

CS381: Numerical Computation & Softwares

Introduction to Julia programming

Jakramate Bootkrajang

Department of Computer Science

Chiang Mai University



- เป็นภาษาโปรแกรมมิ่งที่ถูกพัฒนาขึ้นมาใหม่ เพื่อรองรับการคำนวณทางวิทยาศาสตร์
- การคำนวณที่ต้องเกี่ยวข้องกับ vector และ matrix สามารถทำได้โดยสะดวก
- เริ่มพัฒนาโดยกลุ่มของนักพัฒนาจากมหาวิทยาลัย MIT
- super fast.
- รองรับภาษาอื่นนอกจากภาษาอังกฤษ (Support unicode characters)
 - ▶ แปลว่าเราสามารถตั้งชื่อตัวแปรเป็นภาษาไทยได้

ถูกนำไปใช้งานในสาย Data science มากมาย

- Finance and Economics <https://lectures.quantecon.org/jl/>
- Machine learning <https://github.com/JuliaML>
- Statistics <https://github.com/JuliaStats>
- Bioinformatics <https://github.com/BioJulia>
- Astronomy <https://github.com/JuliaAstro>


การติดตั้ง Julia

สามารถทำได้สองวิธีคือ

- การติดตั้งบนเครื่องของตนเอง
 - ▶ สามารถ download โปรแกรมแปลภาษา Julia ได้จาก <http://www.julialang.org>
 - ▶ การใช้งานแบบนี้มีความยืดหยุ่นสูงและตอบสนองการใช้งานเร็วกว่า
- การใช้งาน Julia ผ่านเว็บ
 - ▶ www.juliabox.com
 - ▶ จำเป็นต้องมี account ของ google หรือ LinkedIn หรือ Github
 - ▶ ใช้งานได้ทันทีโดยไม่ต้องติดตั้งโปรแกรมบนเครื่องตนเอง

Julia BOX

Run Julia in your
Browser

 Log in with GitHub

 Log in with Google

 Log in with LinkedIn



Jupyter Notebook
Interface



75,000+ users served
since 2015



Free registration,
free to use



Perfect for classes,
students, professors and
new Julia users



Includes 275+ carefully
curated popular Julia
packages



Multi-node deployment
capability



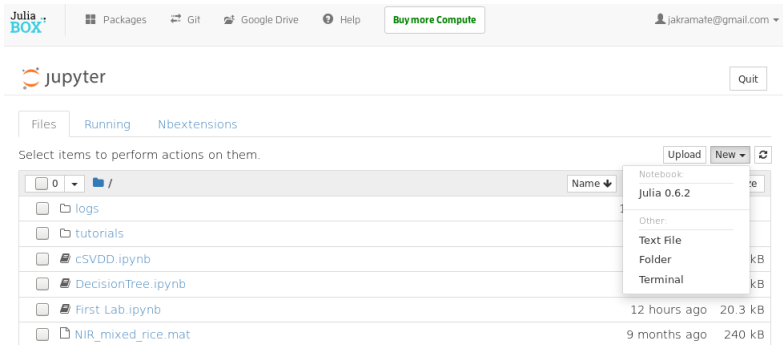
Parallel computing
capability



Buy added memory,
storage, nodes and
enterprise support.

