

Basel III 市場風險標準法試算實作班

使用 Python 與 QuantLib 套件

昀騰金融科技

技術長

董夢雲 博士

dongmy@ms5.hinet.net

主題八、波動性曲線結構類別

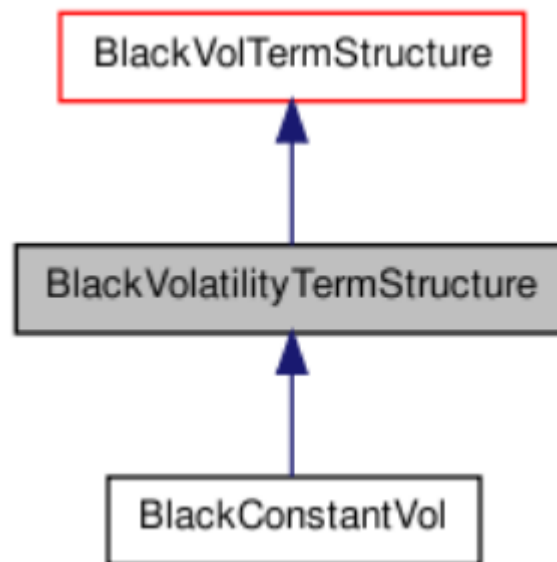
- 一、水平波動性 BlackConstantVol 物件
- 二、波動性曲線 BlackVarianceCurve 物件
- 三、波動性曲面 BlackVarianceSurface 物件
- 四、BlackVolTermStructureHandle 物件

