

## Basic program R

นาย สัจจา ดวงชัยอยู่สุข  
มหาวิทยาลัยหอการค้าไทย

©Kilenthong 2019

# Main Issues

- basic program
- basic code for program R

ในการศึกษาเรื่องความน่าจะเป็น ต้องอาศัย ความรู้ในเรื่องของ Set ดั้งนั้นก่อนที่ จะเข้าเรื่องจึง ต้องเข้าใจพื้นฐานที่จำเป็นต้องใช้ก่อน

# หลักการเบื้องต้น

- R เป็นซอฟต์แวร์สำหรับการจัดการข้อมูลการคำนวณและการแสดงผลทางกราฟิก
- เก็บในชุดของการคำนวณในแบบ arrays
- เป็นภาษาพัฒนาได้ง่ายและมีประสิทธิภาพ
- ในคลาสนี้ จะใช้ RStudio เป็นโปรแกรมช่วยเขียนภาษา R

# หน้าต่างโปรแกรม

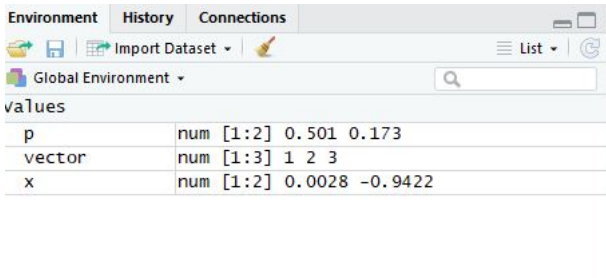
The screenshot displays the RStudio interface with three main panes:

- Source Editor (Top Left):** Contains the text "R Script ไว้เขียน code และ save เก็บเป็นไฟล์ .R".
- Environment Pane (Top Right):** Shows "Global Environment" and the text "Environment ไว้แสดงตัวแปรที่สร้าง".
- Console (Bottom):** Displays the R startup message and the text "Console ไว้แสดงผลต่างๆ".

# ฟังก์ชันพื้นฐาน

## ฟังก์ชันพื้นฐานที่ใช้ในวันนี้

- # คือ comment เวลาที่ไม่ต้องการให้ รันบรรทัดนั้น
- c () ใช้สร้างเวกเตอร์ เช่น `vector = c(1,2,3,5,6)`
- `rnorm(n,mean,sd)` ใช้สุ่มตัวเลข(ค่า z) n จำนวน โดยใช้ normal distribution เช่น `x=rnorm(1,mean=0,sd=1)`
- `pnorm(z, mean, sd)` ใช้เปลี่ยนค่าเป็นพื้นที่ใต้กราฟ  $P(x \leq z)$  เช่น `p=pnorm(0, mean = 0, sd = 1)`



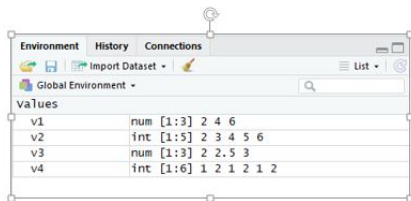
The screenshot shows the R Studio Environment pane with the following content:

values	
p	num [1:2] 0.501 0.173
vector	num [1:3] 1 2 3
x	num [1:2] 0.0028 -0.9422

# ฟังก์ชันพื้นฐาน

ฟังก์ชันพื้นฐานที่ใช้ในวันนี้

```
> v1=c(2,4,6)
> v2=2:6
> v3=seq(2,3,by=0.5)
> v4=rep(1:2,times=3)
> rep(1:2,each=3)
[1] 1 1 1 2 2 2
```



Environment	History	Connections
Global Environment		
Values		
v1	num [1:3]	2 4 6
v2	int [1:5]	2 3 4 5 6
v3	num [1:3]	2 2.5 3
v4	int [1:6]	1 2 1 2 1 2

## การแสดงผลค่า

การแสดงผลค่าต่างๆ นั้นมีหลายคำสั่ง ซึ่งจะแสดงอยู่ในหน้าต่าง console

- print(), show() จะแสดงผลในหน้าต่าง console
- View() จะแสดงผลออกมาในอีกหน้าต่างที่เป็นส่วนของ R script



The screenshot shows the R Studio interface. The top pane displays a data frame with two rows and two columns. The first row contains the values 0.5011151 and 0.1730492. The second row contains the values 1 and 2. The bottom pane shows the R console with the following commands and their outputs:

```
> vector[1]
numeric(0)
> vector[1]
[1] 1
> vector[2]
[1] 2
> vector[3]
[1] 3
> vector[4]
[1] NA
> vector[5]
[1] NA
> print(p)
[1] 0.5011151 0.1730492
> show(p)
[1] 0.5011151 0.1730492
> View(p)
> |
```



## การเขียนเงื่อนไข(if-else)

คำสั่งในการตรวจสอบเงื่อนไขใช้IF

```
if (condition) {
```

```
# code executed when condition is TRUE
```

```
} else {
```

```
# code executed when condition is FALSE
```

```
}
```

ตัวอย่าง

```
6 vector = c( 1,2)
7 r=0.5
8 if (r>0){
9   print(vector[1])
10
11 }else{
12   print(vector[2])
13 }
```

