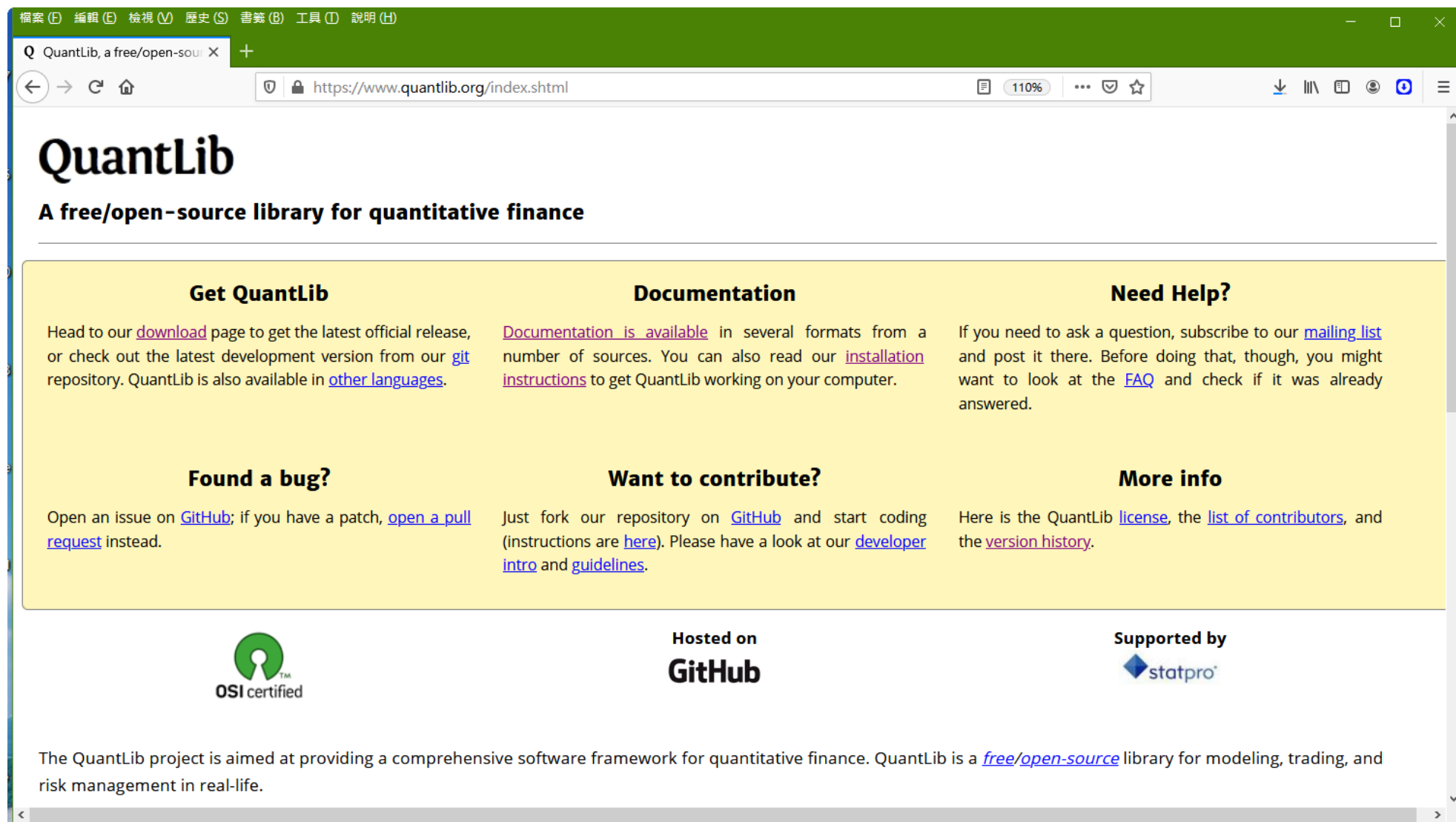
- 其目的在提供對金融工具評價和相關主題有興趣的軟體開發人員，適合的開發工具
- 可在 QuantLib 的網站上，<https://www.quantlib.org>，了解專案的相關內容

## ◆ QuantLib 是以 C++ 語言開發的程式庫，以此為基礎被轉寫成其他不同的語言

- 包括 C#、Objective Caml、Java、Perl、Python、GNU R、Ruby 與 Scheme

## ◆ 在 QuantLibXL 專案中，QuantLib 包裝成 Excel 可以使用的增益集

- QuantLib 不再局限於專業的財務工程師使用，一般的財務人員也可透過 Excel 的使用
- 本課程即是使用此增益集，版本為 1.16.0。





## (一)QuantLib 的歷史

- ◆ 由一群在 Cabota Banca Intesa 的利率衍生商品交易台工作的數量分析專家們，於西元 2000 年時所開創的一個財務金融程式庫開發專案

➤ 目前這些專家們已經成立了一家公司，該公司之前稱之為 RiskMap，現在則命名為 StatPro Italia。

- ◆ QuantLib 專案目前是由 Luigi Ballabio 與 Ferdinando Ametrano 兩位專家所領導

➤ 當他們在實作 Black-Scholes 公式的時候，Ferdinando Ametrano 分享他在另一個結構化專案中的開發經驗。

➤ 該專案允許一種新的合作方式，專案中的任何人可以改進、更正與開發一個免費的共用基礎框架 (Free Common-base Framework)。

✓ 此一意見為 RiskMap 所支持並加以採用。

➤ QuantLib 的創建者秉持著這樣的信念，公開的標準最適合於科學與技術的演進。

✓ 尤其在學術界之內，因為好的實務技能與工具只有被教育界所接受，方可在長期的演化中勝出。

◆ 雖然最早專案的目標對象為學術界與實務界，然而最後卻是進一步促成這兩方人士進一步的交流。

➤ QuantLib 所提供的工具，不論對實務上的實作或是進階的建模，都是相當有用的。

◆ 下表列示了 QuantLib 各版本釋出的時間。目前最新的版本是 1.17 版。除了 C++ 語言之外，其他語言的 QuantLib 專案也陸續成立運作，

➤ 例如使用 C# 語言的 QLNet，<http://sourceforge.net/projects/qlnet/>；

➤ 使用 R 語言的 RQuantLib，<http://dirk.eddelbuettel.com/code/rquantlib.html>；

➤ 使用 Java 語言的 JQuantLib，<http://www.jquantlib.org>；

➤ 以及可配合 Excel 使用的增益集，本專案的子專案 QuantLibXL。

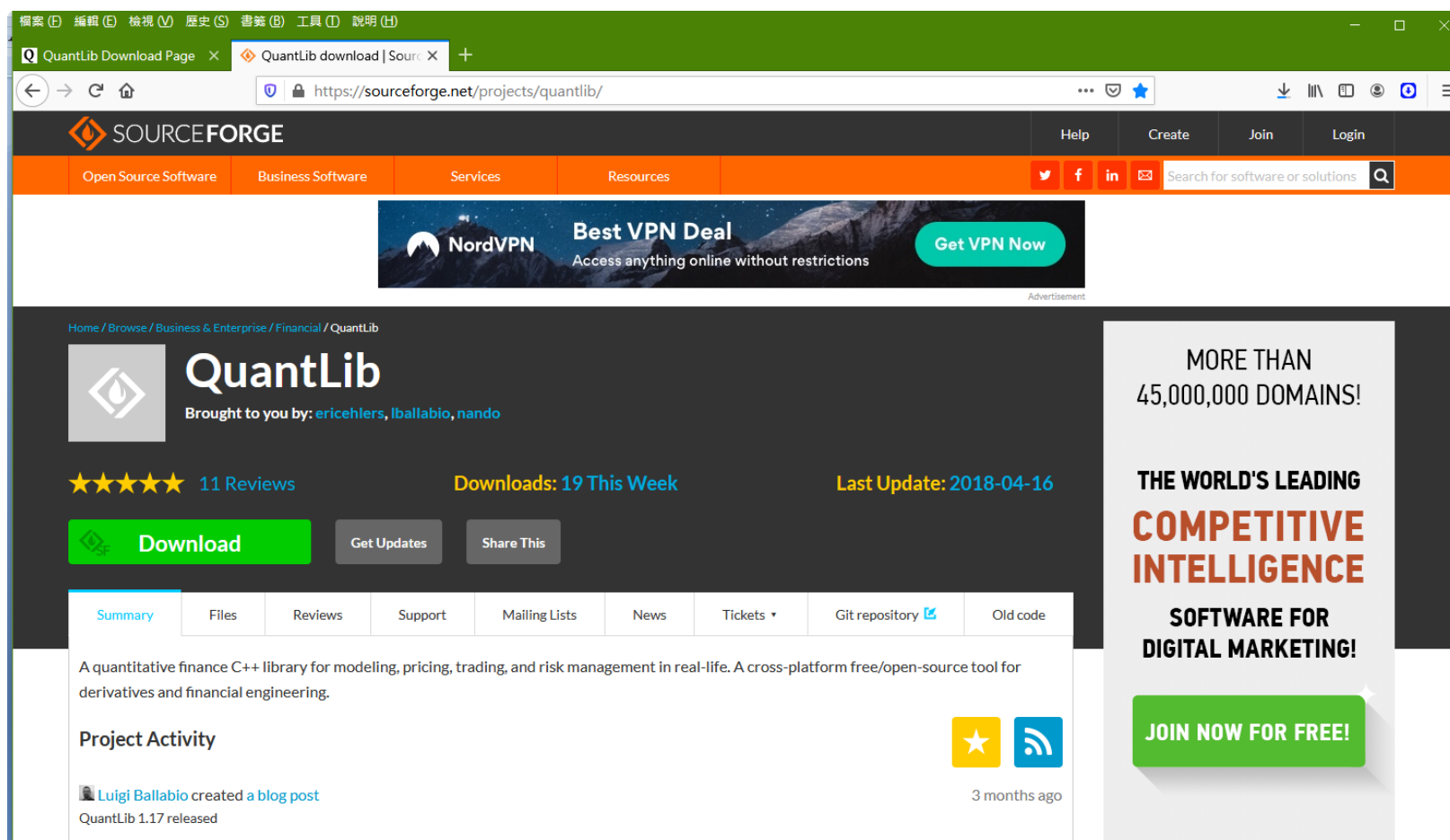
## ◆ QuantLib 各版本的演進與推出時間表

Version	Release date	Notes
0.1.1	Nov 21, 2000	
0.2.0	Sep 18, 2001	
0.3.4	Nov 21, 2003.	
0.3.7	Jul 23, 2004.	此版之後 QuantLib 需要 <a href="#">Boost</a> 程式庫。
0.4.0	Feb 20, 2007.	
0.8.0	May 30, 2007.	版本的大幅更動，目的在加快收斂到 1.0 版。
0.9.0	Dec 24, 2007.	
0.9.9	Nov 2009.	
1.0.0	Feb 24, 2010	
1.0.1	Sep 17, 2010	
1.1.0	May 23, 2011	
1.2.0	March 6, 2012	
1.2.1	Sep 10, 2012	
...	...	
1.16	Aug 5, 2019	
1.17	Dec, 2019	

## (二)QuantLib 程式庫的下載

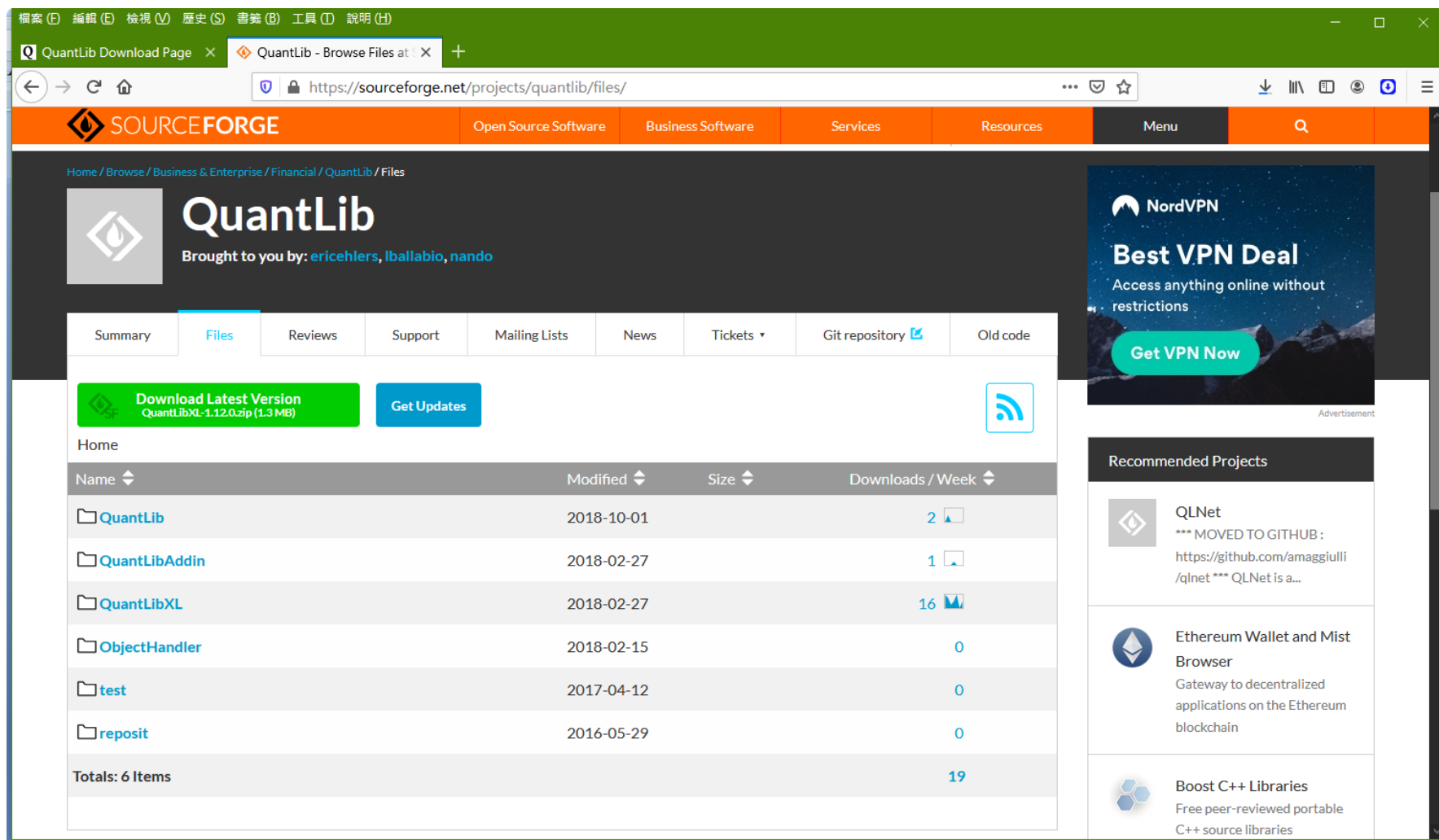
◆ 可在 Open Source Software 網站內的 QuantLib 專案網頁下載，