程式範例：Ch08_01.ipynb ~ Ch08_04.ipynb

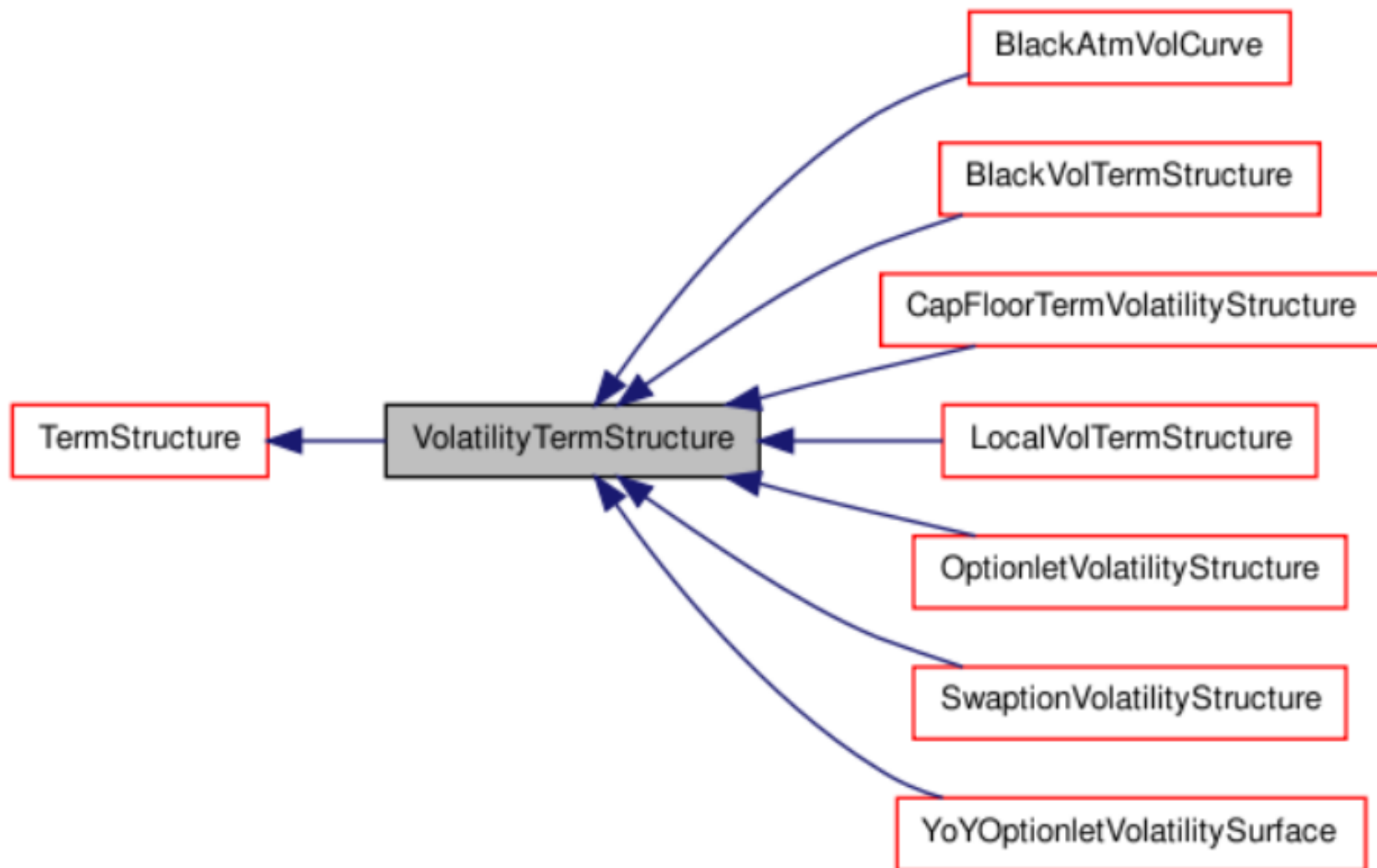
◆ 波動性期限結構物件旨在描述整條的波動性曲線。

- 事實上，波動性不只隨到期時間的不同而有不同，他也因選擇權的執行價格不同而有差異。
 - ✓ 真實的行況是，他是一個三維的曲面。
- 本單元中，我們先由最簡單的情況，固定波動性(Constant Volatility Model)來說明，
 - ✓ 複雜的情況我們在後面的章節在詳加介紹。

◆ BlackVolTermStructure 物件繼承圖



◆ VolatilityTermStructure 物件繼承圖



一、BlackConstantVol 物件建構

◆ 建構一條水平固定的波動性曲線，波動性與時間、執行價格無關，為定數。

- BlackConstantVol ()

- ✧ 語法

```
def __init__(self, *args):  
    _QuantLib.BlackConstantVol_swiginit(self, _QuantLib.new_BlackConstantVol(*args))
```

- ✧ 使用

```
>>> import QuantLib as ql  
>>> settings = ql.Settings.instance()  
>>> evDate = ql.Date(8, 6, 2021)  
>>> settings.setEvaluationDate(evDate)  
>>> Cal = ql.NullCalendar()  
>>> DC365 = ql.Actual365Fixed()  
>>> settlementDays = 2  
>>> vol = 0.30  
>>> ConstVolTS = ql.BlackConstantVol(evDate, Cal, vol, DC365)
```

- BlackCalibrationHelperVector
- BlackCalibrationHelperVector.BlackCalibrationHelp
- BlackCallableFixedRateBondEngine
- BlackCapFloorEngine
- BlackCdsOptionEngine
- BlackConstantVol
- BlackDeltaCalculator
- BlackIborCouponPricer
- BlackKarasinski
- BlackProcess
- BlackScholesMertonProcess
- BlackScholesProcess
- BlackSwaptionEngine
- BlackVarianceCurve
- BlackVarianceSurface
- BlackVarianceSurface.Extrapolation
- BlackVolTermStructure
- BlackVolTermStructureHandle
- Bond
- BondFunctions
- BondHelper
- BondHelperVector
- BondHelperVector.BondHelperVectorEnumerator
- BoolVector
- BoolVector.BoolVectorEnumerator
- BoundaryConstraint
- BoxMullerKnuthGaussianRng
- BoxMullerLecuyerGaussianRng
- BoxMullerMersenneTwisterGaussianRng
- Brazil
- Brazil.Market
- Brent
- BRLCurrency
- BrownianBridge
- BrownianGeneratorFactory
- Business252
- BusinessDayConvention
- BYRCurrency
- CADCurrency
- CADLiber
- Calendar

- BlackConstantVol(QuantLib.Date, QuantLib.Calendar, double, QuantLib.DayCounter)
- BlackConstantVol(QuantLib.Date, QuantLib.Calendar, QuantLib.QuoteHandle, QuantLib.DayCounter)
- BlackConstantVol(System.IntPtr, bool)
- BlackConstantVol(uint, QuantLib.Calendar, double, QuantLib.DayCounter)
- BlackConstantVol(uint, QuantLib.Calendar, QuantLib.QuoteHandle, QuantLib.DayCounter)
- Dispose(bool)
- getCPtr(QuantLib.BlackConstantVol)
- swigCMemOwnDerived
- swigCPtr

```

public BlackConstantVol(QuantLib.Date referenceDate, QuantLib.Calendar c, double volatility, QuantLib.DayCounter
dayCounter)
QuantLib.BlackConstantVol 的成員

