บทที่ 2

การจัดการข้อมูลและหน่วยความจำ

2.1 ชนิดข้อมูลในภาษา R

ในการใช้งาน R นั้น จำเป็นอย่างยิ่งที่จะต้องทำความเข้าใจกับชนิดข้อมูลที่ใช้ในการเขียนโปรแกรมก่อน เนื่องจากชนิดข้อมูลของภาษา R นั้นมีหลายชนิด และมีรูปแบบการเขียนโปรแกรมที่แตกต่างจากโปรแกรมภาษาอื่นพอสมควร โดยชนิดข้อมูลในภาษา R สามารถแบ่งได้ 2 ประเภท คือ ประเภทข้อมูลเดี่ยว (single type) เป็นข้อมูลที่ไม่สามารถแบ่งแยกได้ ได้แก่ ชนิดข้อมูลจำนวนเต็ม ชนิดข้อมูลจำนวนจริง ชนิดข้อมูลตรรกศาสตร์ และชนิดข้อมูลตัวอักษร เป็นต้น และประเภทข้อมูลชุด เป็นข้อมูลที่เกิดจากนำเอาข้อมูลชนิดเดี่ยวมากกว่า 1 ชนิดมาสร้างเป็นตัวแปรชุดเพื่อนำไปใช้ในการวิเคราะห์ข้อมูลต่อได้ ซึ่งรายละเอียดจะนำเสนอในหัวข้อถัดไป ดังนี้

2.1.1 ประเภทข้อมูลเดี่ยว (single type) หมายถึง ข้อมูลขนาดเล็กที่สุดที่ไม่สามารถแบ่งย่อยลงไปได้อีก จึงไม่สามารถนำไปใช้คำนวณทางสถิติได้ ประกอบด้วยข้อมูลประเภทต่าง ๆ ดังมีรายละเอียดดังนี้

2.1.1.1 ข้อมูลตัวเลข นื่องจาก R เป็นภาษาที่เน้นการนำไปใช้งานทางสถิติเป็นหลัก การเก็บตัวเลขแบบจุดทศนิยม (ในภาษา R เรียกข้อมูลชนิดนี้ว่า numeric) จึงเป็นมาตรฐานที่ R เลือกใช้เพราะสามารถนำไปใช้คำนวณต่อได้ง่าย นอกจากนี้ R ยังมีข้อมูลตัวเลขอีกสองประเภทสำหรับเลือกใช้ตามความเหมาะสม ซึ่งก็คือ

1) จำนวนเต็ม (integer) สามารเขียนด้วยการพิมพ์ตัวอักษร L ไว้ข้างหลังตัวเลข เช่น 100L ซึ่งจำนวนเต็มมีไว้สำหรับการคำนวณต่าง ๆ เป็นแบบวิยุตคณิต (discrete mathematics) หรือใช้เพื่อบ่งบอกจำนวนรอบการวนซ้ำ

2) จำนวนเชิงซ้อน (complex) สามารเขียนด้วยการพิมพ์ตัวอักษร i ไว้ข้างหลังตัวเลข เช่น 1i, 3+4i จำนวนเชิงซ้อนมีไว้สำหรับงานทางด้านวิทยาศาสตร์-วิศวกรรมบางแขนง เช่น การคำนวณด้านแม่เหล็กไฟฟ้า พลศาสตร์ของไหล

ข้อสังเกตุสำหรับการลำดับของตัวเลขในภาษา R จะเรียงจากต่ำไปสูง ได้แก่ จำนวนเต็ม จำนวนทศนิยม และจำนวนเชิงซ้อน หากมีการใช้งานตัวเลขคละประเภทกัน R จะแปลงตัวเลขที่อยู่ในชั้นต่ำกว่าขึ้นไปเป็นชั้นที่สูงกว่าเสมอ

2.1.1.2 ข้อมูลตัวอักษร ภาษา R ไม่แบ่งแยกข้อมูลตัวอักษรเป็นหลายระดับ (แบบภาษา C) แต่จะมองว่าตัวอักษรชุดหนึ่งไม่ว่าจะยาวเท่าใดคือหนึ่งข้อความเสมอ (แบบภาษา Python) และเรียกข้อมูลชนิดนี้ว่า character การประกาศข้อความสามารถใช้ได้ทั้งอัญประกาศเดี่ยว (') หรืออัญประกาศคู่ (") ล้อมรอบข้อความที่ต้องการโดยไม่มีความแตกต่างกันแต่อย่างใด