➤ <https://sourceforge.net/projects/quantlib/>



➤ 點選網頁上 Files，便可進入下面網頁

✓ 此處只有更新到 1.12.0 板，點選 QuantLib 進入下載。



SourceForge

Home / Browse / Business & Enterprise / Financial / QuantLib / Files

# QuantLib

Brought to you by: ericehlers, lballabio, nando

Summary **Files** Reviews Support Mailing Lists News Tickets ▾ Git repository [🔗](#) Old code

[Download Latest Version](#)  
QuantLibXL-1.12.0.zip (1.3 MB)

[Get Updates](#)

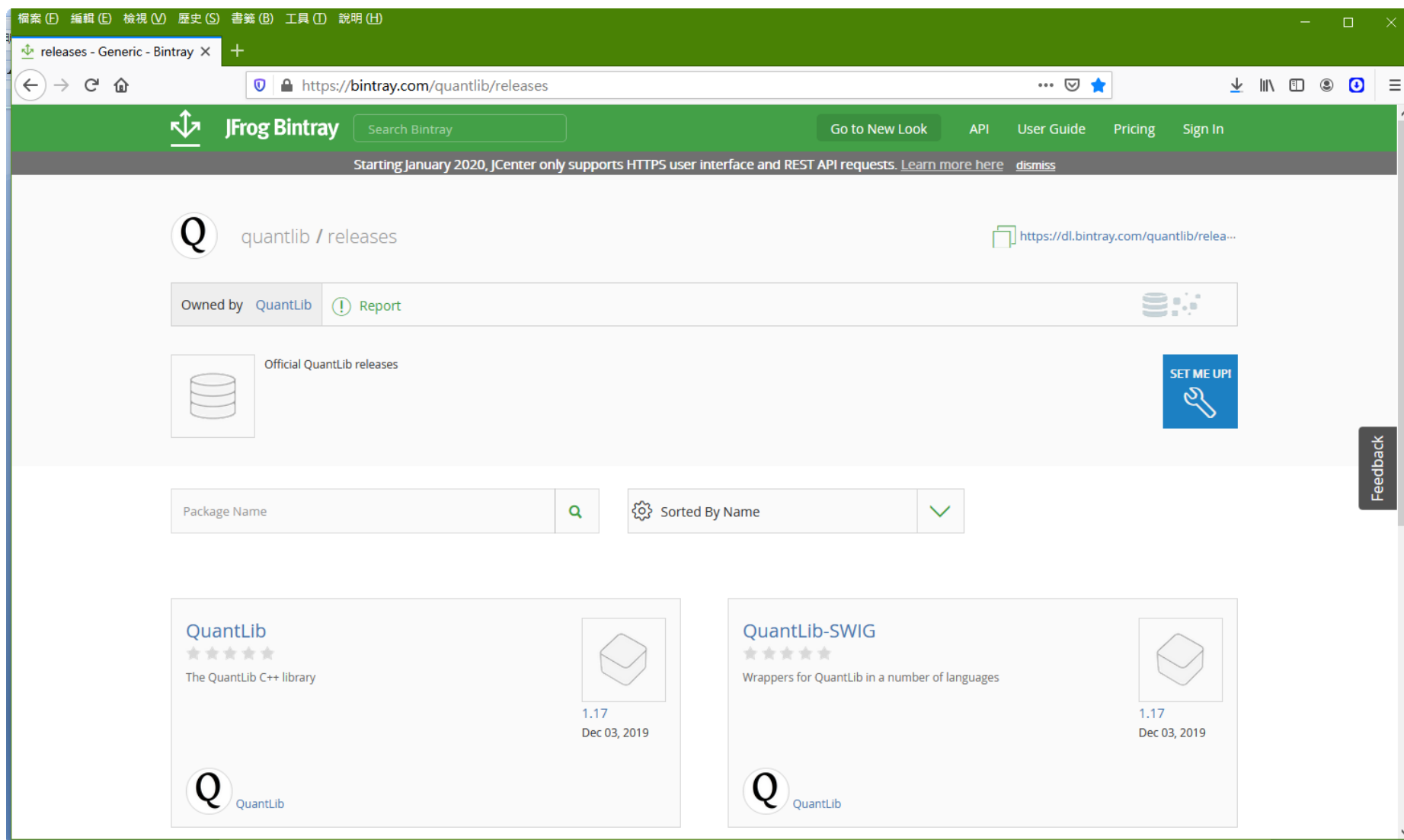
Home

Name	Modified	Size	Downloads / Week
QuantLib	2018-10-01		2
QuantLibAddin	2018-02-27		1
QuantLibXL	2018-02-27		16
ObjectHandler	2018-02-15		0
test	2017-04-12		0
reposit	2016-05-29		0
<b>Totals: 6 Items</b>			<b>19</b>

Recommended Projects

- QLNet**  
\*\*\* MOVED TO GITHUB :  
<https://github.com/amaggiulli/qlnet> \*\*\* QLNet is a...
- Ethereum Wallet and Mist Browser**  
Gateway to decentralized applications on the Ethereum blockchain
- Boost C++ Libraries**  
Free peer-reviewed portable C++ source libraries

◆ 另一個最新的下載點，<https://bintray.com/quantlib/releases>，



➤ 點選 QuantLib 進入下載。

The screenshot shows a web browser window with the address bar displaying `https://bintray.com/quantlib/releases/QuantLib`. The page has a green header with navigation tabs: General, Readme, Release Notes, Reviews (0), Statistics, and Files. The 'General' tab is active.

**About This Package** [Version Notification Links](#)

Website	<a href="http://www.quantlib.org">http://www.quantlib.org</a>
Issue Tracker	<a href="https://github.com/lballabio/QuantLib/issues">https://github.com/lballabio/QuantLib/issues</a>
VCS	<a href="https://github.com/lballabio/QuantLib">https://github.com/lballabio/QuantLib</a>
Product	None
Maturity	Official
Licenses	BSD 3-Clause

**Downloads (ver: 1.17)**

File	Size	sha256
<a href="#">QuantLib-1.17.tar.gz</a>	8.55 MB	77443c3f9e9da3130ce0b2bce739928c14089f70d1e8abc7dfad0adb40a84fb
<a href="#">QuantLib-1.17.zip</a>	11.02 MB	ac5e0f3915b353c00c420649ed14baf82e0e147b81f1b32070f3c3e2942a919d

**Versions** [Latest Version Badge](#)

Version	Latest Version Badge
1.17	
1.16	
1.15	
1.14	
1.13	

[View All](#)

**Watchers (0)** [Watch](#)

This package is not being watched by anyone yet.

**Linked to (0)**

This package is not linked to any repository yet.

◆ QuantLib 的 Wiki 網頁為，<http://en.wikipedia.org/wiki/QuantLib>。

W QuantLib - Wikipedia

← → ↻ ↩ 🔒 https://en.wikipedia.org/wiki/QuantLib 120% ☆

Not logged in Talk Contributions Create account Log in

Article **Talk** Read Edit View history Search Wikipedia

# QuantLib

From Wikipedia, the free encyclopedia

**This article has multiple issues.** Please help **improve it** or discuss these issues on the **talk page**. [hide]

*(Learn how and when to remove these template messages)*

- The topic of this article **may not meet Wikipedia's notability guidelines for products and services**. *(January 2014)*
- This article includes a **list of references**, related reading or **external links**, **but its sources remain unclear because it lacks inline citations**. *(March 2010)*
- This article **relies too much on references to primary sources**. *(January 2017)*

**QuantLib** is an open-source software library which provides tools for software developers and practitioners interested in **financial instrument** valuation and related subjects. QuantLib is written in **C++**.

**Contents** [hide]

- History
  - Release History
- Usage
- Licensing
- Features
- See also
- External links

## QuantLib

QuantLib logo (Fontin Bold font)

<b>Developer(s)</b>	QuantLib Team
<b>Stable release</b>	1.8 / May 18, 2016; 3 years ago
<b>Repository</b>	<a href="https://github.com/lballabio/quantlib">github.com/lballabio/quantlib</a>
<b>Written in</b>	C++
<b>Type</b>	Numerical library
<b>License</b>	modified BSD license

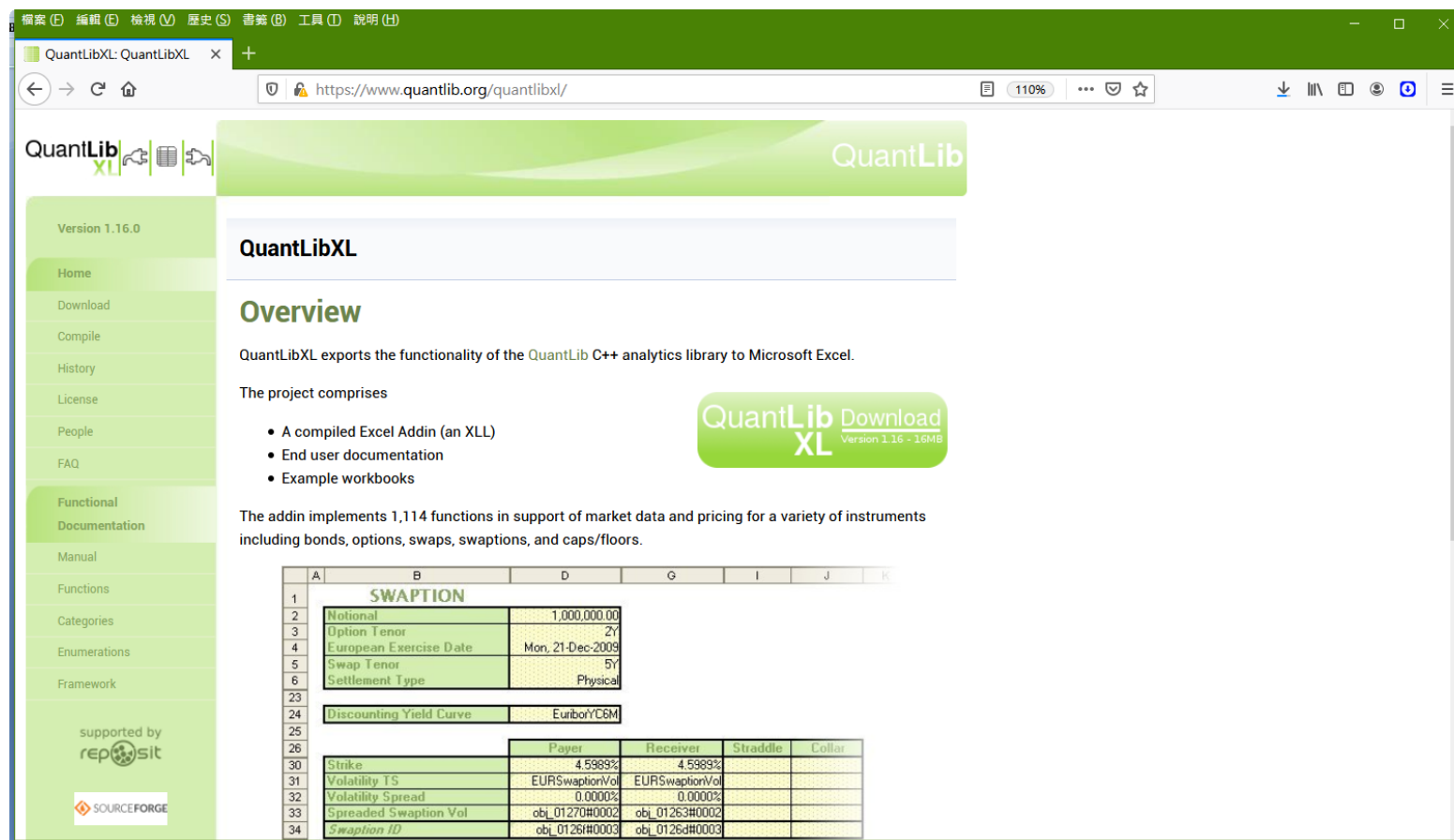
History [edit]