```

- BlackVarianceSurface.Extrapolation
- BlackVolTermStructure
- BlackVolTermStructureHandle
- Bond
- BondFunctions
- BondHelper
- BondHelperVector
- BondHelperVector.BondHelperVectorEnumerator
- BoolVector
- BoolVector.BoolVectorEnumerator
- BoundaryConstraint
- BoxMullerKnuthGaussianRng
- BoxMullerLecuyerGaussianRng
- BoxMullerMersenneTwisterGaussianRng
- Brazil
- Brazil.Market
- Brent
- BRLCurrency
- BrownianBridge
- BrownianGeneratorFactory
- Business252
- BusinessDayConvention
- BYRCurrency
- CADCurrency
- CADLibor
- Calendar
- CalibratedModel
- CalibratedModelHandle
- CalibrationErrorTuple
- CalibrationHelperBase
- CalibrationHelperVector
- CalibrationHelperVector.CalibrationHelperVectorEnumerator
- CalibrationPair
- CalibrationSet
- CalibrationSet.CalibrationSetEnumerator
- Callability
- Callability.Type
- CallabilityPrice
- CallabilityPrice.Type
- CallabilitySchedule
- CallabilitySchedule.CallabilityScheduleEnumerator

- blackForwardVariance(double, double, double)
- blackForwardVariance(double, double, double, bool)
- blackForwardVariance(QuantLib.Date, QuantLib.Date, double)
- blackForwardVariance(QuantLib.Date, QuantLib.Date, double, bool)
- blackForwardVol(double, double, double)
- blackForwardVol(double, double, double, bool)
- blackForwardVol(QuantLib.Date, QuantLib.Date, double)
- blackForwardVol(QuantLib.Date, QuantLib.Date, double, bool)
- blackVariance(double, double)
- blackVariance(double, double, bool)
- blackVariance(QuantLib.Date, double)
- blackVariance(QuantLib.Date, double, bool)
- blackVol(double, double)**
- blackVol(double, double, bool)
- blackVol(QuantLib.Date, double)
- blackVol(QuantLib.Date, double, bool)
- BlackVolTermStructure(System.IntPtr, bool)
- Dispose(bool)
- getCPtr(QuantLib.BlackVolTermStructure)
- swigCMemOwnDerived
- swigCPtr

```
public double blackVol(double arg0, double strike)
    QuantLib.BlackVolTermStructure 的成員
```

- VanillaForwardPayoff
- VanillaOption
- VanillaSwap
- VanillaSwap.Type
- VanillaSwingOption
- VannaVolgaBarrierEngine
- VannaVolgaKDoubleBarrierEngine
- VannaVolgaWODoubleBarrierEngine
- VarianceGammaEngine
- VarianceGammaProcess
- Vasicek
- VEBCurrency
- VNDCurrency
- VolatilityTermStructure
- VolatilityType
- Weekday
- WeekendsOnly
- Wibor
- WulinYongDoubleBarrierEngine
- YearOnYearInflationSwap
- YearOnYearInflationSwap.Type
- YearOnYearInflationSwapHelper
- YieldTermStructure
- YieldTermStructureHandle
- YoYHelper
- YoYHelperVector
- YoYHelperVector.YoYHelperVectorEnumerator
- YoYInflationCap
- YoYInflationCapFloor
- YoYInflationCollar
- YoYInflationFloor
- YoYInflationIndex
- YoYInflationTermStructure
- YoYInflationTermStructureHandle
- YYEUHICP
- YYEUHICPXT
- YYFRHICP
- YYUKRPI
- YYUSCPI
- YYZACPI
- ZabrFullFd

- Dispose(bool)
- getCPtr(QuantLib.VolatilityTermStructure)
- maxStrike()**
- minStrike()
- VolatilityTermStructure(System.IntPtr, bool)
- swigCMemOwnDerived
- swigCPtr

```
public double maxStrike()
```

[QuantLib.VolatilityTermStructure](#) 的成員

- Switzerland
- Taiwan
- Taiwan.Market
- TARGET
- TemperatureExponential
- TermStructure
- TermStructureConsistentModel
- Thailand
- THBCurrency
- THBFIX
- Thirty360
- Thirty360.Convention
- Tibor
- TimeBasket
- TimeGrid
- TimeUnit
- TrapezoidIntegralDefault
- TrapezoidIntegralMidPoint
- TreeCallableFixedRateBondEngine
- TreeCapFloorEngine
- TreeSwaptionEngine
- TridiagonalOperator
- TRLCurrency
- TRLibor
- TRYCurrency
- TTDCurrency
- Turkey
- TWDCurrency
- TypePayoff
- Ukraine
- Ukraine.Market
- UKRPI
- UnaryFunction
- UnaryFunctionDelegate
- UnaryFunctionDelegate.SwigDelegateUnaryFunction
- UniformLowDiscrepancySequenceGenerator
- UniformRandomGenerator
- UniformRandomSequenceGenerator
- UnitedKingdom
- UnitedKingdom.Market
- UnitedStates