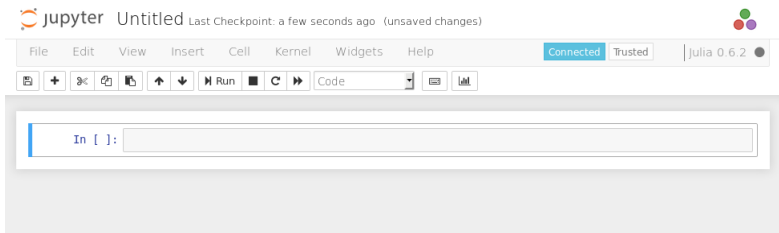
เริ่มสร้างไฟล์ใหม่

- สามารถทำได้โดยกด New -> แล้วเลือกสร้างไฟล์ Julia เวอร์ชัน 0.6.2

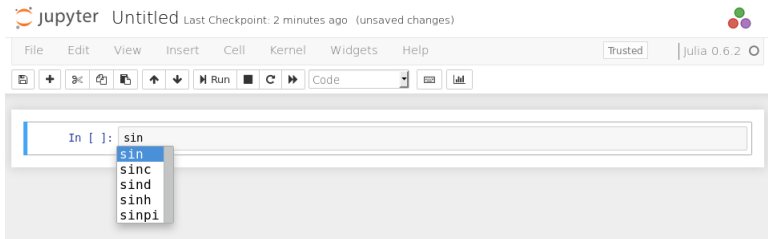


The screenshot shows the JupyterLab interface. At the top, there is a navigation bar with the Julia logo, 'Packages', 'Git', 'Google Drive', 'Help', and a 'Buy more Compute' button. The user's email 'jakramate@gmail.com' is visible in the top right. Below the navigation bar is the 'jupyter' logo and a 'Quit' button. The main area has tabs for 'Files', 'Running', and 'Nbextensions'. A message says 'Select items to perform actions on them.' Below this is a file browser showing a directory structure with folders 'logs' and 'tutorials', and files 'cSVDD.ipynb', 'DecisionTree.ipynb', 'First Lab.ipynb', and 'NIR_mixed_rice.mat'. A 'New' dropdown menu is open, showing options: 'Notebook: Julia 0.6.2', 'Other: Text File', 'Folder', and 'Terminal'. The 'Julia 0.6.2' option is highlighted.

การใช้งานทั่วไป



- พิมพ์คำสั่งที่ต้องการลงในช่อง Input cell
- หากมีหลายคำสั่งสามารถกด Enter เพื่อใส่คำสั่งเพิ่มเติมในบรรทัดใหม่
- หากต้องการประมวลผลคำสั่งทั้งหมดที่ใส่ลงไป ให้กด Ctrl+Enter ผลลัพธ์จะแสดงด้านล่าง



- ระหว่างพิมพ์คำสั่งสามารถกด Tab เพื่อให้ Julia แสดงคำสั่งที่ขึ้นต้นด้วยสิ่งที่พิมพ์ค้างไว้

Primitive data types

- Number:
 - ▶ Integers, Floating points
 - ▶ Special numbers: NaN (Not a number), Inf (Infinity)
- Array:
 - ▶ Vector: Array หนึ่งมิติ
 - ▶ Matrix: Array สองมิติ
- String
- Note: Julia is a case-sensitive language.

- สร้าง array โดยใช้ [] ครอบตัวเลข
 - ▶ $X = [1, 2, 3, 4]$ สร้าง 4x1 vector
 - ▶ $Y = [1 \ 2; 3 \ 4]$ สร้าง 2x2 matrix
- สามารถเรียกดูมิติของ array โดยใช้ฟังก์ชัน `size()`
 - ▶ Vector: Array หนึ่งมิติ
 - ▶ Matrix: Array สองมิติ

Useful array creating functions

- `ones(n,m)`, `zeros(n,m)` สร้างเมทริกซ์ขนาด $n \times m$ ที่มีสมาชิกเป็น 1 หรือ 0 ทั้งหมด
- `eye(m)` สร้าง identity matrix
- `fill(k, n, m)` สร้างเมทริกซ์ขนาด $n \times m$ ที่มีสมาชิกทุกตัวมีค่าเท่ากับ k
- `randn(n,m)` สร้างเมทริกซ์ขนาด $n \times m$ ที่มีสมาชิกทุกตัวสุ่มจาก normal distribution

Array indexing

- `a = randn(4, 4)`
- `a[1, 1]` คือการดึงสมาชิกแถวที่ 1 คอลัมน์ 1
- `a[1, :]` คือการดึงสมาชิกทุกตัวของแถวแรก
- `a[:, 1]` คือการดึงสมาชิกทุกตัวของหลักแรก