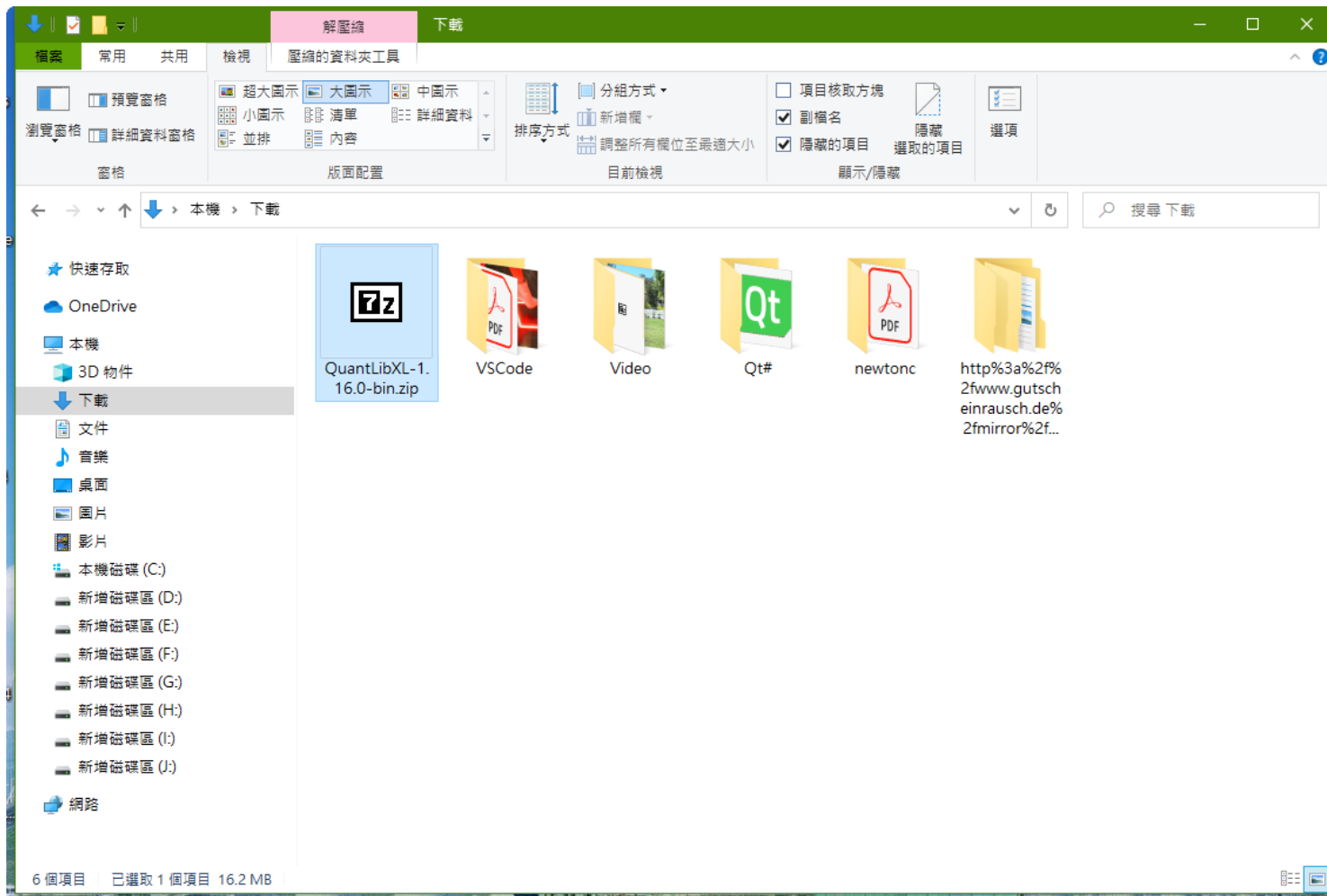
## 二、QuantLibXL 增益集的安裝

◆ 進入如下圖的網頁，<http://quantlib.org/quantlibxl/>，

➤ 按右上方 QuantLibXL Download 的按鈕，最新版 1.16 版。

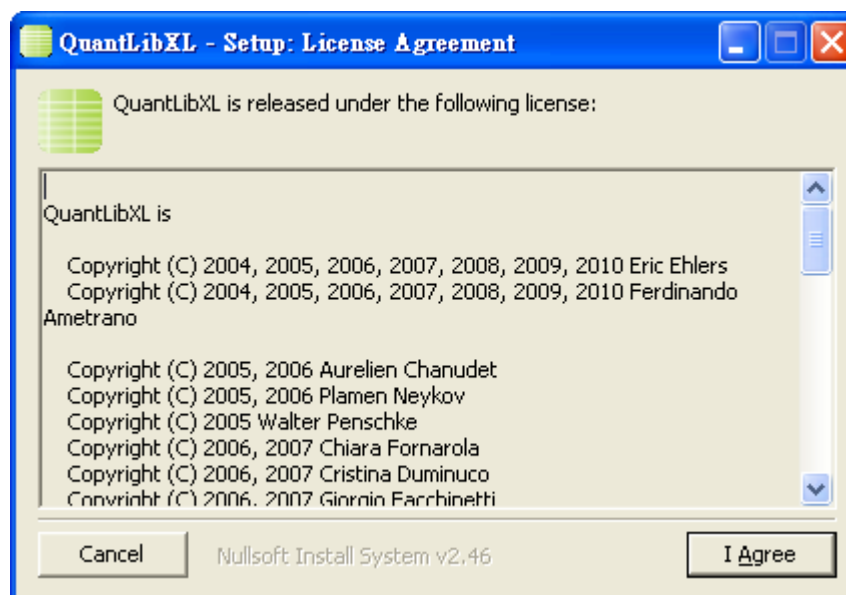


► 自動下載到電腦下載區內。



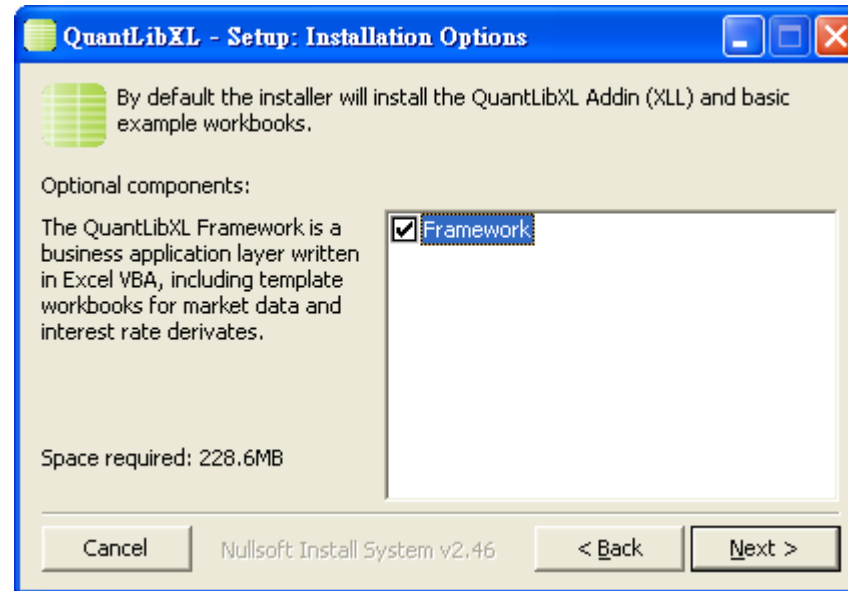
◆ 本書中使用的是 1.16 版，下面以 1.2.0 版本為例，說明安裝的過程。

➤ 點選 QuantLibXL-bin-1.2.0.exe 執行之，出現 QuantLibXL 版權協議說明，按同意按鈕，



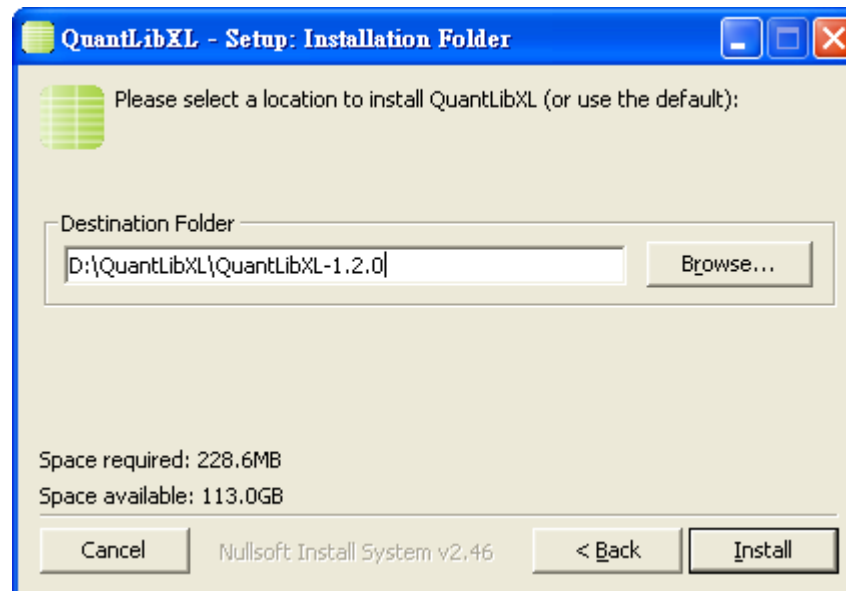
➤ 出現 QuantLibXL 安裝選項。

- ✓ 此選項是詢問讀者是否要安裝 QuantLibXL Framework，它包含一些針對市場資料與利率衍生商品的樣板試算表，是以 Excel VBA 所撰寫的。讀者可以選取以安裝之，按 Next 往下執行。

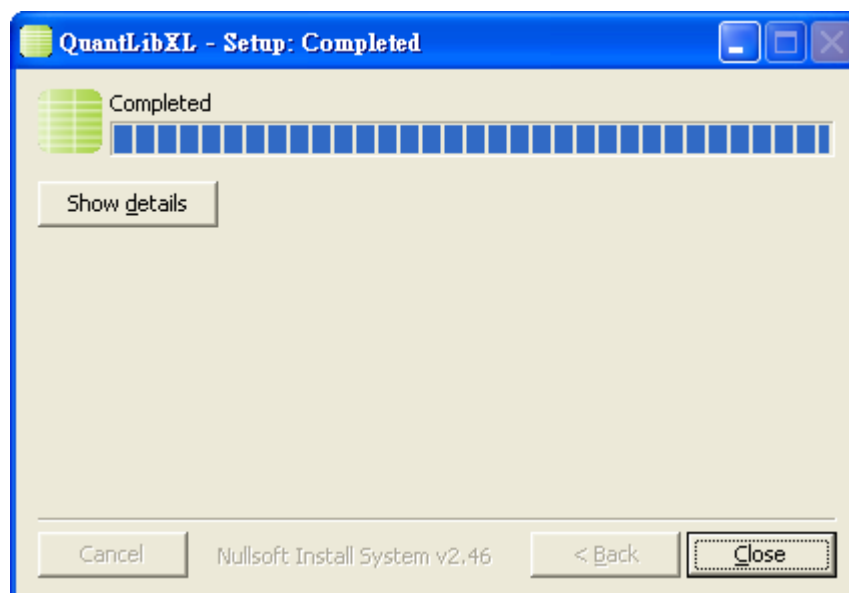


➤ 接著出現 QuantLibXL 安裝路徑設定。

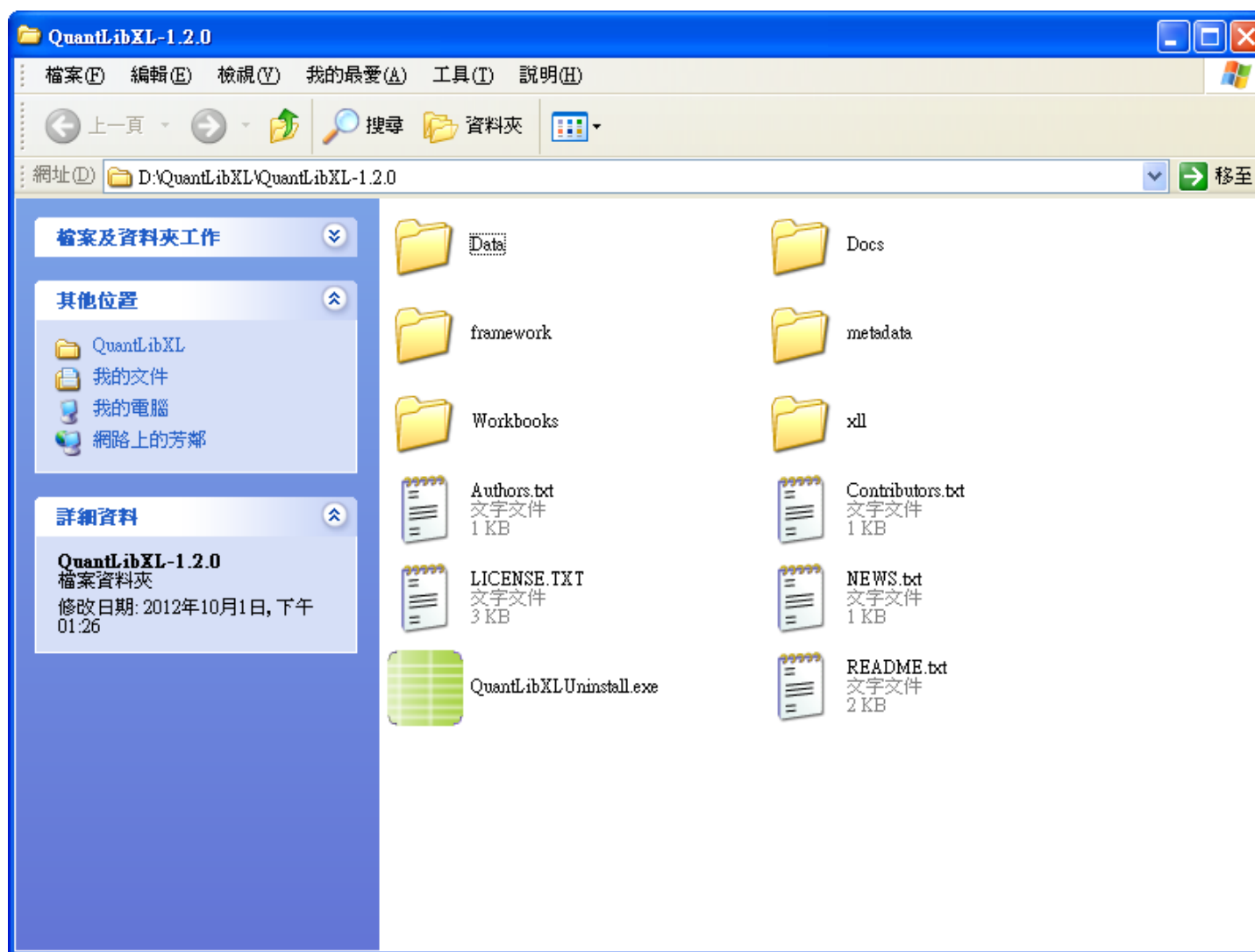
- ✓ 讀者可自行設定到想要的地方，作者將舊版設定到 D:\QuantLibXL\QuantLibXL-1.2.0。
- ✓ 作者將 1.16.0 版設定到 C:\QuantLibXL16。
- ✓ 按安裝按鈕，便會開始安裝之。