- allowsExtrapolation()
- calendar()
- dayCounter()
- disableExtrapolation()
- Dispose(bool)
- enableExtrapolation()
- getCPtr(QuantLib.TermStructure)
- maxDate()
- maxTime()
- referenceDate()
- TermStructure(System.IntPtr, bool)
- timeFromReference(QuantLib.Date)
- swigCMemOwnDerived
- swigCPtr

```

public QuantLib.Date referenceDate()
QuantLib.TermStructure 的成員

```

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

Home Page - Select or create x Ch08_01 - Jupyter Notebook x +

localhost:8888/notebooks/Ch08_01.ipynb 150% ☆

jupyter Ch08_01 (autosaved) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
In [1]: import QuantLib as ql
        settings = ql.Settings.instance()
        evDate = ql.Date(8, 6, 2021)
        settings.setEvaluationDate(evDate)
        Cal = ql.NullCalendar()
        DC365 = ql.Actual365Fixed()
        settlementDays = 2

In [2]: vol = 0.30

In [3]: ConstVolTS = ql.BlackConstantVol(evDate, Cal, vol, DC365)

In [4]: volly = ConstVolTS.blackVol(1.0, 100)

In [5]: print(volly)

0.3
```

- Function : `blackVol()`

- ✧ 根據輸入的選擇權的到期日與執行價格，傳回相對應的 Black Volatility。

- ✧ 語法

```
def blackVol(self, *args):  
    return _QuantLib.BlackVolTermStructure_blackVol(self, *args)
```

- ✧ 使用

```
>>> vol1Y = ConstVolTS.blackVol(1.0, 100.0)
```

- Function : `blackVariance()`

- ✧ 根據輸入的選擇權的到期日與執行價格，傳回相對應的 Black Variance。

- ✧ 語法

```
def blackVariance(self, *args):  
    return _QuantLib.BlackVolTermStructure_blackVariance(self, *args)
```

- ✧ 使用

```
>>> var1Y = ConstVolTS.blackVariance(1.0, 100.0)
```

- Function : `blackForwardVol()`

- ✧ 根據輸入的未來評估波動性的日期與選擇權的到期日，傳回相對應的未來 At-The-Money(ATM)的 Black Volatility。

- ✧ 語法

```
def blackForwardVol(self, *args):  
    return _QuantLib.BlackVolTermStructure_blackForwardVol(self, *args)
```

- ✧ 使用

- Function : `blackForwardVariance()`

- ✧ 根據輸入的未來評估波動性的日期與選擇權的到期日，傳回相對應的未來 At-The-Money(ATM)的 Black Variance。

- ✧ 語法