2.1.1.3 ข้อมูลตรรกะ การแสดงข้อมูลตรรกะ (logical) ว่าถูกหรือผิด จะใช้คำว่า TRUE กับ FALSE ตามลำดับ หรือสามารถป้อนแค่ตัวอักษรตัวแรก (T, F) เข้าไปได้ แต่ไม่สามารถใช้ตัวเลขทดแทนได้ นอกจากนี้ R ยังมีข้อมูลตรรกะแบบพิเศษอีกตัวหนึ่งเพิ่มเข้ามา คือ ข้อมูลว่าง (NA) ซึ่งจะเกิดขึ้นเมื่อพยายามเข้าถึงข้อมูลในตำแหน่งที่ไม่มีข้อมูล หรือพยายามคำนวณค่าทั้งที่ไม่มีข้อมูลอยู่ หมายเหตุว่า R จะไม่ถือว่าเกิดข้อผิดพลาดขึ้นจนต้องหยุดการคำนวณ แต่อาจขึ้นคำเตือนบางครั้งเมื่อพยายามทำงานบนข้อมูลที่ไม่สมบูรณ์นี้

2.1.2 ประเภทข้อมูลชุด หมายถึง ข้อมูลที่เก็บรวบรวมข้อมูลเดี่ยวหลาย ๆ ประเภทมาไว้ในที่เดียวกัน ทำให้สามารถนำไปวิเคราะห์เชิงสถิติต่อได้ ในภาษา R มีประเภทข้อมูลชุดอยู่หลายประเภท แต่มีหลัก ๆ ดังนี้

2.1.2.1 เวกเตอร์ (vectors) เวกเตอร์เป็นการเก็บข้อมูลประเภทเดียวกันทั้งหมด เช่น ข้อมูลประเภทตัวเลข (numeric) ข้อมูลประเภทข้อความ (text) เป็นต้น

2.1.2.2 ลิสต์ (lists) ลิสต์จะต่างกับเวกเตอร์ตรงที่สามารถเก็บข้อมูลหลายประเภทได้พร้อมกันซึ่งเกิดจากการนำเวกเตอร์หลาย ๆ ตัวมาต่อกันเป็นลิสต์

2.1.2.3 ดาต้าเฟรม (data frames) เป็นการเก็บข้อมูลให้อยู่ในรูปแบบคล้ายกับตาราง ซึ่งมีแถว (row) และคอลัมน์ (column) ข้อมูลที่เก็บในดาต้าเฟรมไม่จำเป็นต้องเป็นประเภทเดียวกัน

2.1.2.4 เมตริกซ์ (matrices) มีลักษณะการเก็บข้อมูลคล้ายกับดาต้าเฟรม แต่ข้อมูลจะเป็นประเภทเดียวกันและเป็นประเภทตัวเลขเพื่อนำไปใช้ในการคำนวณต่าง ๆ

2.1.2.5 แฟคเตอร์ (factor) เป็นโครงสร้างข้อมูลที่ใช้สำหรับฟิลด์ที่ต้องใช้เฉพาะค่าที่กำหนดไว้ล่วงหน้าเท่านั้น (ข้อมูลหมวดหมู่) โดยปกติค่าข้อมูลจะเป็นตัวอักษระ

โดยข้อมูลรายละเอียดการใช้งานประเภทข้อมูลชุดจะอธิบายในบทที่ 6 ต่อไป

2.2 คำสงวนในภาษา R

ในทุกโปรแกรมภาษาทุกโปรแกรมจะมีกลุ่มคำที่มีความหมายพิเศษซึ่งไม่สามารถนำมาใช้ในการตั้งชื่อตัวแปร หรือฟังก์ชันได้ซึ่งเรียกว่า คำสงวน (Reserve Word) ในโปรแกรมภาษา R ตัวอย่างคำสงวน ได้แก่

if	else	repeat	while	function
for	in	next	break	TRUE
FALSE	NULL	Inf	NaN	NA
NA_integer_	NA_real_	NA_complex_	NA_character_	

2.2 ตัวแปรและค่าคงที่

2.2.1 ตัวแปร หมายถึง ชื่อตำแหน่งที่ใช้จองพื้นที่ในในหน่วยความจำที่ใช้สำหรับการเก็บข้อมูลซึ่งค่าของตัวแปรสามารถเปลี่ยนแปลงได้ตามความต้องการ ซึ่งในการตั้งชื่อของตัวแปรนั้นจะต้องไม่ซ้ำและต้องพิจารณาให้ถูกต้องตามกฎการตั้งชื่อตัวแปร

2.2.2 กฎการตั้งชื่อตัวแปร

1) สามารถตั้งชื่อด้วยตัวหนังสือ ตัวเลข หรือเครื่องหมายมหัพภาค (.) และเครื่องหมายสัญประกาศ (_) ได้

2) จะต้องเริ่มต้นด้วยตัวอักษร หรือเครื่องหมายมหัพภาค (.) แต่ถ้าเริ่มต้นด้วยเครื่องหมายมหัพภาค (.) จะไม่สามารถตามด้วยตัวเลขได้

3) ไม่สามารถใช้คำสงวน (reserved word) ในการตั้งชื่อตัวแปรได้

ตัวอย่างการตั้งชื่อตัวแปรที่ถูกต้อง

```
total, Sum, .fine.with.dot, this_is_acceptable, Number5
```