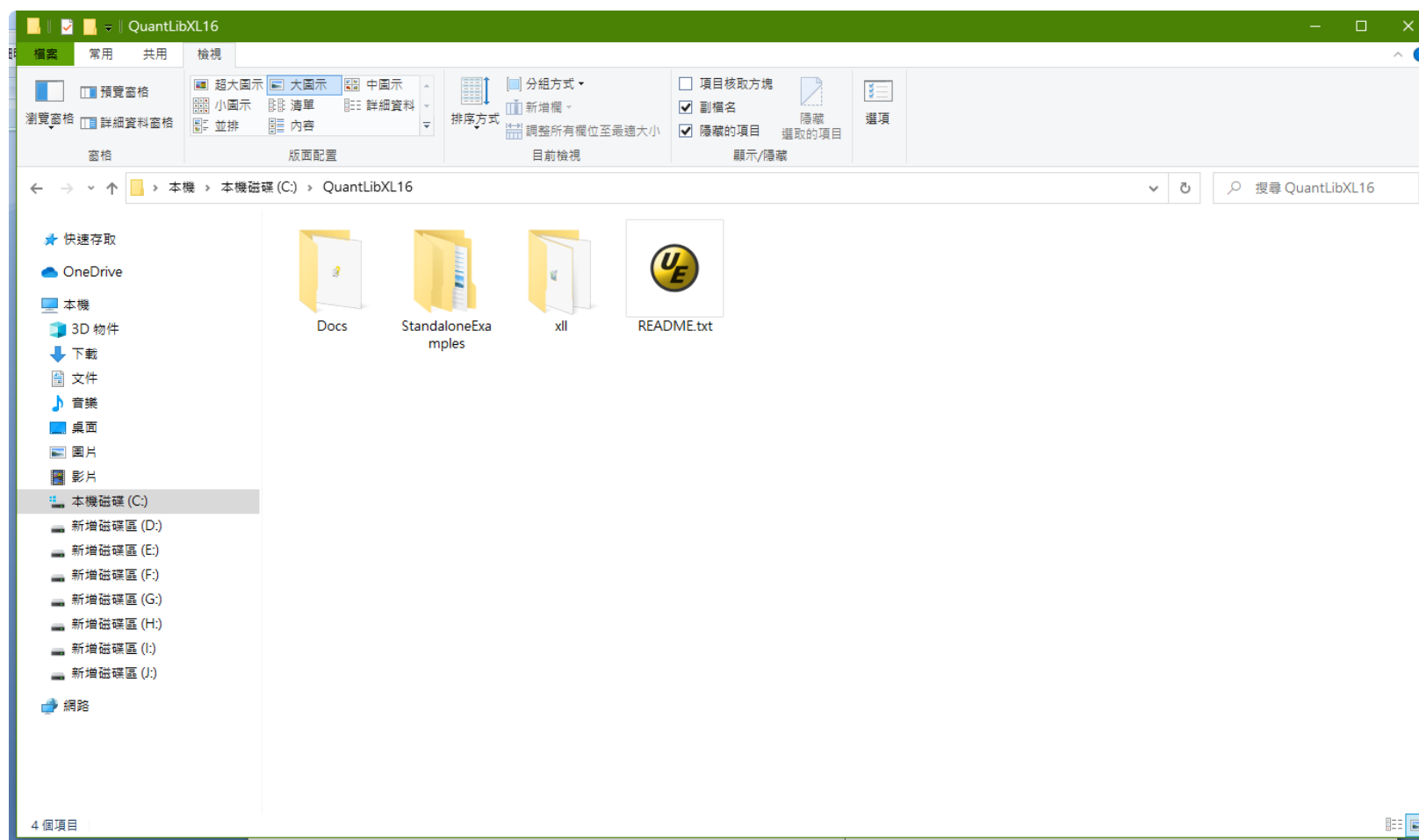
► 安裝完成畫面



## ◆ 舊版安裝的目錄與檔案



► 新版 1.16.0 的安裝目錄。

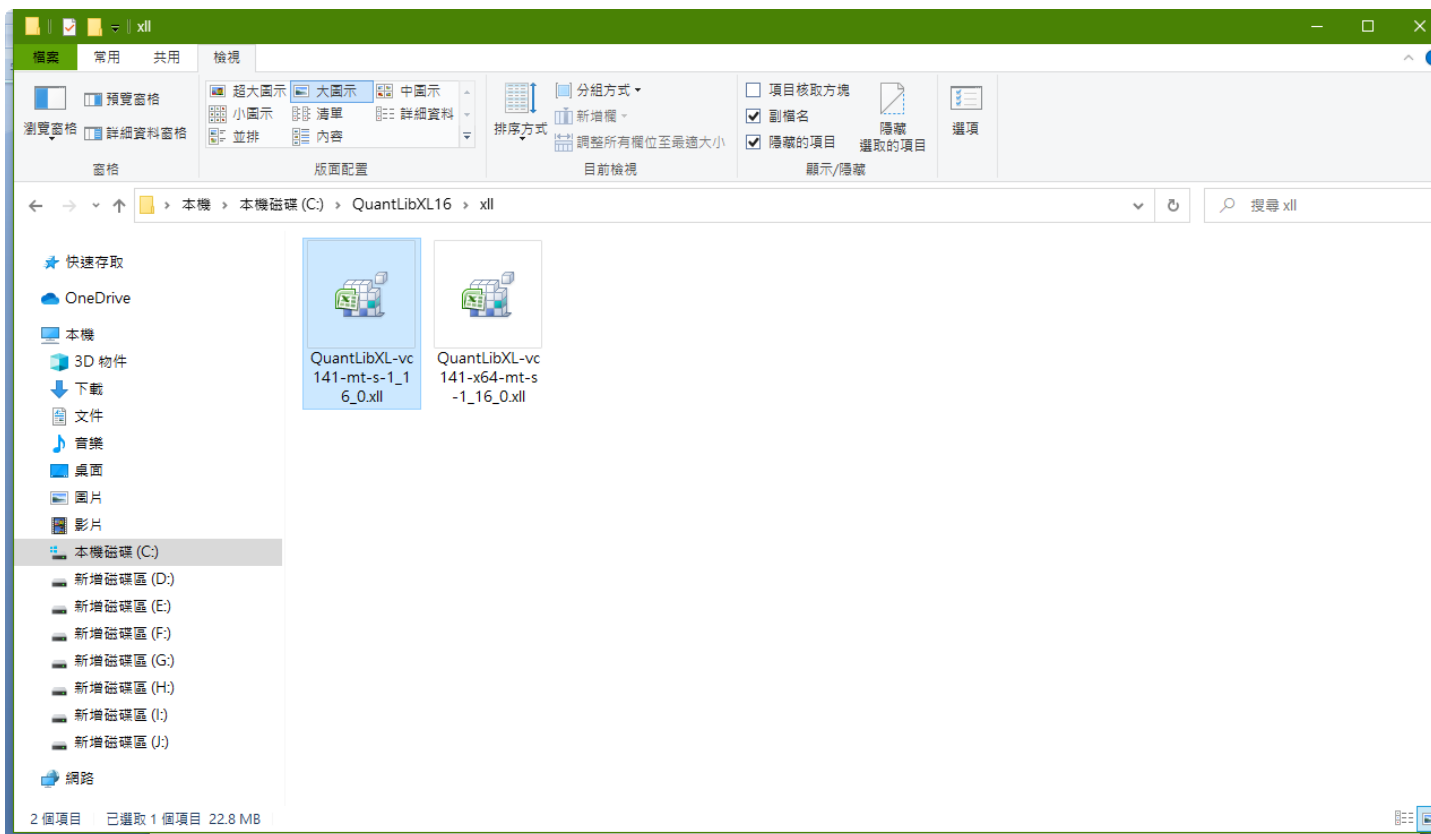




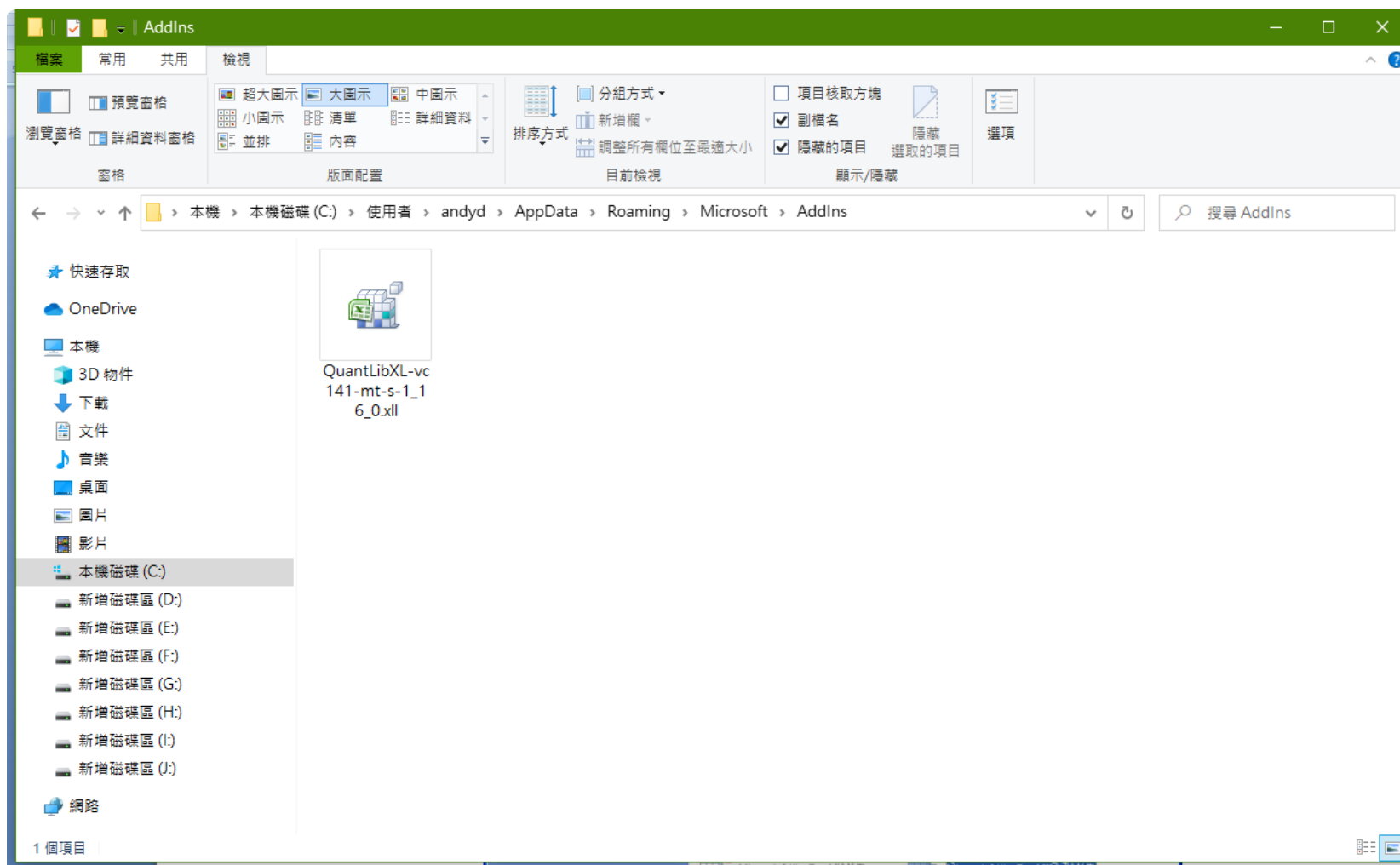
### 三、QuantLibXL 的使用測試

◆ 首先，將增益集檔案，QuantLibXL-vc141-mt-s-1\_16\_0.xll，拷貝到正確的位置。

➤ 增益集在路徑 C:\QuantLibXL16\xll 下面

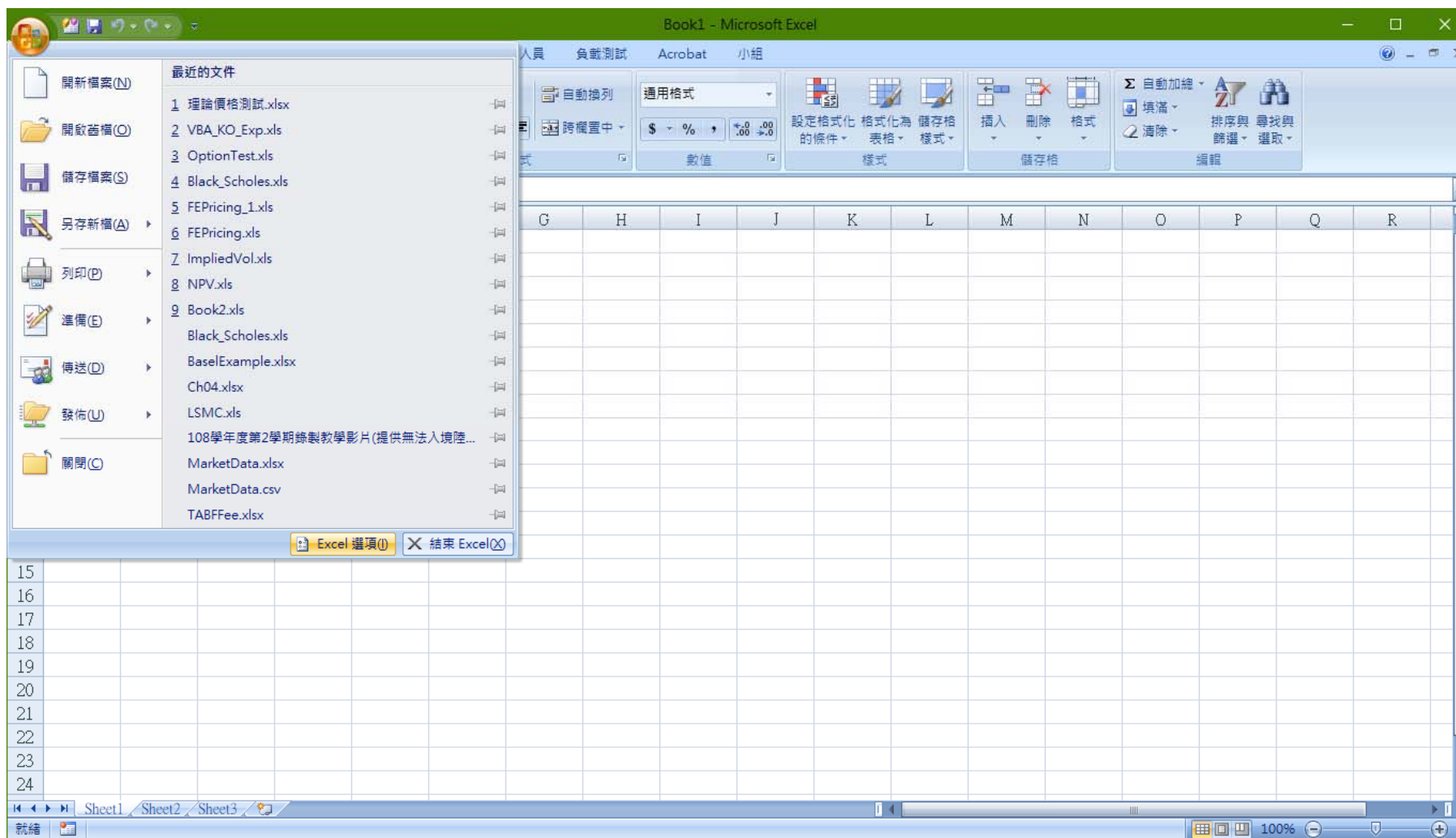


- 在 Windows 10 作業系統下，拷貝到 Excel 增益集的預設目錄下，C:\Users\andydong\AppData\Roaming\Microsoft\AddIns。