```
def blackForwardVariance(self, *args):  
    return _QuantLib.BlackVolTermStructure_blackForwardVariance(self, *args)
```

- ✧ 使用

- ▷ BlackCalibrationHelperVector
- ▷ BlackCalibrationHelperVector.BlackCalibrationHelperVectorE
- ▷ BlackCallableFixedRateBondEngine
- ▷ BlackCapFloorEngine
- ▷ BlackCdsOptionEngine
- ▷ BlackConstantVol
- ▷ BlackDeltaCalculator
- ▷ BlackIborCouponPricer
- ▷ BlackKarasinski
- ▷ BlackProcess
- ▷ BlackScholesMertonProcess
- ▷ BlackScholesProcess
- ▷ BlackSwaptionEngine
- ▷ BlackVarianceCurve
- ▷ BlackVarianceSurface
- ▷ BlackVarianceSurface.Extrapolation
- ▷ BlackVolTermStructure
- ▷ BlackVolTermStructureHandle
- ▷ Bond
- ▷ BondFunctions
- ▷ BondHelper
- ▷ BondHelperVector
- ▷ BondHelperVector.BondHelperVectorEnumerator
- ▷ BoolVector
- ▷ BoolVector.BoolVectorEnumerator
- ▷ BoundaryConstraint
- ▷ BoxMullerKnuthGaussianRng
- ▷ BoxMullerLecuyerGaussianRng
- ▷ BoxMullerMersenneTwisterGaussianRng
- ▷ Brazil
- ▷ Brazil.Market
- ▷ Brent
- ▷ BRLCurrency
- ▷ BrownianBridge
- ▷ BrownianGeneratorFactory
- ▷ Business252
- ▷ BusinessDayConvention
- ▷ BYRCurrency
- ▷ CADCurrency
- ▷ CADLibor
- ▷ Calendar

- ⊗ blackForwardVariance(double, double, double)
- ⊗ blackForwardVariance(double, double, double, bool)
- ⊗ blackForwardVariance(QuantLib.Date, QuantLib.Date, double)
- ⊗ blackForwardVariance(QuantLib.Date, QuantLib.Date, double, bool)
- ⊗ blackForwardVol(double, double, double)
- ⊗ blackForwardVol(double, double, double, bool)
- ⊗ blackForwardVol(QuantLib.Date, QuantLib.Date, double)
- ⊗ **blackForwardVol(QuantLib.Date, QuantLib.Date, double, bool)**
- ⊗ blackVariance(double, double)
- ⊗ blackVariance(double, double, bool)
- ⊗ blackVariance(QuantLib.Date, double)
- ⊗ blackVariance(QuantLib.Date, double, bool)
- ⊗ blackVol(double, double)
- ⊗ blackVol(double, double, bool)
- ⊗ blackVol(QuantLib.Date, double)
- ⊗ blackVol(QuantLib.Date, double, bool)
- ⊗ BlackVolTermStructure(System.IntPtr, bool)
- ⊗ Dispose(bool)
- ⊗ getCPtr(QuantLib.BlackVolTermStructure)
- ⊗ swigCMemOwnDerived
- ⊗ swigCPtr

public [double](#) blackForwardVol([QuantLib.Date](#) arg0, [QuantLib.Date](#) arg1, [double](#) strike, [bool](#) extrapolate)
[QuantLib.BlackVolTermStructure](#) 的成員

- Function : maxStrike()

- ✧ 傳回此物件的可計算的最大執行價格。在此例中，由於我們沒有設定(Constant Vol 不需設定)，因此預設以最大實數來表達。

- ✧ 語法

```
def minStrike(self):  
    return _QuantLib.VolatilityTermStructure_minStrike(self)
```

- ✧ 使用

- Function : minStrike()

- ✧ 傳回此物件的可計算的最小執行價格。在此例中，由於我們沒有設定(Constant Vol 不需設定)，因此以預設最小實數來表達。

- ✧ 語法

```
def maxStrike(self):  
    return _QuantLib.VolatilityTermStructure_maxStrike(self)
```

- ✧ 使用

二、BlackVarianceCurve 物件建構

◆ 建構一條波動性期限結構曲線，波動性為到期時間的函數，為一線性關係。

- BlackVarianceCurve ()

- ✧ 語法

```
def __init__(self, referenceDate, dates, volatilities, dayCounter, forceMonotoneVariance=True):  
    _QuantLib.BlackVarianceCurve_swiginit(self, _QuantLib.new_BlackVarianceCurve  
        (referenceDate, dates, volatilities, dayCounter, forceMonotoneVariance))
```

- ✧ 使用

```
>>> d1 = ql.Date(10, 9, 2021)  
>>> d2 = ql.Date(10, 12, 2021)  
>>> d3 = ql.Date(10, 3, 2022)  
>>> d4 = ql.Date(10, 6, 2022)  
>>> V_Date = ql.DateVector([d1, d2, d3, d4])  
>>> V_Vol = ql.DoubleVector([0.2, 0.3, 0.35, 0.4])  
>>> blackVarTS = ql.BlackVarianceCurve(refDate, V_Date, V_Vol, DC365)  
>>> blackVarTS.blackVol(0.65, 100)
```

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

Home Page - Select or create x Ch08_02 - Jupyter Notebook x +

localhost:8888/notebooks/Ch08_02.ipynb 150% ☆

jupyter Ch08_02 (unsaved changes) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
In [1]: import QuantLib as ql
settings = ql.Settings.instance()
evDate = ql.Date(8, 6, 2021)
settings.setEvaluationDate(evDate)
Cal = ql.NullCalendar()
DC365 = ql.Actual365Fixed()
settlementDays = 2
refDate = Cal.advance(evDate, 2, ql.Days, ql.Following, False)

In [2]: d1 = ql.Date(10, 9, 2021)
d2 = ql.Date(10, 12, 2021)
d3 = ql.Date(10, 3, 2022)
d4 = ql.Date(10, 6, 2022)

In [3]: V_Date = ql.DateVector([d1, d2, d3, d4])

In [4]: V_Vol = ql.DoubleVector([0.2, 0.3, 0.35, 0.4])

In [5]: blackVarTS = ql.BlackVarianceCurve(refDate, V_Date, V_Vol, DC365)

In [6]: blackVarTS.blackVol(0.65, 100)

Out[6]: 0.33547315457860766
```