→

```
> vector = c( 1,2)
> r=0.5
> if (r>0){
+   print(vector[1])
+ }else{
+   print(vector[2])
+ }
[1] 1
```

```
6 vector = c( 1,2)
7 r=-0.5
8 if (r>0){
9   print(vector[1])
10
11 }else{
12   print(vector[2])
13 }
```

→

```
> vector = c( 1,2)
> r=-0.5
> if (r>0){
+   print(vector[1])
+ }else{
+   print(vector[2])
+ }
[1] 2
```

# การเขียนเงื่อนไข(if-else)


การเขียนเงื่อนไขมีดังนี้

condition	meaning
$a=b$	are equal
$a!=b$	not equal
$a>b$	greater than
$a<b$	less than
$a>=b$	greater than or equal to
$a<=b$	less than or equal to
<code>is.na(a)</code>	is missing
<code>is.null(a)</code>	is null

## ลูป (for)

- คำสั่งในการวนลูป คือ  
for (ตัวแปร in เงื่อนไขการวนลูป ) {  
# code executed  
}
- ตัวอย่าง

```
15 for(i in 1:5){  
16   show(i)  
17 }
```



```
> for(i in 1:5){  
+   show(i)  
+ }  
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5
```

## ลูป (while)

- คำสั่งในการวนลูป คือ  
while (เงื่อนไขการวนลูป) {  
# code executed  
}
- ตัวอย่าง

```
i <- 1  
while (i < 6) {  
  print(i)  
  i = i+1  
}
```

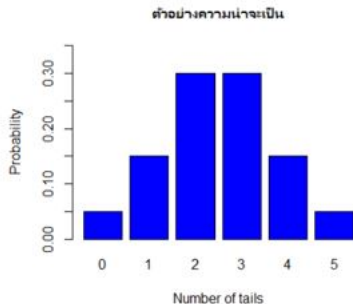


```
> i <- 1  
> while (i < 6) {  
+   print(i)  
+   i = i+1  
+ }  
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5
```

## plot graph: bar plot

- คำสั่งในการสร้าง bar graph คือ  
barplot(y,names.arg=x,xlab=ข้อความ แกน x  
,ylab=ข้อความ แกน y,col=สีของแท่ง เช่น "red"  
,main=ข้อความ title  
,ylim=c(0,0.35), border=สีขอบของแท่ง เช่น "black")
- ตัวอย่าง

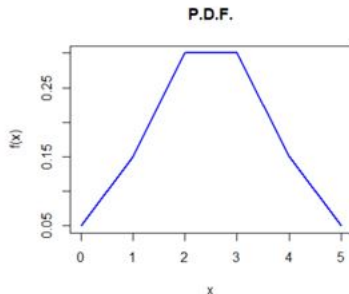
```
> # Plot the bar chart
> x=c(0,1,2,3,4,5)
> y=c(0.05,0.15,0.3,0.3,0.15,0.05)
> barplot(y,names.arg=x,xlab="Number of tails"
+         ,ylab="Probability",col="blue"
+         ,main="ตัวอย่างความน่าจะเป็น"
+         ,ylim=c(0,0.35), border="black")
```



## plot graph:line graph

- คำสั่งในการสร้าง bar graph คือ  
barplot(x,y,xlab=ข้อความ แกน x  
,ylab=ข้อความ แกน y,col=สีของแท่ง เช่น "red"  
,main=ข้อความ title  
,ylim=c(0,0.35), lwd=ระดับความหนาของเส้น)
- ตัวอย่าง

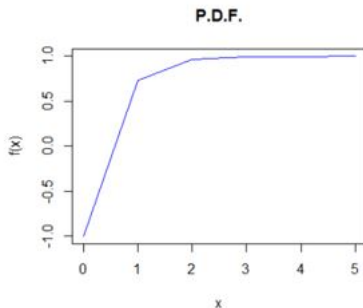
```
> plot(x, y,  
+      main="P.D.F.",  
+      ylab="f(x)",  
+      type="l",  
+      col="blue"  
+      ,lwd=2)
```



## plot graph:line graph of function

- หมายเหตุ y สามารถเขียนเป็นฟังก์ชันได้
- ตัวอย่าง

```
plot(x, 1-2*exp(-2*x),  
     main="P.D.F.",  
     ylab="f(x)",  
     type="l",  
     col="blue")
```



## plot graph:line graph of multiple function

- ถ้าจำวาดกราฟหลายเส้น ในกราฟเดียวกันให้ใช้คำสั่ง “line” หรือ “points” แทน “plot” สำหรับเส้นถัดไป
- ตัวอย่าง

```
> x1=seq(-10,-1,0.2)
> x2=seq(-1,1,0.2)
> x3=seq(1,3,0.2)
> x4=seq(3,4,0.2)
> x5=seq(4,10,0.2)
> x6=seq(-10,3,0.2)
> plot(x1,0*x1,type="l",lwd=4,xlim=c(-9,10),
+      ylim=c(0,1.01),main="C.D.F (F(x))",xlab="x",ylab="F(x)")
> lines(x2,0.25+0.25*x2,lwd=4,col="blue")
> lines(x3,0.5+0*x3,lwd=4,col="red")
> lines(x4,-1+0.5*x4,lwd=4,col="green")
> points(x5,1+0*x5,lwd=1,col="black")
> lines(x6,0.5+0*x6,lty=2,col="red")
```