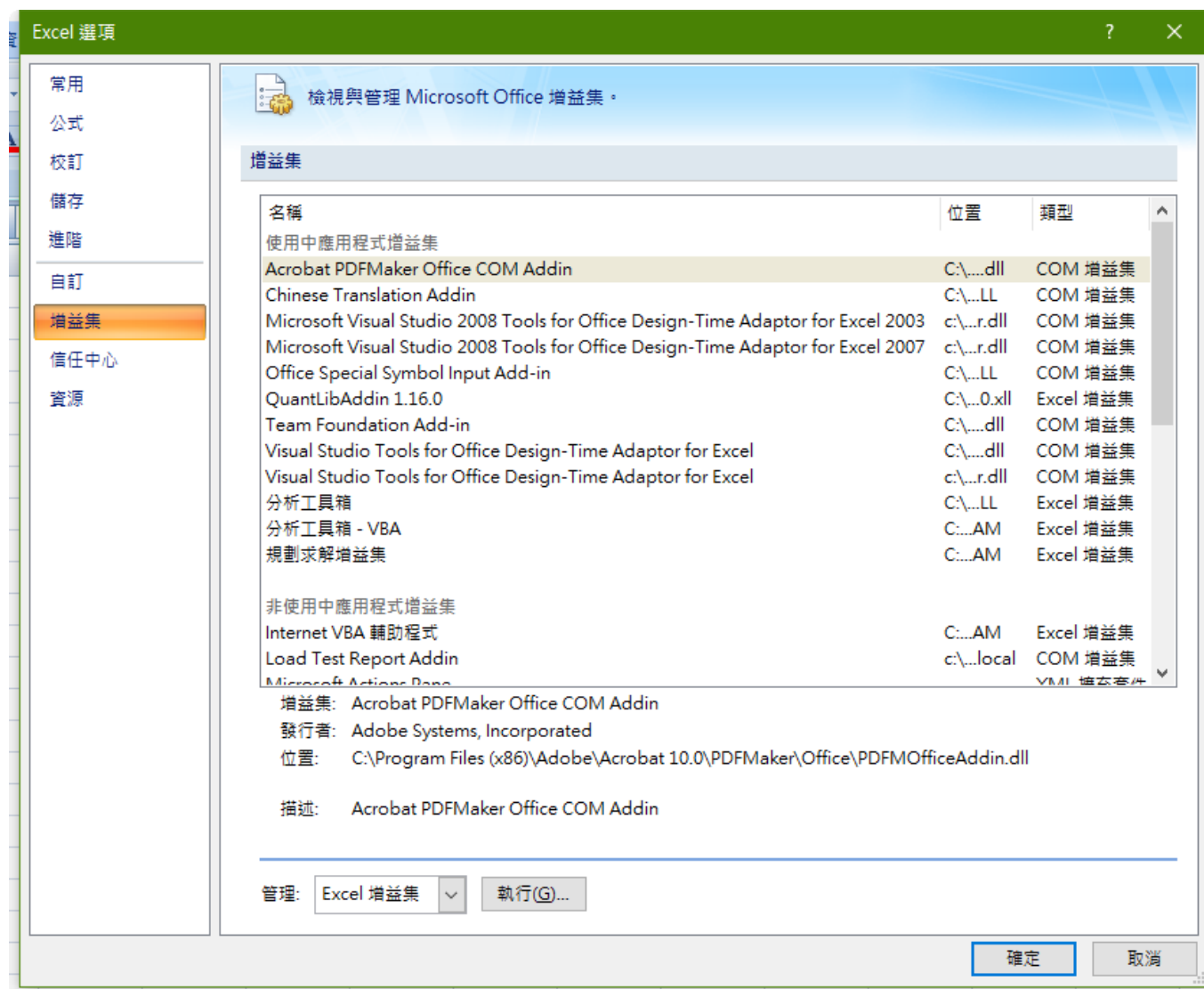


## ◆ 其次，開啟增益集。

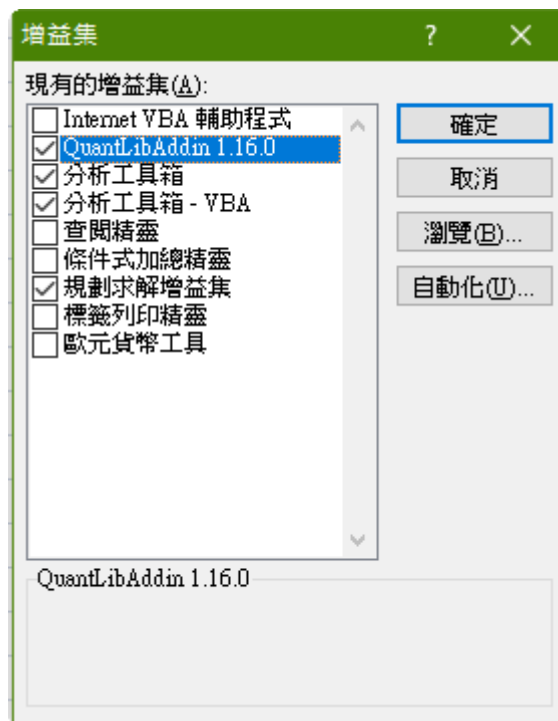
➤ 打開 Excel，由樞紐，開啟 Excel 選項。



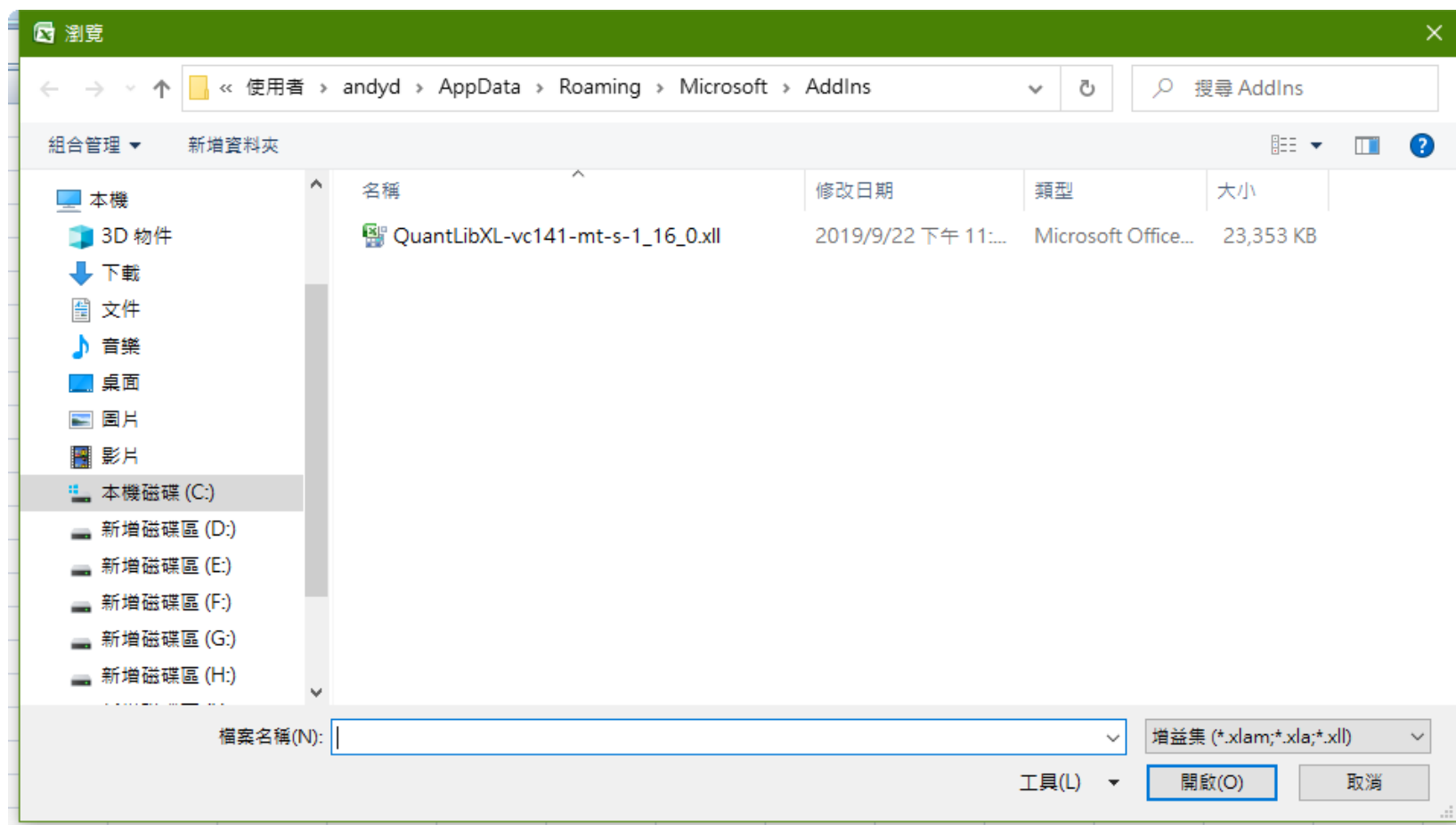
➤ 在增益集選項下，按下方執行按鈕。



► 可以看到 QuantLibAddin 1.16.0 已經出現在選項之中，勾選之即可。



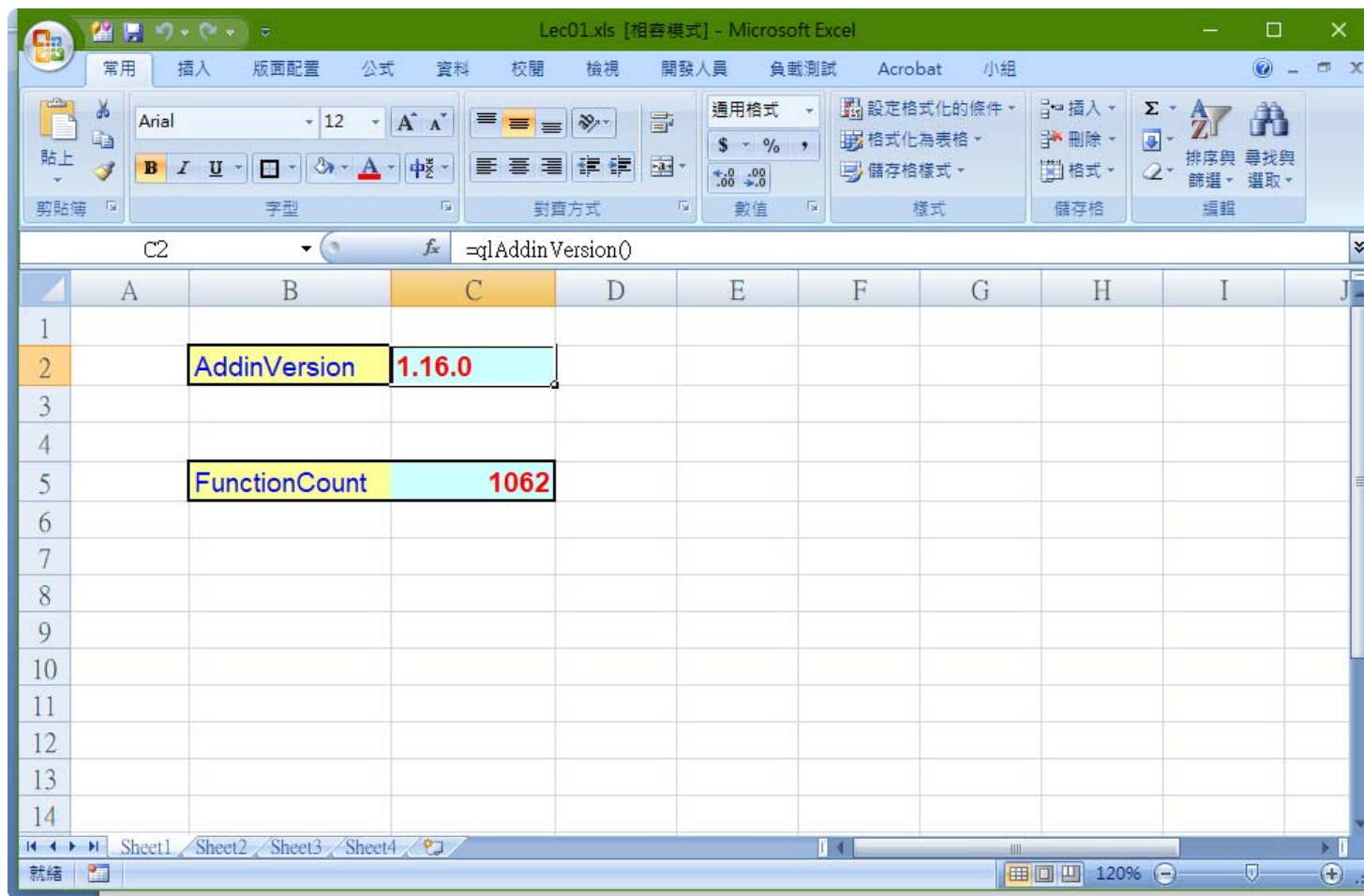
- 如果 QuantLibXL 增益集沒有放在預設的目錄下，則讀者必須使用《瀏覽》按鈕，找尋所放置的檔案位置。



## ◆ 增益集版本函數

► 增益集版本的函數，`qlAddinVersion()`，可以傳回我們安裝的增益集之版本資訊。

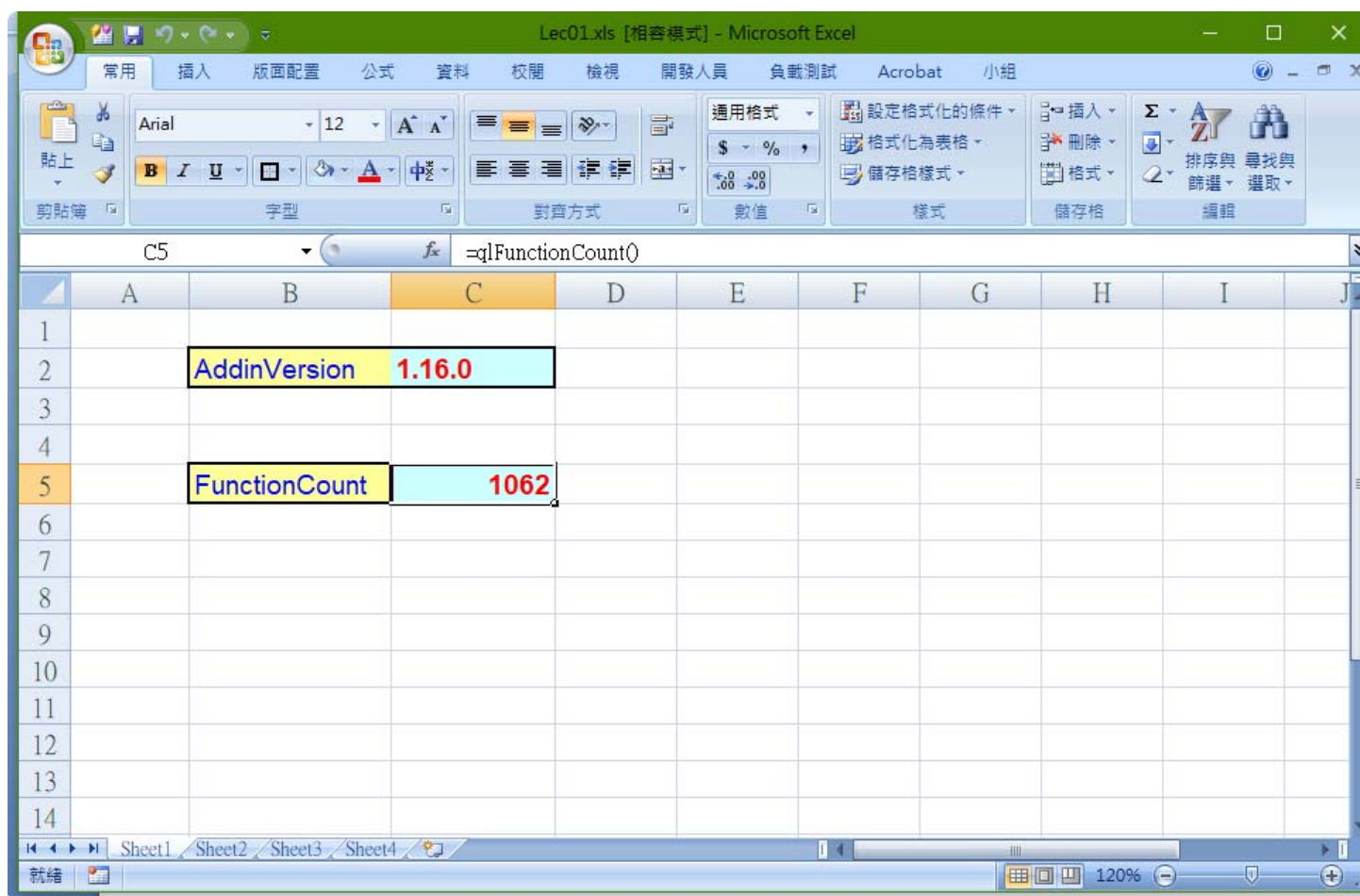
✓ 在 B3 格輸入，`=qlAddinVersion()`，便可傳回，1.16.0。