- BlackKarasinski
- BlackProcess
- BlackScholesMertonProcess
- BlackScholesProcess
- BlackSwaptionEngine
- BlackVarianceCurve
- BlackVarianceSurface
- BlackVarianceSurface.Extrapolation
- BlackVolTermStructure
- BlackVolTermStructureHandle
- Bond
- BondFunctions
- BondHelper
- BondHelperVector
- BondHelperVector.BondHelperVectorEnumerator
- BoolVector
- BoolVector.BoolVectorEnumerator
- BoundaryConstraint
- BoxMullerKnuthGaussianRng
- BoxMullerLecuyerGaussianRng
- BoxMullerMersenneTwisterGaussianRng
- Brazil
- Brazil.Market
- Brent
- BRLCurrency
- BrownianBridge
- BrownianGeneratorFactory
- Business252
- BusinessDayConvention
- BYRCurrency
- CADCurrency
- CADLibor
- Calendar
- CalibratedModel
- CalibratedModelHandle
- CalibrationErrorTuple
- CalibrationHelperBase
- CalibrationHelperVector
- CalibrationHelperVector.CalibrationHelperVectorEn
- CalibrationPair
- CalibrationSet

- BlackVarianceCurve(QuantLib.Date, QuantLib.DateVector, QuantLib.DoubleVector, QuantLib.DayCounter)
- BlackVarianceCurve(QuantLib.Date, QuantLib.DateVector, QuantLib.DoubleVector, QuantLib.DayCounter, bool)
- BlackVarianceCurve(System.IntPtr, bool)
- Dispose(bool)
- getCPtr(QuantLib.BlackVarianceCurve)
- swigCMemOwnDerived
- swigCPtr

```

public BlackVarianceCurve(QuantLib.Date referenceDate, QuantLib.DateVector dates, QuantLib.DoubleVector volatilities,
QuantLib.DayCounter dayCounter)
QuantLib.BlackVarianceCurve 的成員

```

三、BlackVarianceSurface 物件建構

◆ 建構一條水平固定的波動性曲線，波動性為時間與執行價格的函數，為一平面關係。

- BlackVarianceSurface ()

- ✧ 語法

```
def __init__(self, *args, **kwargs):  
    _QuantLib.BlackVarianceSurface_swiginit(self, _QuantLib.new_BlackVarianceSurface(*args,  
        **kwargs))
```

- ✧ 使用

```
>>> d1 = ql.Date(10, 9, 2021)  
>>> d2 = ql.Date(10, 12, 2021)  
>>> d3 = ql.Date(10, 3, 2022)  
>>> d4 = ql.Date(10, 6, 2022)  
>>> V_Date = ql.DateVector([d1, d2, d3, d4])  
>>> V_Strike = ql.DoubleVector([85.0, 100.0, 115.0])  
  
>>> M_Vol = ql.Matrix(3, 4, 0.0)  
>>> M_Vol[0][0] = 0.25
```

```
>>> M_Vol[0][1] = 0.35
```

```
>>> M_Vol[0][2] = 0.40
```

```
>>> M_Vol[0][3] = 0.45
```

```
>>> M_Vol[1][0] = 0.20
```

```
>>> M_Vol[1][1] = 0.30
```

```
>>> M_Vol[1][2] = 0.35
```

```
>>> M_Vol[1][3] = 0.40
```

```
>>> M_Vol[2][0] = 0.25
```

```
>>> M_Vol[2][1] = 0.35
```

```
>>> M_Vol[2][2] = 0.40
```

```
>>> M_Vol[2][3] = 0.45
```

```
>>> blackSurface = ql.BlackVarianceSurface(refDate, Cal, V_Date, V_Strike, M_Vol, DC365)
```

```
>>> blackSurface.blackVol(0.65, 95)
```

```
>>>blackSurface.blackVol(0.65, 105)
```

