

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

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

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

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

Julia 程式語言學習馬拉松

Day 11



cupay 陪跑專家 : James Huang



日期與時間





重要知識點



- Julia 提供了不同的型別和大量的函式，讓開發和研究人員可以方便的操作日期與時間資料，今天的內容要介紹日期與時間型別及其操作。
- 除了 Julia 內建的日期與時間模組，今天的內容也將介紹用來處理時區的 `TimeZones.jl` 套件。



日期與時間的建立



- 日期與時間相關的屬性與函式定義在 Dates 模組內，開始使用前先用 using 或 import 引入。

```
In [1]: using Dates
```

- 最常用的型別有 Date、DateTime、Time。
- Date 與 DateTime 型別對於日期的計算遵循 ISO 8601 的日期和時間的表示方法。
- 對於公元前日期的表示方式比較特別，公元前一年表示為 0000 年，公元前二年為 -0001 年。



日期與時間的建立



- 使用 `DateTime`、`Date`、`Time` 建構子建立日期與時間，除了第一個值是必要輸入的之外，其他未輸入的值會以預設值自動帶入。
 - `DateTime(year, [month, day, hour, minute, second, millisecond])`
 - `Date(year, [month, day])`
 - `Time(hour, [minute, second])`
- 除了使用建構子外，也可以使用 `Period` 型別指定。請留意 `Period` 型別為可變參數其均以大寫開頭，例如：`DateTime(Month(10), Millisecond(100), Year(2019))`



從日期與時間取得各周期值



- 透過函式或 Period 型別，可以取得各周期值。

函式或 Period	說明
year(), Year()	取得年，回傳值分別為 Int64 和 Period 型別。
month(), Month()	取得月，回傳值分別為 Int64 和 Period 型別。
week(), Week()	取得週數 (從年初起算)，回傳值分別為 Int64 和 Period 型別。
day(), Day()	取得日，回傳值分別為 Int64 和 Period 型別。
yearmonth()	取得 (年, 月)，回傳值為 Tuple。
monthday()	取得 (月, 日)，回傳值為 Tuple。
yearmonthday()	取得 (年, 月, 日)，回傳值為 Tuple。
value()	距離公元第一天的總天數



日期與時間的轉換



- DateTime、Date、Time 之間可以互相轉換，但是也有一些限制，整理如下：

轉換	能否轉換	說明
DateTime 轉成 Date	Yes	時、分、秒會被捨去
Date 轉成 DateTime	Yes	時、分、秒會採預設值 0
DateTime 轉成 Time	Yes	
Time 轉成 DateTime	No	
Date 轉成 Time	No	
Time 轉成 Date	No	



日期時間與字串的轉換



- 日期時間與字串之間可以互相轉換，透過格式定義，就可以解析出正確的值並進行轉換。
- 將日期時間轉換為字串，可以透過 `Dates.format()` 函式，例如：`Dates.format(now(), "yyyy-mm-dd HH:MM:SS")` 將目前日期時間轉換為格式中定義的字串。
- 將字串轉換為日期/時間，是透過 `Date` / `DateTime` / `Time` 建構子，配合格式定義完成。
- 將字串轉換為日期/時間，若先利用 `DateFormat` 轉換格式定義，可以大幅提昇執行效能。
- 格式定義的所有定義請參照今日範例。



日期與時間的運算 - 比較



- DateTime 之間、 Date 之間、 Time 之間可以運用比較運算子互相比較。
- DateTime 與 Date 之間可以互相比較，由於精度不同，但只要年、月、日相同就算相同。
- Time 與 DateTime 或是 Date 之間不能互相比較。



日期與時間的運算 - 差異 (減法)



- 2 個日期/時間之間的差異，可以透過減法，在同型別之間可以直接相減。但是不同的型別相減的回傳值單位不同。

型別	減法回傳值單位
DateTime	Millisecond
Date	Day
Time	Nanosecond

- 在不同的型別之間，不能直接加減，但是可以與 Period 型別之物件進行加減。



日期與時間的運算 - 加法



- 日期與時間的加法，可以透過與 Period 型別的物件來進行加法，例如：`now() + Year(2)` 會回傳現在時間的 2 年後。
- Period 型別的物件，可以跨型別運算，例如：`Month(2) + Day(3)`，其回傳的型別為 `Dates.CompoundPeriod`。
- 另外，Period 型別的物件同型別之間支援除法；但是不管是同型別或是跨型別，都是不支援乘法的。