Functions on array

$x = [-1 \ 0 \ 1]$

ฟังก์ชัน	ค่าที่ให้	ตัวอย่าง
<code>sum(x)</code>	หาผลบวกของสมาชิกในอะเรย์	<code>y = sum(x)</code>
<code>mean(x)</code>	หาค่าเฉลี่ยของสมาชิกในอะเรย์	<code>m = mean(x)</code>
<code>std(x)</code>	หา s.d. ของสมาชิกในอะเรย์	<code>t = std(x)</code>
<code>var(x)</code>	หา variance ของสมาชิกในอะเรย์	<code>t = var(x)</code>
<code>maximum(x)</code>	หาค่าสูงสุดของอะเรย์	<code>t = maximum(x)</code>
<code>minimum(x)</code>	หาค่าต่ำสุดของอะเรย์	<code>z = minimum(x)</code>
<code>sort(x)</code>	จัดเรียงค่าของสมาชิกในอะเรย์ x	<code>z = sort(x)</code>

Range

- สามารถสร้างลำดับของตัวเลขได้โดย start:stop
 - ▶ ตัวอย่างเช่น 1:10 จะทำการสร้างลำดับเริ่มต้นตั้งแต่ 1 ไปถึง 10
- สามารถกำหนดการเพิ่มของเลขโดยระบุ step เพิ่ม
 - ▶ ตัวอย่างเช่น 1:0.5:10 สร้างลำดับ 1,1.5,2,2.5,...,10
- สามารถเรียกดูลำดับได้โดยใช้ฟังก์ชัน collect()
 - ▶ `collect(1:10)`

Basic operations

สัญลักษณ์	หน้าที่	ตัวอย่าง
=	กำหนดค่าตัวแปร	$y = 5$
+	บวก	$y = x + 1$
-	ลบ	$z = m - 1$
*	คูณ	$k = 2 * 2$
/	หาร	$n = t / 10$
^	ยกกำลัง	$y = x^4$

หมายเหตุ จุดตามด้วยสัญลักษณ์ ใช้กระทำการบนเมทริกซ์แบบตัวต่อตัว

Element-wise operation

- ใช้ `.` นำหน้า operator ทางคณิตศาสตร์
- `ones(2, 2) * ones(2, 2)` คือ Matrix multiplication
- `ones(2, 2) .* ones(2, 2)` คือ Element by element multiplication

Basic maths function

ฟังก์ชัน	ค่าที่ให้	ตัวอย่าง
<code>sqrt(x)</code>	รากที่สองของ x	<code>y = sqrt(x+5)</code>
<code>abs(x)</code>	ค่าสัมบูรณ์ของ x	<code>d = abs(x) * y</code>
<code>sin(x)</code>	ค่าไซน์ของ x (เรเดียน)	<code>t = x + sin(x)</code>
<code>asin(x)</code>	ค่า arcsin ของ x (เรเดียน)	<code>t = x + asin(x)</code>
<code>sind(x)</code>	ค่าไซน์ของ x (degree)	<code>t = x + sind(x)</code>
<code>log(x)</code>	ค่า natural log ของ x	<code>z = log(1+x)</code>
<code>log10(x)</code>	ค่า log ฐาน 10 ของ x	<code>z = log10(1-2*x)</code>
<code>exp(x)</code>	ค่า exponential ของ x (e^x)	<code>p = .7 * exp(x)</code>

Try-out: `result = exp(-2.33) * cos(22 * 180 / pi)`

Boolean expressions

- Expression ที่ประมวลผลเป็น จริง หรือ เท็จ (true / false)