- BlackKarasinski
- BlackProcess
- BlackScholesMertonProcess
- BlackScholesProcess
- BlackSwaptionEngine
- BlackVarianceCurve
- BlackVarianceSurface
- BlackVarianceSurface.Extrapolation
- BlackVolTermStructure
- BlackVolTermStructureHandle
- Bond
- BondFunctions
- BondHelper
- BondHelperVector
- BondHelperVector.BondHelperVectorEnumerator
- BoolVector
- BoolVector.BoolVectorEnumerator
- BoundaryConstraint
- BoxMullerKnuthGaussianRng
- BoxMullerLecuyerGaussianRng
- BoxMullerMersenneTwisterGaussianRng
- Brazil
- Brazil.Market
- Brent
- BRLCurrency
- BrownianBridge
- BrownianGeneratorFactory
- Business252
- BusinessDayConvention
- BYRCurrency
- CADCurrency
- CADLibor
- Calendar
- CalibratedModel
- CalibratedModelHandle
- CalibrationErrorTuple
- CalibrationHelperBase
- CalibrationHelperVector
- CalibrationHelperVector.CalibrationHelperVectorEnumerator
- CalibrationPair
- CalibrationSet

- BlackVarianceSurface(QuantLib.Date, QuantLib.Calendar, QuantLib.DateVector, QuantLib.DoubleVector, QuantLib.Matrix, QuantLib.DayCounter)
- BlackVarianceSurface(QuantLib.Date, QuantLib.Calendar, QuantLib.DateVector, QuantLib.DoubleVector, QuantLib.Matrix, QuantLib.DayCounter, QuantLib.BlackVarianceSurface.Extrapolation)
- BlackVarianceSurface(QuantLib.Date, QuantLib.Calendar, QuantLib.DateVector, QuantLib.DoubleVector, QuantLib.Matrix, QuantLib.DayCounter, QuantLib.BlackVarianceSurface.Extrapolation, QuantLib.BlackVarianceSurface.Extrapolation)
- BlackVarianceSurface(QuantLib.Date, QuantLib.Calendar, QuantLib.DateVector, QuantLib.DoubleVector, QuantLib.Matrix, QuantLib.DayCounter, QuantLib.BlackVarianceSurface.Extrapolation, QuantLib.BlackVarianceSurface.Extrapolation, QuantLib.BlackVarianceSurface.Extrapolation)
- BlackVarianceSurface(System.IntPtr, bool)
- Dispose(bool)
- getCPtr(QuantLib.BlackVarianceSurface)
- setInterpolation()
- setInterpolation(string)
- swigCMemOwnDerived
- swigCPtr

```

public BlackVarianceSurface(QuantLib.Date referenceDate, QuantLib.Calendar cal, QuantLib.DateVector dates, QuantLib.DoubleVector
strikes, QuantLib.Matrix blackVols, QuantLib.DayCounter dayCounter)
QuantLib.BlackVarianceSurface 的成員

```

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

Home Page - Select or create × Ch08_03 - Jupyter Notebook × +

localhost:8888/notebooks/Ch08_03.ipynb 150% ☆

jupyter Ch08_03 (unsaved changes) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
In [1]: import QuantLib as ql
settings = ql.Settings.instance()
evDate = ql.Date(8, 6, 2021)
settings.setEvaluationDate(evDate)
Cal = ql.NullCalendar()
DC365 = ql.Actual365Fixed()
settlementDays = 2
refDate = Cal.advance(evDate, 2, ql.Days, ql.Following, False)

In [2]: d1 = ql.Date(10, 9, 2021)
d2 = ql.Date(10, 12, 2021)
d3 = ql.Date(10, 3, 2022)
d4 = ql.Date(10, 6, 2022)

In [3]: V_Date = ql.DateVector([d1, d2, d3, d4])

In [4]: V_Strike = ql.DoubleVector([85.0, 100.0, 115.0])

In [5]: M_Vol = ql.Matrix(3, 4, 0.0)
```

```
In [6]: M_Vol[0][0] = 0.25  
        M_Vol[0][1] = 0.35  
        M_Vol[0][2] = 0.40  
        M_Vol[0][3] = 0.45
```

```
In [7]: M_Vol[1][0] = 0.20  
        M_Vol[1][1] = 0.30  
        M_Vol[1][2] = 0.35  
        M_Vol[1][3] = 0.40
```

```
In [8]: M_Vol[2][0] = 0.25  
        M_Vol[2][1] = 0.35  
        M_Vol[2][2] = 0.40  
        M_Vol[2][3] = 0.45
```

```
In [9]: blackSurface = ql.BlackVarianceSurface(refDate, Cal, V_Date, V_Strike, M_Vol, DC365)
```

```
In [10]: blackSurface.blackVol(0.65, 95)
```

```
Out[10]: 0.3528903270887654
```

```
In [11]: blackSurface.blackVol(0.65, 105)
```

```
Out[11]: 0.3528903270887654
```

四、BlackVolTermStructureHandle 物件

◆ 在 QuantLib 中有一類物件稱之為 Handle，其功能類似一個指標物件

- 藉由封裝成一個物件，變成一個 Smart Pointer
 - ✓ 可以提領其包含的物件的所有屬性與方法
- 當包含的物件轉換時，可以將此情況反映給使用的物件
 - ✓ 實作 Observer/Observable 樣式，搭配 Lazy Object 的架構。

◆ 建構子

● BlackVolTermStructureHandle ()

✧ 語法