## ◆ 增益集中函數數量函數

► 計算增益集中函數數量的函數，qlFunctionCount()，可傳回增益集中函數數量

✓ 在 C5 格輸入，=qlFunctionCount()，便可傳回，1062。在 1.0.1 版本中，提供了 898 個函數。





## 四、QuantLibXL 的運作邏輯

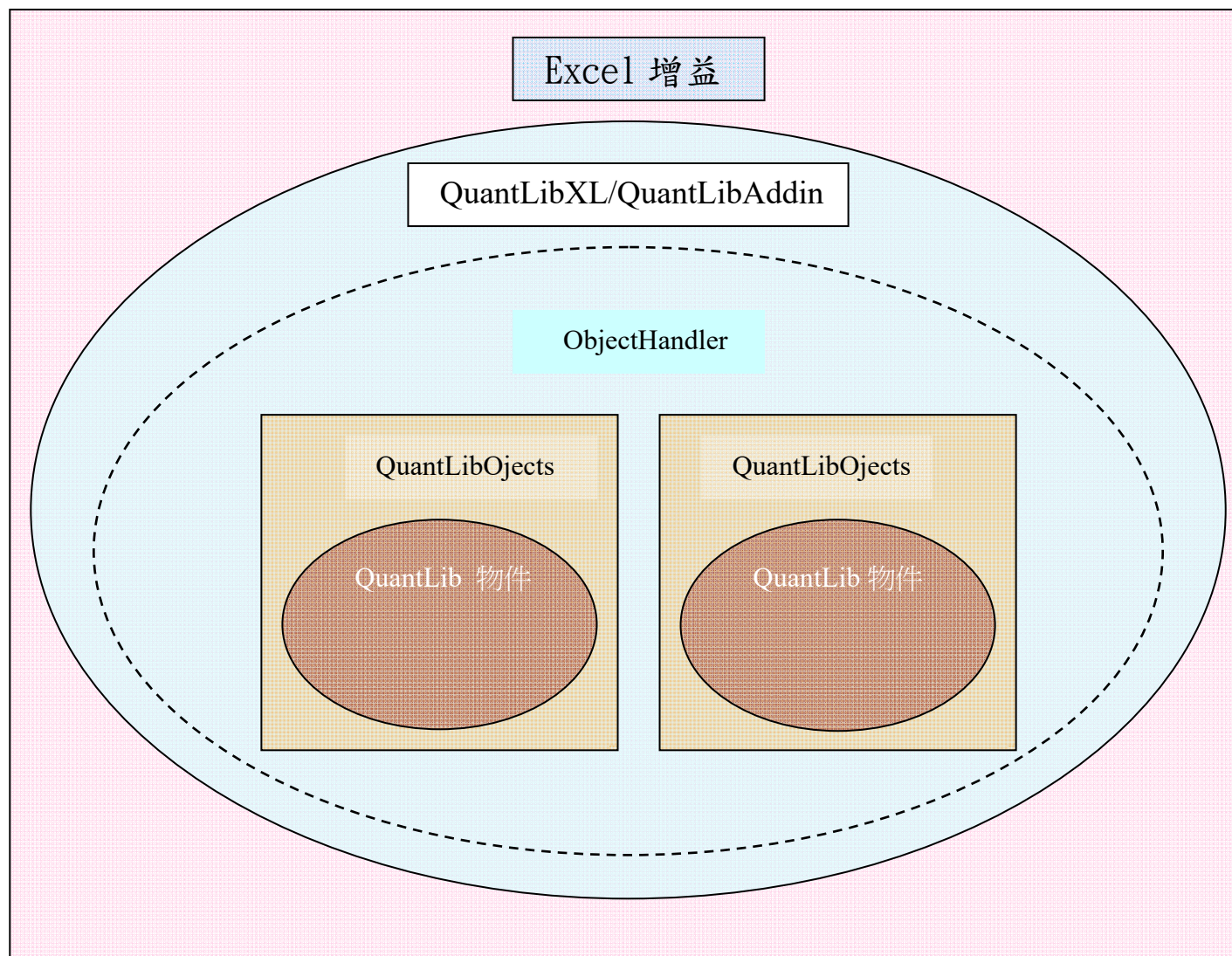
### ◆ QuantLib 程式庫是一個靜態類別庫(Static Library Of Classes) ,

- 在給終端使用者使用之前，它必須被包覆在一個應用層之內。
  - ✓ 此應用層可以是可執行檔、試算表增益集、或網路伺服程式等。
  - ✓ 應用層的功能在於提供一個儲藏 QuantLib 物件的機置，一旦這些物件被產生之後，便可以儲藏這些被產生的 QuantLib 物件。
- 為了達成這一目的，QuantLib 開發團對成立了另外的專案。
  - ✓ 包括了 QuantLibAddin 與 QuantLibXL 這兩個專案。
  - ✓ QuantLibAddin 為一個介面的專案，其目的在讓 QuantLib 程式庫能夠使用在試算表的環境中，而試算表是市場上最為終端使用者所接受的環境。
  - ✓ QuantLibXL 則是 QuantLibAddin 在 Microsoft Excel 上的一個實作產品。
- 由於試算表本身沒有提供儲藏 QuantLib 物件機置的功能，我們只能在增益集中增加這些機置。

## ◆ QuantLibAddin 使用 ObjectHandler 作為他的物件儲藏所。

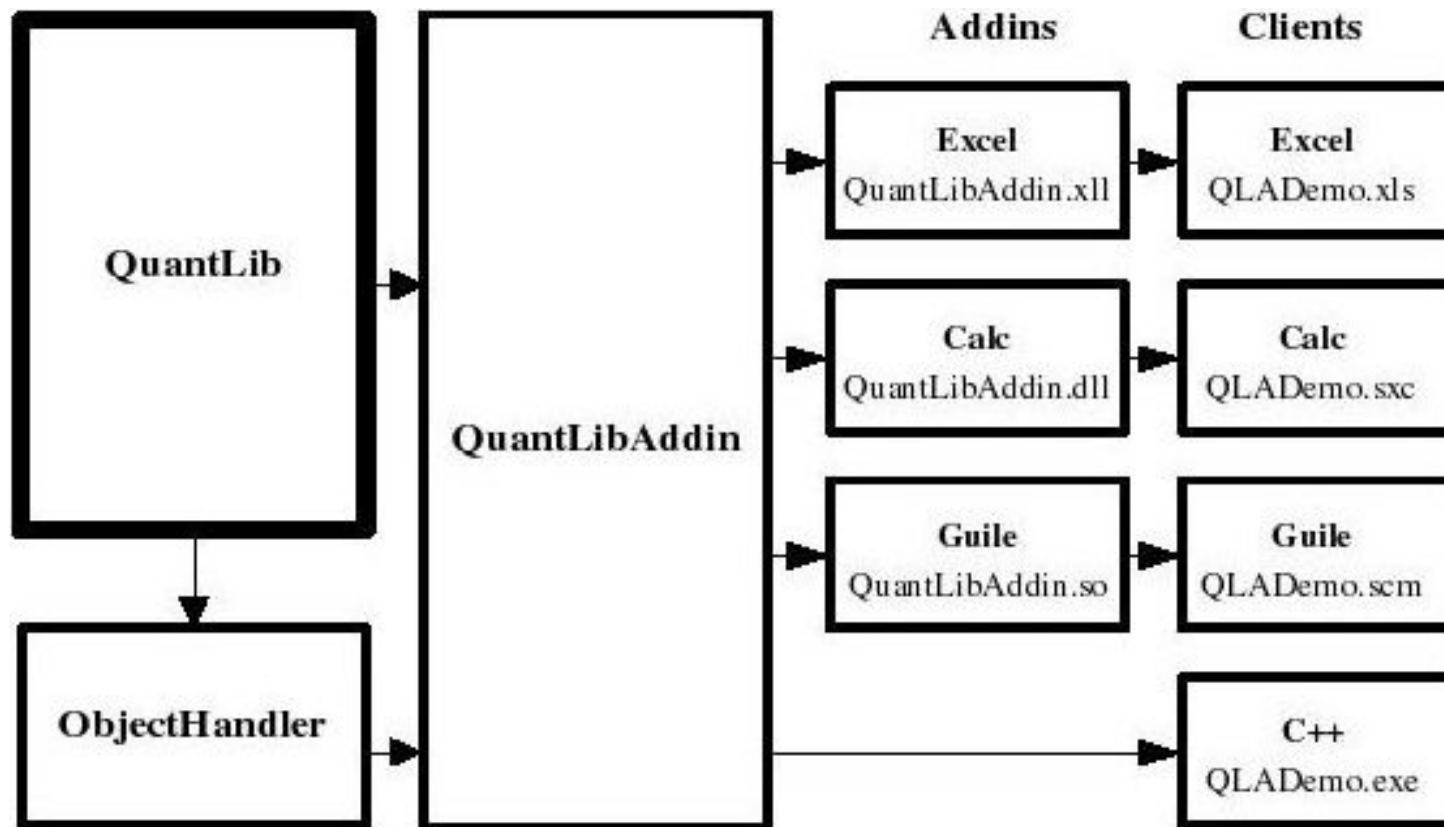
- QuantLibAddin 的核心就是一個被稱為 QuantLibObjects 的程式庫。
  - ✓ QuantLibObjects 會包覆 QuantLib 物件，使其可以被儲存在 ObjectHandler 的儲藏所。
  - ✓ QuantLibAddin 實作一組介面，透過這組介面，終端使用者可以使用 QuantLib 的物件。
- 當終端使用者在試算表上的儲存格內輸入=qlPiecewiseYieldCurve()後，
  - ✓ 一個 QuantLib Yield Curve 物件便被產生出來，此物件儲存在 ObjectHandler 的儲藏所內。
  - ✓ 同時，一個指向此物件的參考(Reference)也被傳回到此一儲存格之內。
- QuantLibAddin 提供這組介面給許多不同終端使用者的平台，
  - ✓ 包括 Microsoft Excel 與 OpenOffice.org Calc。

## ◆ QuantLibXL 運作示意圖



◆ QuantLib 程式庫透過 ObjectHandler 被 QuantLibAddin 使用。

- 終端使用者 Clients 藉由 Excel 的試算表，透過 QuantLibXL 來使用 QuantLibAddin 的功能，進一步呼叫到 QuantLib 的物件功能。



## 五、QuantLibXL 的使用說明

### (一)物件導向的使用邏輯

◆ 在 QuantLibXL 增益集中，一些簡單的功能直接呼叫函數便可。

➤ 如 QuantLibXL 版本函數 `qlAddinVersion()`，以及 QuantLibXL 函數數量的函數 `qlFunctionCount()`。

◆ 但是有些函數在使用前，必需先產生特定的物件(Object)，然後才可以使用。

➤ 以程式設計的說法是，這些函數是特定物件的方法(Method)，當這些物件實體化後，這些依附在物件的方法才可以被使用。

✓ 而物件在實體化的過程中，相關的資料(Property)也會被初始化，如此在呼叫方法時，才有東西可以執行。

◆ 我們使用一個亂數產生器物件，來說明相關的邏輯。

➤ 在 QuantLibXL 中有一個產生亂數產生器物件的函數 `qlMersenneTwisterRsg()`，呼叫此函數便可以產生一個亂數產生器物件。

◆ 參考 QuantLibXL 的文件，可以得知 qlMersenneTwisterRsg()的輸入與輸出格式如下，

### qlMersenneTwisterRsg

```
string returnValue  
qlMersenneTwisterRsg(  
    string ObjectId  
    long Dimension  
    long Seed  
    bool Permanent  
    any Trigger  
    bool Overwrite)
```

### Description:

Construct an object of class MersenneTwisterRsg and return its id

### Supported Platforms:

C++, Excel

### Parameters:

<i>ObjectId</i>	id of object to be created
<i>Dimension</i>	dimension.
<i>Seed</i>	seed.
<i>Permanent</i>	object permanent/nonpermanent
<i>Trigger</i>	dependency tracking trigger
<i>Overwrite</i>	overwrite flag

■ qlMersenneTwisterRsg()：產生一個 Mersenne Twister 的亂數產生器

- ✧ 傳回值型態：字串(string，產生之物件的代碼)
- ✧ 傳入參數一：字串(string，產生之物件希望的名稱)，MTRsgObj
- ✧ 傳入參數二：長整數(long，亂數的維度)，1
- ✧ 傳入參數三：長整數(long，亂數的種子)，1234
- ✧ 傳入參數四：布爾代數(bool，物件是否為永久的)，FALSE
- ✧ 傳入參數五：任意資料(any，觸發控制)，A1
- ✧ 傳入參數六：布爾代數(bool，物件是否為覆蓋的)，TRUE

➤ 在 C3 格中輸入，=qlMersenneTwisterRsg(C4, C5, C6, C7, C8, C9)，按下 Enter 鍵，便傳回 MTRsgObj#0000。

- ✓ 代表我們已經成功在增益集中產生一個叫作 MTRsgObj 的物件，其編號為#0000。
- ✓ 此物件為一維亂數產生器，以 1234 為其亂數種子。
- ✓ 它不是永久的，是覆蓋的，後面再解釋這兩個輸入的用意。



Lec01.xls [相容模式] - Microsoft Excel

常用 插入 版面配置 公式 資料 校閱 檢視 開發人員 負載測試 Acrobat 小組

貼上 剪貼簿 字型 對齊方式 數值 樣式 儲存格 編輯

C3 =qlMersenneTwisterRsg(C4,C5,C6,C7,C8,C9)