產生時間序列



- 搭配 range 以及 Period 型別，可以很方便地產生時間序列，例如：`Date(2019, 1, 1) : Month(1) : Date(2019, 12, 31)`，宣告了起始日期、區間、結束日期，產生 2019 年每個月 1 日的時間序列，其回傳值型別為 StepRange。
- 透過迭代的方式可以列出序列中的每個日期。
- Date、DateTime、Time 均可以透過相同的方式產生序列。



日期與時間對照函式 (Mapping Function)



- 在實務應用上，我們常會根據日期去得到或查詢某些屬性，例如是星期幾，或是該年是否為閏年... 等，Julia 也提供了許多內建的對照函式可以直接使用。

函式	說明
<code>monthname()</code>	該月份的英文名稱
<code>isleapyear()</code>	該年是否為閏年
<code>dayofyear()</code>	該天是該年的第幾天
<code>quarterofyear()</code>	該天是屬於第幾季
<code>dayofquarter()</code>	該天是該季的第幾天



日期與時間對照函式 (Mapping Function)



- 有關於更多對照函式的實際應用，請參照今日範例。

函式	說明
firstdayofweek()	該週第一天的日期
lastdayofweek()	該週最後一天的日期
lastdayofquarter()	該季最後一天的日期
toprev()	尋找前一個符合條件的日期
tonext()	尋找下一個符合條件的日期



TimeZones.jl



- Julia 本身的 DateTime 不具備時區的概念。
- Time Zones 套件根據 IANA 資料庫，提供 Julia 的使用者取得時區資料，並且可以以類似 DateTime 的方式操作日期與時間資料。
- 安裝及開始使用 Time Zones：

```
In [35]: 1 using Pkg  
          2 Pkg.add("TimeZones")  
  
Resolving package versions...  
Updating `~/.julia/environments/v1.1/Project.toml`  
[no changes]  
Updating `~/.julia/environments/v1.1/Manifest.toml`  
[no changes]  
  
In [36]: 1 using TimeZones
```



TimeZones.jl



- Time Zones 的幾個使用範例：

```
In [27]: 1 # 目前的格林威治時間  
2 now(tz"UTC")
```

```
Out[27]: 2019-08-15T13:48:58.923+00:00
```

```
In [28]: 1 # 目前的日期，在使用目前的日期時須留意不同的時區可能因為跨日的問題而有日期的差異  
2 today(tz"UTC+8")
```

```
Out[28]: 2019-08-15
```

```
In [29]: 1 # 其型別為 ZonedDateTime  
2 typeof(now(tz"UTC"))
```

```
Out[29]: ZonedDateTime
```

```
In [30]: 1 # 透過 ZonedDateTime 建構子建立具有時區的日期，並進行算出不同時區之間的時間差異  
2 # 例如台北時間和格林威治時間差 28800000 毫秒，即為 8 小時  
3 ZonedDateTime(2016, 1, 1, 12, tz"UTC") - ZonedDateTime(2016, 1, 1, 12, tz"Asia/Taipei")
```

```
Out[30]: 28800000 milliseconds
```

知識點 回顧

- 介紹 Julia 內建提供了日期/時間型別的操作，支援日期/時間的建立、轉換、運算。詳細的使用範例請參見今日的程式。
- 由於 Julia 本身的 DateTime 不具備時區的概念，所以如果需要處理時區相關的日期/時間，需藉助其他套件。Time Zones 套件提供 Julia 的使用者取得時區資料，並且可以用類似 DateTime 的方式操作日期與時間資料。



推薦閱讀

- [Dates 模組官方文件](#)
- [Julia - Conversion Functions](#)
- [Julia - Constants](#)





解題時間

請跳出 PDF 至官網 Sample Code
& 作業開始解題