ฟังก์ชัน	ค่าที่ให้	ตัวอย่าง
==	เป็นจริงเมื่อค่าสองค่าเท่ากัน x	y == 5
!=	เป็นจริงเมื่อค่าสองค่าไม่เท่ากัน	7 != 7
<=	เป็นจริงเมื่อค่าด้านซ้ายน้อยกว่าหรือเท่ากับค่าด้านขวา	p <= 40
>=	เป็นจริงเมื่อค่าด้านซ้ายมากกว่าหรือเท่ากับค่าด้านขวา	p >= 2
<	เป็นจริงเมื่อค่าด้านซ้ายน้อยกว่าค่าด้านขวา	x < 5
>	เป็นจริงเมื่อค่าด้านซ้ายมากกว่าค่าด้านขวา	x+y+z > 16
&&	คำเชื่อม และ	x >= 5 && x <= 10
	คำเชื่อม หรือ	x <= 5 x >= 10
!	นิเสธ	!(5 == 5)

Control structure: if

การเลือกทำโดยใช้ if...else

```
if (age >= 18)
    print("You can drive")
else
    print("Sorry you are too young")
end
```

สังเกตว่าจะต้องปิดด้วย end

Control structure: if...elseif

การเลือกทำโดยใช้ if...elseif

```
if (age >= 18)
    print("You can drive")
elseif (age >= 15)
    print("You can ride a motorcycle")
else
    println("Sorry you are too young")
end
```

สังเกตว่าจะต้องปิดด้วย end

Control structure: for

การทำซ้ำโดยใช้ for loop รู้จำนวนรอบในการทำซ้ำที่แน่นอน

```
for i in 1:10      # loop over some range
    println(e^i)   # print with newline
end
```

```
A = [1 3 5 7 9]   # loop over an array
for j in A
    print(j%2)     # print without newline
end
```

สังเกตว่าจะต้องปิดด้วย end

Control structure: while

การทำซ้ำแบบ while ไม่รู้จำนวนรอบที่แน่นอน ทำจนกว่า condition จะเป็นเท็จ

```
while (condition)
    some statements
end
```

```
p = 10 # initialise condition variable
while (p > 0)
    println(p/10)
    p = p - 1 # update condition variable
end
```

- กลุ่มของคำสั่งที่ถูกผูกกันไว้ พร้อมกับถูกตั้งชื่อ
- Useful functions
 - ▶ `print("put this on the screen")`
 - ▶ `println("with new line")`
 - ▶ `readline()`
 - ▶ `typeof()` เรียกดูชนิดของตัวแปร


```
# inline function definition
```

```
f(x) = x3
```

```
df(x) = 3*x2
```

```
# more common function definition
```

```
function cube(x)
```

```
    y = x3           # compute the cube of x
```

```
    return y         # return the result
```

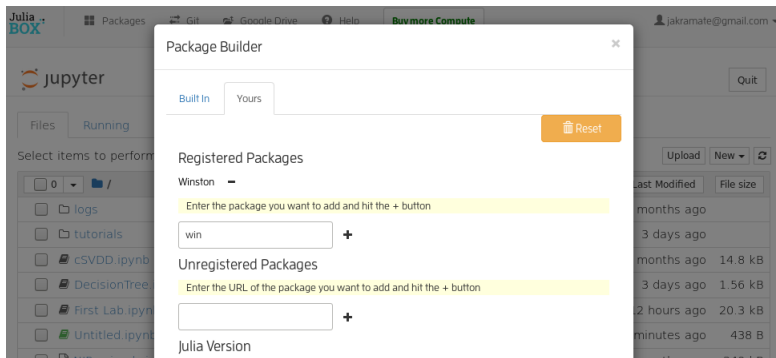
```
end
```

Packages

- กลุ่มของฟังก์ชันที่ใช้ในการประมวลผลข้อมูลเฉพาะทาง
- Julia มี package มากมายสามารถดูข้อมูลได้จาก
 - ▶ Julia package listing — <https://pkg.julialang.org/>
- เช่น Package สำหรับการพลอตกราฟ
- การติดตั้ง Package
 - ▶ ทำได้โดยคำสั่ง `Pkg.add("packageName")` (สำหรับ Julia ที่ติดตั้งบนเครื่อง)
- ก่อนใช้งาน Package ต้องเรียกคำสั่ง `using packageName`