	A	B	C	D	E	F	G
1	0						
2							
3		<b>MersenneTwisterRsg</b>	<b>MTRsgObj#0001</b>			<b>Variates</b>	<b>Output</b>
4	string	ObjectId	MTRsgObj		string	RsgObjectId	MTRsgObj#00
5	long	Dimmension	1		long	Samples	
6	long	Seed	4321		any	Trigger	MTRsgObj#00
7	bool	Permenanent	FALSE				
8	any	Trigger	0				
9	bool	Overwrite	TRUE				
10							

Sheet1 Sheet2 Sheet3 Sheet4

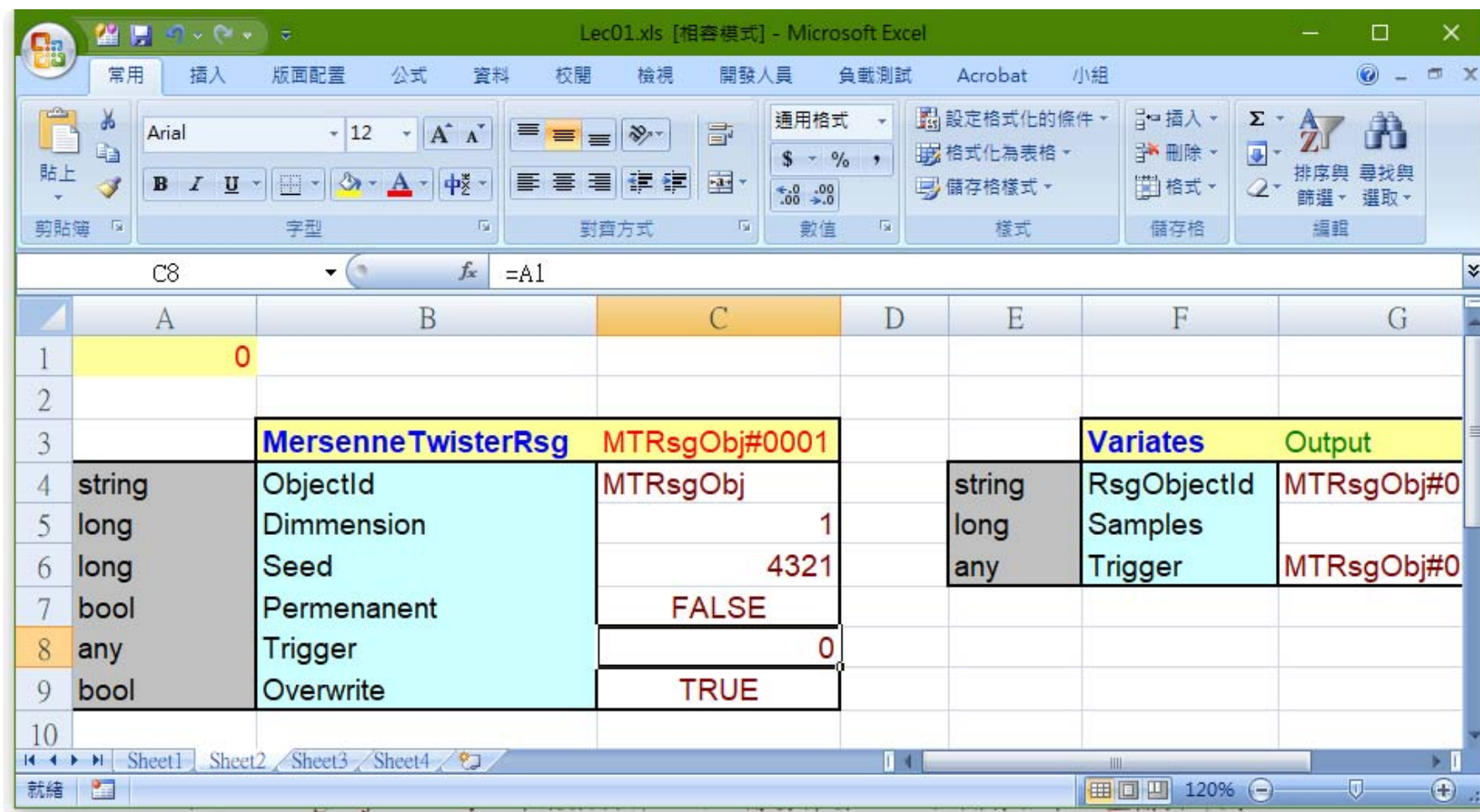
就緒 120%



► 在 C8 中我們以 A1 格作為觸發控制鍵，

✓ 一但我們更改 A1 的數值，你會發現 C3 格會重新產生一個物件，MTRsg#0001。

✓ 由其編號#0001 可知，它是另一個物件。



◆ 在 QuantLibXL 中有一個產生亂數的函數 `qlVariates()`，呼叫此函數便可以產生亂數資訊。

➤ 參考 QuantLibXL 的文件，可以得知此函數的輸入與輸出格式如下，

■ `qlVariates()`：產生亂數資料陣列

- ✧ 傳回值型態：陣列(`vector<vector<double>>`，產生之亂數資料)， $10 \times 1$
- ✧ 傳入參數一：字串(使用的亂數產生器物件名稱)，`MTRsgObj#0000`
- ✧ 傳入參數二：長整數(亂數的數目)，10
- ✧ 傳入參數三：任意資料(觸發控制)，A1

➤ 此函數需要輸入一個亂數產生器，我們使用之前產生的 `MersenneTwister` 物件作為輸入。

- ✓ 我們需要告知函數要產生的亂數數目，在此例是 10 個。
- ✓ 由於 `MTRsgObj#0000` 為一維亂數物件，因此需要保留  $10 \times 1$  個儲存格來放置輸出陣列。
- ✓ 如果 `MTRsgObj#0000` 為二維亂數物件，便需要保留  $10 \times 2$  個儲存格來放置輸出陣列。

► 由於此函數的輸出為陣列資料，我們需要先 Mark 住輸出的區域。

- ✓ 在下圖中，先把 H4 到 H13 的區域反白選定，接著輸入=qlVariates(G4, G5, G6)，
- ✓ 在同時按住 Ctrl+Shift 的情況下，按下 Enter 鍵即可產生結果，如圖 3.3 所示。

	C	D	E	F	G	H	I	J
1								
2								
3	MTRsgObj#0001			Variates	Output	Output		
4	MTRsgObj		string	RsgObjectId	MTRsgObj#0001	=qlVariates		
5	1		long	Samples	10	0.1466		
6	4321		any	Trigger	MTRsgObj#0001	0.8151		
7	FALSE					0.7423		
8	0					0.7679		
9	TRUE					0.1824		
10						0.2864		
11	1					0.7615		
12	MTRsgObj#0001					0.1931		
13	D:\Work1.ws					0.7980		
14	TRUE							

## (二)資料異動的傳遞

- ◆ 在 QuantLibXL 中，有些函數需要使用很多的輸入參數，一但這些輸入參數改變，這些函數便需要重新計算。
  - Excel 自動重算功能雖然方便，但是可能會產生許多不必要的運算，造成電腦太大的負擔，
  - 尤其是 QuantLibXL 中部份的函數需要執行大量的運算。
- ◆ 在 QuantLibXL 中，有兩個我們可以善加利用的函數，可以幫忙我們控制與傳遞資料的變動，簡化我們的作業。

## ■ ohRange() :

- ✧ 傳回值型態 : string
- ✧ 傳入參數一 : string(產生之物件希望的名稱) , Range
- ✧ 傳入參數二 : vector<vector<double>>(範圍) , H2:I3
- ✧ 傳入參數四 : bool(物件是否為永久的) , FALSE
- ✧ 傳入參數五 : any(觸發控制) , A1
- ✧ 傳入參數六 : bool(物件是否為覆蓋的) , FALSE

➤ 函數 ohRange() 可以將特定範圍的儲存格，產生一個範圍物件。

- ✓ 在 C2 格中，我們將 H2:I3 的範圍，傳入 ohRange() 函數中，以之產生 Range#0000 物件。
- ✓ 我們也可以使用 Excel 的名稱功能，先將範圍命名。
- ✓ 例如在 D2 中，我們輸入=ohRange(D3, Array2, D5, D6, D7)，第二個參數 Array2，是我們針對 H5:I6 所設定的名稱。

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E	F	G	H	I
1	TRUE								
2		OhRange	Range#0000	Vector#0000	Array#0000		Array1	4	5
3	string	ObjectId	Range	Vector	Array			4	8
4	vector	Values							
5	bool	Permanent	FALSE	FALSE	FALSE		Array2	1	2
6	any	Trigger	TRUE	TRUE	TRUE			3	4
7	bool	Overwrite	FALSE	FALSE	FALSE				
8							Array3	5	6
9								6	8
10									
11		OhTrigger	3						
12	any	Dummy1	Range#0000						
13	any	Dummy2	Vector#0000						
14	any	Dummy3	Array#0000						

➤ 一但範圍內的資料有異動，ohRange()函數便會重新產生物件，反應這個異動。

✓ 我們將 H2 格資料由 4 改成 3，C2 格便產生新的物件 Range#0001，便是這項作用。

	A	B	C	D	E	F	G	H	I
1	TRUE								
2		OhRange	Range#0001	Vector#0000	Array#0000		Array1	3	5
3	string	ObjectId	Range	Vector	Array			4	8
4	vector	Values							
5	bool	Permanent	FALSE	FALSE	FALSE		Array2	1	2
6	any	Trigger	TRUE	TRUE	TRUE			3	4
7	bool	Overwrite	FALSE	FALSE	FALSE				
8							Array3	5	6
9								6	8
10									
11		OhTrigger	4						
12	any	Dummy1	Range#0001						
13	any	Dummy2	Vector#0000						
14	any	Dummy3	Array#0000						
15									

■ ohTrigger() :

- ✧ 傳回值型態 : long
- ✧ 傳入參數一 : any
- ✧ 傳入參數二 : any
- ✧ 傳入參數三 : any
- ✧ ...
- ✧ 傳入參數十三 : any



➤ 函數 ohTrigger() 可以讓我們更進一步擴大傳遞異動的功能，

- ✓ ohTrigger() 可以接受最多 13 個輸入資料，並傳回一個長整數。
- ✓ 在 C11 格中，我們輸入=ohTrigger(C12, C13, C14)，傳回 0 這個數值。

	A	B	C	D	E	F	G	H	I
1	TRUE								
2		OhRange	Range#0000	Vector#0000	Array#0000		Array1	3	5
3	string	ObjectId	Range	Vector	Array			4	8
4	vector	Values							
5	bool	Permanent	FALSE	FALSE	FALSE		Array2	1	2
6	any	Trigger	TRUE	TRUE	TRUE			3	4
7	bool	Overwrite	FALSE	FALSE	FALSE				
8							Array3	5	6
9								6	8
10									
11		OhTrigger	0						
12	any	Dummy1	Range#0000						
13	any	Dummy2	Vector#0000						
14	any	Dummy3	Array#0000						

➤ 在圖中，我們異動了 H2 的內容，由 3 改成 2。

- ✓ 首先，C2 格的 ohRange()傳回值改變成 Range#0001。
- ✓ 然而，C11 格的 ohTrigger()傳回值改變成 1。我們可以使用這些功能，來控制我們想要的資料異動。

	A	B	C	D	E	F	G	H	I
1	TRUE								
2		OhRange	Range#0001	Vector#0000	Array#0000		Array1	2	5
3	string	ObjectId	Range	Vector	Array			4	8
4	vector	Values							
5	bool	Permanent	FALSE	FALSE	FALSE		Array2	1	2
6	any	Trigger	TRUE	TRUE	TRUE			3	4
7	bool	Overwrite	FALSE	FALSE	FALSE				
8							Array3	5	6
9								6	8
10									
11		OhTrigger	1						
12	any	Dummy1	Range#0001						
13	any	Dummy2	Vector#0000						
14	any	Dummy3	Array#0000						
15									

### (三)Permanent Flag

◆ 用來指示此物件是否為永久與否。

➤ 預設值為 **False**，表示此物件可以被刪除。

◆ 如果不是永久物件，可以使用 **ohRepositoryDeleteAllObjects()** /

**ohRepositoryCollectGarbage()** 這兩個函數來刪除非永久物件。

➤ 參數 **DeletePermanent** 預設為 **False**，只刪除非 **Permanent** 物件；設為 **True** 時，所有物件皆刪除。

■ **ohRepositoryDeleteAllObjects()**：刪除所有的物件。

✧ 傳回值型態：**void**

✧ 傳入參數：**(bool DeletePermant, any Trigger)**

■ **ohRepositoryCollectGarbage()**：刪除沒有使用的孤兒物件。

✧ 傳回值型態：**void**

✧ 傳入參數：**(bool DeletePermant, any Trigger)**

## (四)Overwrite Flag

- ◆ 此旗標預設值為 **False**，視函數有不同意義，通常使用表，覆蓋在記憶體中有相同名稱的物件。使用說明可參見 **ohObjectLoad()**函數文件說明如下，

### ■ ohObjectLoad()

- ✧ 傳回值型態：`vector<string>`

- ✧ 傳入參數：`(string Directory, string Pattern, bool Recuse, bool Overwrite, any Trigger)`

### ■ Description:

- ✧ Deserialize list of objects from given file, return IDs of deserialized objects.

### ■ Parameters:

Directory Directory from which objects are to be deserialized.

Pattern Name of XML file from which objects are to be deserialized, or a pattern in UNIX format (wildcard is `*`). Default value = `.*\xml`.

Recurse Recurse subdirectories of Directory when searching for filenames matching Pattern. Default value = false.

Overwrite Overwrite any existing Object that has the same ID as one being loaded. Default value = false.

Trigger dependency tracking trigger

◆ 另一使用方式，可參見 `ohObjectSave()` 函數文件說明如下，此旗標預設值為 `False`。

■ `ohObjectSave()`

✧ 傳回值型態：`long`

✧ 傳入參數：`(vector<string> ObjectList, string Filename, bool Overwrite, bool IncludeGroups, any Trigger)`

■ Description:

✧ Serialize list of objects to given path, return count of objects serialized.