```
def __init__(self, *args):  
    _QuantLib.BlackVolTermStructureHandle_swiginit(self,  
        _QuantLib.new_BlackVolTermStructureHandle(*args))
```

✧ 使用

```
>>> h_ConstVolTS = ql.BlackVolTermStructureHandle(ConstVolTS)  
>>> h_ConstVolTS.blackVol(1.0, 100.0)  
>>> h_ConstVolTS.dayCounter()
```

- BlackProcess
- BlackScholesMertonProcess
- BlackScholesProcess
- BlackSwaptionEngine
- BlackVarianceCurve
- BlackVarianceSurface
- BlackVarianceSurface.Extrapolation
- BlackVolTermStructure
- BlackVolTermStructureHandle
- Bond
- BondFunctions
- BondHelper
- BondHelperVector
- BondHelperVector.BondHelperVectorEnumerator
- BoolVector
- BoolVector.BoolVectorEnumerator
- BoundaryConstraint
- BoxMullerKnuthGaussianRng
- BoxMullerLecuyerGaussianRng
- BoxMullerMersenneTwisterGaussianRng
- Brazil
- Brazil.Market
- Brent
- BRLCurrency
- BrownianBridge
- BrownianGeneratorFactory
- Business252
- BusinessDayConvention
- BYRCurrency
- CADCurrency
- CADLibor
- Calendar
- CalibratedModel
- CalibratedModelHandle
- CalibrationErrorTuple
- CalibrationHelperBase
- CalibrationHelperVector
- CalibrationHelperVector.CalibrationHelperVectorEnumerator
- CalibrationPair
- CalibrationSet
- CalibrationSet.CalibrationSetEnumerator

- ~BlackVolTermStructureHandle()
- allowsExtrapolation()
- asObservable()
- blackForwardVariance(double, double, double)
- blackForwardVariance(double, double, double, bool)
- blackForwardVariance(QuantLib.Date, QuantLib.Date, double)
- blackForwardVariance(QuantLib.Date, QuantLib.Date, double, bool)
- blackForwardVol(double, double, double)
- blackForwardVol(double, double, double, bool)
- blackForwardVol(QuantLib.Date, QuantLib.Date, double)
- blackForwardVol(QuantLib.Date, QuantLib.Date, double, bool)
- blackVariance(double, double)
- blackVariance(double, double, bool)
- blackVariance(QuantLib.Date, double)
- blackVariance(QuantLib.Date, double, bool)
- blackVol(double, double)
- blackVol(double, double, bool)
- blackVol(QuantLib.Date, double)
- blackVol(QuantLib.Date, double, bool)
- BlackVolTermStructureHandle()
- BlackVolTermStructureHandle(QuantLib.BlackVolTermStructure)
- BlackVolTermStructureHandle(System.IntPtr, bool)
- calendar()
- dayCounter()
- __deref__()
- disableExtrapolation()
- Dispose()
- Dispose(bool)
- empty()
- enableExtrapolation()
- getCPtr(QuantLib.BlackVolTermStructureHandle)
- maxDate()
- maxStrike()
- maxTime()
- minStrike()
- referenceDate()
- timeFromReference(QuantLib.Date)

```
public BlackVolTermStructureHandle(QuantLib.BlackVolTermStructure arg0)
    QuantLib.BlackVolTermStructureHandle 的成員
```


檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

Home Page - Select or create. x Ch08_04 - Jupyter Notebook x +

localhost:8888/notebooks/Ch08_04.ipynb 150% ☆

jupyter Ch08_04 (autosaved) Python 3

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
In [1]: import QuantLib as ql
settings = ql.Settings.instance()
evDate = ql.Date(8, 6, 2021)
settings.setEvaluationDate(evDate)
Cal = ql.NullCalendar()
DC365 = ql.Actual365Fixed()
settlementDays = 2

In [2]: vol = 0.30
ConstVolTS = ql.BlackConstantVol(evDate, Cal, vol, DC365)

In [3]: h_ConstVolTS = ql.BlackVolTermStructureHandle(ConstVolTS)

In [4]: print(h_ConstVolTS.blackVol(1.0, 100.0))
print(h_ConstVolTS.dayCounter())

0.3
Actual/365 (Fixed) day counter
```