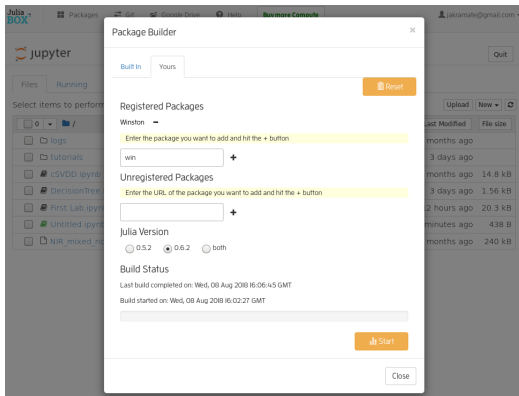
Packages (สำหรับ Juliabox)

- สังเกตมุมซ้ายบนในหน้าหลักของ Juliabox จะมี ปุ่ม Packages สำหรับเพิ่ม package
- เมื่อคลิกจะเปิดหน้าต่างสำหรับเพิ่ม package ให้พิมพ์ชื่อ package ที่ต้องการเพิ่มในช่อง Registered package แล้วกดเครื่องหมายบวก



Packages (สำหรับ Juliabox)

- กด Start เพื่อให้โปรแกรมติดตั้ง package



Useful packages: Plotting

- การสร้างกราฟใน Julia สามารถทำได้หลายวิธี มี package รองรับอยู่มาก
- ที่เราสนใจเป็นพิเศษคือ package ที่ชื่อว่า Winston ซึ่งเป็น package สำหรับ plot กราฟที่ใช้
งานง่าย
- ให้นักศึกษาทดลองติดตั้ง package ที่ชื่อว่า Winston ดู

การเรียกใช้งาน Winston

```
In [1]: using Winston
```

```
In [ ]: x = [1,2,3,4]
```

```
In [ ]: y1 = x.^2  
        y2 = x.^3  
  
        plot(x, y1, "r")  
        oplot(x, y2, "b")  
  
        title("my first graph")  
        xlabel("x")  
        ylabel("y")
```

Plotting function

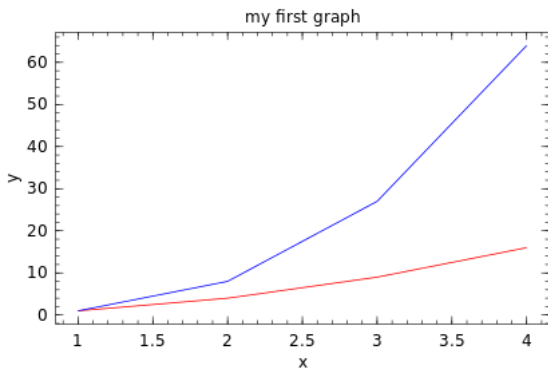
- `plot()` : ใช้พล็อตเส้น โดยมี parameter 3 ตัว คือ เวกเตอร์ x , เวกเตอร์ y และ string ที่บอกลักษณะเส้น
- `oplot()` : ใช้พล็อตเส้นทับบนกราฟเดิม โดยมี parameter 3 ตัว คือ เวกเตอร์ x , เวกเตอร์ y และ string ที่บอกลักษณะเส้น
- `title()` ใช้กำหนดหัวกราฟ `xlabel()` ใช้กำหนดชื่อแกน x และ `ylabel()` ใช้กำหนดชื่อแกน y

String บอกลักษณะเส้น

- “[color][style]”
- โดยที่ [color] เป็นรหัสสี เช่น
 - ▶ r = สีแดง, b = สีน้ำเงิน, k = สีดำ เป็นต้น
- และ [style] เป็นลักษณะเส้นหรือ marker เช่น
 - ▶ o แทน marker วงกลม , - แทน เส้นทึบ, : แทนเส้นประ เป็นต้น
- Reference:

<http://winston.readthedocs.io/en/latest/fun/plot.html>

The result



Must read

- Learn X in Y minutes for Julia

`https://learnxinyminutes.com/docs/julia/`