■ Parameters:

<i>ObjectList</i>	list of IDs of objects to be serialized.
<i>Filename</i>	file name to which objects are to be serialized.
<i>Overwrite</i>	overwrite the output file if it exists. Default value = false.
<i>IncludeGroups</i>	include Groups in the serialisation. Default value = true.
<i>Trigger</i>	dependency tracking trigger

Microsoft Excel - Lec01.xls

檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H) 輸入需要解答的問題

120%

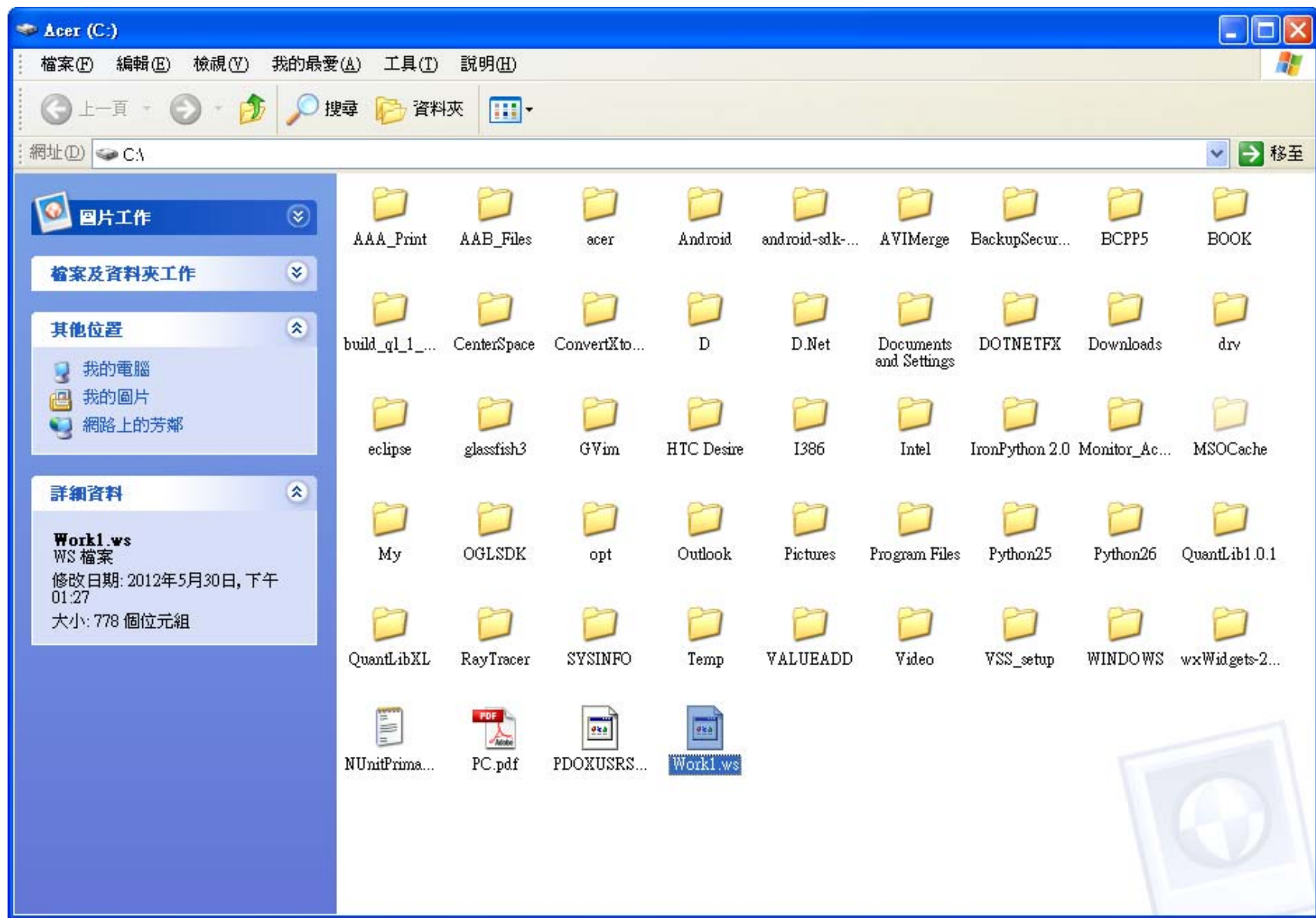
Arial 12 B I U

CORREL ✕ ✓ fx =ohObjectSave(C12,C13,C14,C15,C16)

	A	B	C	D	E	F
2						
3		<b>MersenneTwisterRsg</b>	<b>MTRsgObj#0000</b>			<b>Variates</b>
4	string	ObjectId	MTRsgObj		string	RsgObjectId
5	long	Dimmension	1		long	Samples
6	long	Seed	4321		any	Trigger
7	bool	Permenanent	FALSE			
8	any	Trigger	0			
9	bool	Overwrite	FALSE			
10						
11		<b>ObjectSave</b>	<b>=ohObjectSave(C</b>			
12	vec<string>	ObjectId	MTRsgObj#0000			
13	string	FileName	C:\Work1.ws			
14	bool	Overwrite	TRUE			
15	bool	IncludeGroup	TRUE			
16	any	Trigger				

Sheet1 Sheet2 Sheet3 Sheet4

編輯 NUM



## ■ Work1.ws 的檔案內容

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!DOCTYPE boost_serialization>
<boost_serialization signature="serialization::archive" version="7">
<object_list class_id="1" tracking_level="1" version="0" object_id="_0">
<count>1</count>
<item_version>1</item_version>
<item class_id="0" tracking_level="0" version="1">
  <px class_id="170" tracking_level="1" version="0" object_id="_1">
    <ObjectId>MTRsgObj</ObjectId>
    <ClassName>qlMersenneTwisterRsg</ClassName>
    <Dimension>1</Dimension>
    <Seed>4321</Seed>
    <Permanent>0</Permanent>
    <UserProperties class_id="236" tracking_level="0" version="0">
      <count>0</count>
      <item_version>0</item_version>
    </UserProperties>
  </px>
</item>
```



```
</object_list>  
</boost_serialization>
```

## ■ ohObjectLoad()

✧ 傳回值型態 : vector<string>

✧ 傳入參數 : (string Directory, string Pattern, bool Recurse, bool Overwrite, any Trigger)

## ■ Description:

✧ Deserialize list of objects from given file, return IDs of deserialized objects.

## ■ Parameters:

*Directory* Directory from which objects are to be deserialized.

*Pattern* Name of XML file from which objects are to be deserialized, or a pattern in UNIX format (wildcard is .\*). Default value = \*.\*\xml.

*Recurse* Recurse subdirectories of Directory when searching for filenames matching Pattern. Default value = false.

*Overwrite* Overwrite any existing Object that has the same ID as one being loaded. Default value = false.

*Trigger* dependency tracking trigger

Microsoft Excel - Lec01\_1.xls

檔案(F) 編輯(E) 檢視(V) 插入(I) 格式(O) 工具(T) 資料(D) 視窗(W) 說明(H) 輸入需要解答的問題

Arial 12 B I U \$ % , .00 100%

D2 =ohObjectLoad(D3,D4,D5,D6,D7)

	A	B	C	D	E	F	G	H	I
1									
2			ObjectLoad	MTRsgObj			Variates	Output	Output
3		string	Directory	C:\		string	RsgObjectId	MTRsgObj	0.0708
4		Pattern	FileName	Work1.ws		long	Samples	10	0.1466
5		bool	Recurse	FALSE		any	Trigger	MTRsgObj	0.8151
6		bool	Overwrite	TRUE					0.7423
7		any	Trigger						0.7679
8									0.1824
9									0.2864
10									0.7615
11									0.1931
12									0.7980
13									
14									
15									
16									
17									

Sheet1 / Sheet2 / Sheet3 /

就緒 NUM

## ■ ohObjectSaveString()

✧ 傳回值型態 : string

✧ 傳入參數 : (vector<string> ObjectList, bool Overwrite, any Trigger)

## ■ Description:

✧ Serialize list of objects to given path, return resulting XML.

## ■ Parameters:

*ObjectList* list of IDs of objects to be serialized.

*Overwrite* overwrite the output file if it exists. Default value = false.

*Trigger* dependency tracking trigger

## ■ ohObjectLoadString()

✧ 傳回值型態 : `vector<string>`

✧ 傳入參數 : (`string Xml`, `bool Overwrite`, `any Trigger`)

## ■ Description:

✧ Deserialize list of objects from given string, return IDs of deserialized objects.

## ■ Parameters:

*Xml* XML string from which objects are to be deserialized.

*Overwrite* Overwrite any existing Object that has the same ID as one being loaded. Default value = false.

*Trigger* dependency tracking trigger

## (五)錯誤資訊的取得

◆ 在使用 QuantLibXL 函數時，有時因輸入資料的問題，可能會產生錯誤。

➤ 此時可以使用一個輔助函數，ohRangeRetriveError()，來取得系統提供的錯誤訊息。

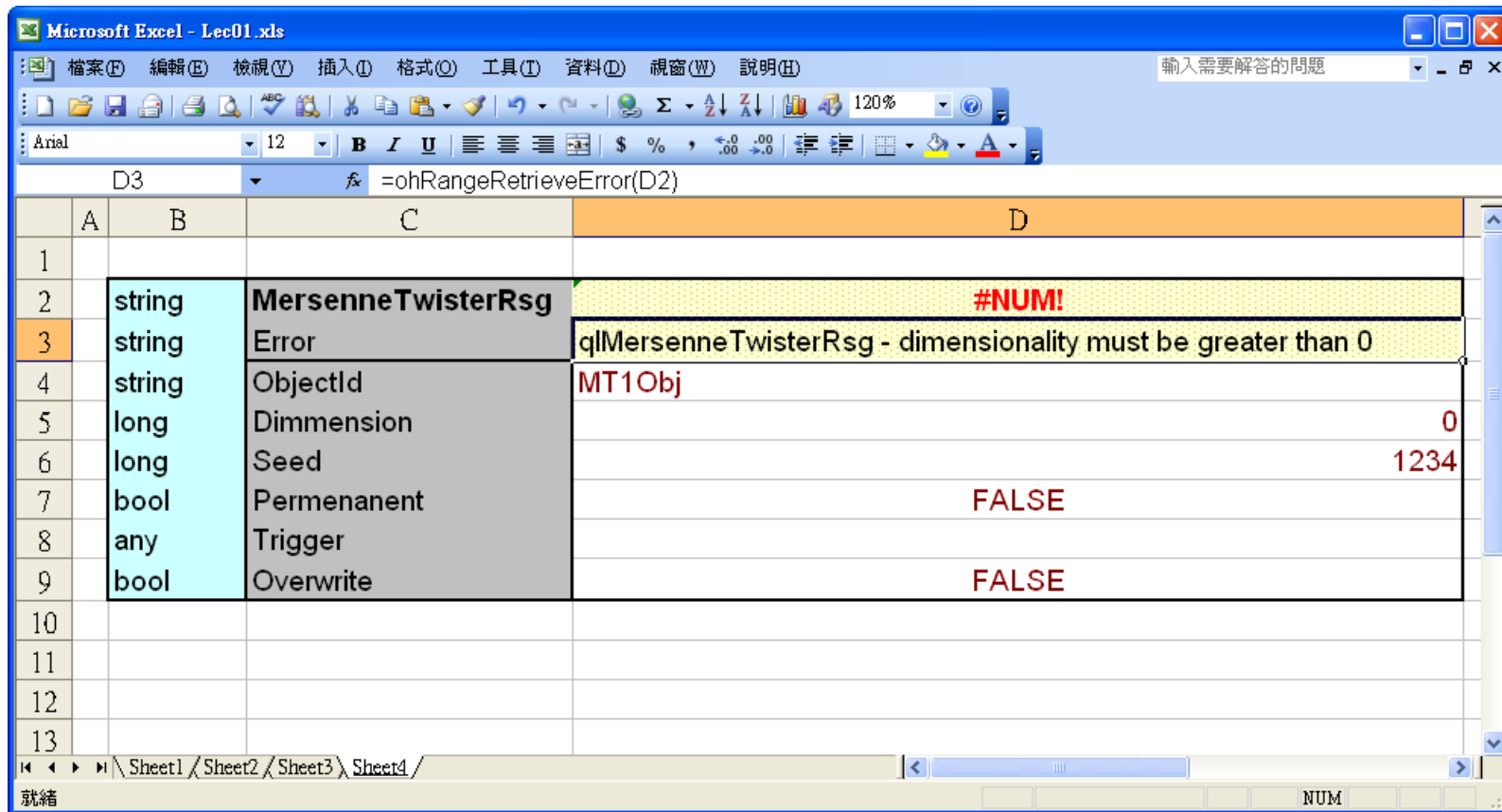
■ ohRangeRetrieveError()：傳回與給定 Range 有關的錯誤訊息

✧ 傳回值型態：string，

✧ 傳入參數：reference(Range，欲了解錯誤訊息的範圍)，

➤ 在圖中，我們在 C3 格輸入=ohRangeRetrieveError(C2)，C2 格是我們建立一個 Mersenne Twister 亂數產生器物件的位置。

- ✓ 由於我們在 C5 格輸入 0(維度不應為零)，以至於 C2 無法產生物件。
- ✓ C3 格的函數捕捉到 C2 的錯誤，
- ✓ 因此顯示錯誤訊息 "qlMersenneTwisterRsg - dimensionality must be greater than 0"。



## (六)LogFile

◆ 我們可以使用 `ohLogSetFile()` 函數來開啟日誌檔的寫入，以 `ohLogWriteMessage()` 寫入訊息。

➤ 此處 `LogLevel` 預設為 4，無須更改。

### ■ `ohLogSetFile()`

✧ 傳回值型態：`string`

✧ 傳入參數：`(string LogFileName, long LogLevel, any Trigger)`

### ■ `ohLogWriteMessage()`

✧ 傳回值型態：`void`

✧ 傳入參數：`(string LogMessage, long LogLevel, any Trigger)`

## (七)QuantLibXL 輸入資料的格式

◆ 在使用 QuantLibXL 函數時，其輸入參數與輸出資料各有格式的說明，下面針對我們會遇到的各種資料格式統一說明之。

➤ long：格式 long 表示資料為整數型態，除了代表整數外，也代表日期格式的資料。

✓ 在 Excel 中是以 1900/1/1 開始起算日期，當作第一天，因此其以整數 1 代表 1900/1/1，往後每增加一天，日期數就加一。

➤ bool：格式 bool 表示資料為 TRUE 或 FALSE 的布爾代數型態。

➤ string：格式 string 表示資料為字串型態。

➤ double：格式 double 表示資料為實數型態。

➤ vector<double>：格式 vector<double>表示資料為一維的實數向量型態，在 Worksheet 上代表一行或一系列的實數資料，

✓ 例如 A1:A10、A10:J10 都是。在本書中，我們會以 vec<double>簡寫表達之。



- `vector<vector<double>>`：格式 `vector<vector<double>>` 表示資料為二維的實數向量型態，在 Worksheet 上代表多行或多列的實數資料，
  - ✓ 例如 A1:J10 即是。在本書中，我們會以 `vec<vec<double>>` 簡寫表達之。
- `any`：格式 `any` 表示資料可為任何的型態，整數、字串、布爾代數、...，皆有可能。
- ◆ 至於其他可能的型態，如 `vector<long>`，讀者應當可以合理推知代表整數向量資料或日期向量資料。