ตัวอย่างการตั้งชื่อตัวแปรที่ไม่ถูกต้อง

```
tot@l, 5um, _fine, TRUE, .0ne
```

2.2.3 การประกาศตัวแปรในภาษา R

การประกาศตัวแปรในภาษา R จะใช้สัญลักษณ์ที่แตกต่างจากโปรแกรมภาษาอื่นซึ่งจะไม่มีการกำหนดชนิดของตัวแปรเบื้องต้น แต่ชนิดของตัวแปรจะถูกกำหนดโดยค่าที่กำหนดให้กับตัวแปร โดยทั่วไปจะใช้สัญลักษณ์ลูกศรซ้าย (<-) ดังเช่นตัวอย่างการประกาศตัวแปรต่อไปนี้

1) ข้อมูลชนิดจำนวนเต็ม

รูปแบบการใช้งาน:

```
> ชื่อตัวแปร <- ค่าของตัวแปร
```

ตัวอย่างเช่น

```
> A <- 4
```

2) ข้อมูลชนิดจำนวนจริง

รูปแบบการใช้งาน:

```
> ชื่อตัวแปร <- ค่าของตัวแปร
```

ตัวอย่างเช่น

```
> B <- 4.15
```

3) ข้อมูลชนิดตัวอักษร

รูปแบบการใช้งาน:

```
> ชื่อตัวแปร <- “ค่าของตัวแปร”
```

ตัวอย่างเช่น

```
> N <- “Kairung”
```

4) ข้อมูลชนิดตรรกะ

รูปแบบการใช้งาน:

```
> ชื่อตัวแปร <- ค่าของตัวแปร
```

ตัวอย่างเช่น

```
> C <- TRUE
```

2.2.4 การกำหนดค่าคงที่

ค่าคงที่ หมายถึง ค่าที่ไม่สามารถเปลี่ยนแปลงได้ดังเช่นชื่อของมัน ซึ่งค่าคงที่สามารถเป็นได้ทั้งตัวเลขและตัวอักษร การกำหนดค่าคงที่สามารถแสดงได้ในตัวอย่างดังต่อไปนี้

1) ค่าคงที่ที่เป็นตัวเลขจำนวนเต็ม เป็นค่าคงที่ของข้อมูลชนิดจำนวนเต็มจะใช้ตัวอักษร L ต่อท้ายตัวเลขที่ต้องการกำหนดให้เป็นค่าคงที่ของตัวแปร ซึ่งมีรูปแบบคล้ายกันกับการประกาศตัวแปร มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> ชื่อตัวแปร <- ค่าของตัวแปรL
```

ตัวอย่างเช่น

```
> C <- 5L
```

2) ค่าคงที่ที่เป็นจำนวนจริง เป็นค่าคงที่ของข้อมูลชนิดจำนวนจริงจะใช้ตัวอักษร i ต่อท้ายตัวเลขที่ต้องการกำหนดให้เป็นค่าคงที่ของตัวแปร มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> ชื่อตัวแปร <- ค่าของตัวแปรi
```

ตัวอย่างเช่น

```
> D <- 5i
```

3) ค่าคงที่อักขระ เป็นค่าคงที่ของอักขระหรือข้อความสามารถแสดงโดยใช้เครื่องหมายอัญประกาศเดี่ยว (') หรือเครื่องหมายอัญประกาศคู่ (") เป็นตัวคั่นค่าคงที่ได้เลย ดังแสดงในตัวอย่างการใช้งาน มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> “ค่าคงที่” หรือ ’ค่าคงที่’
```

ตัวอย่างเช่น

```
> “kairung” หรือ ’kairung’
```

2.2.4 คำสั่งสำหรับการตรวจสอบชนิดของข้อมูล

1) typeof() เป็นคำสั่งที่ใช้ในการตรวจสอบชนิดข้อมูลของตัวแปร มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> typeof(ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> A<-10
> typeof(A)
[1] "double"
```

2) is.integer() ใช้ตรวจสอบชนิดข้อมูลของตัวแปรว่าเป็นข้อมูลชนิดจำนวนเต็มหรือไม่ มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> is.integer(ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> B<-12L
>  is.integer(B)
[1] TRUE
```

3) is.double() ใช้ตรวจสอบชนิดข้อมูลของตัวแปรว่าเป็นข้อมูลชนิดจำนวนจริงหรือไม่ มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> is.double(ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> C<-10
> is.double(C)
[1] TRUE
```

4) is.numeric() ใช้ตรวจสอบชนิดข้อมูลของตัวแปรว่าเป็นข้อมูลชนิดตัวเลขหรือไม่ มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> is.numeric(ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> D<-10
> is.numeric(D)
[1] TRUE
```

5) is.function() ใช้ตรวจสอบชนิดข้อมูลของตัวแปรว่าเป็นฟังก์ชันหรือไม่ มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> is.function(ชื่อฟังก์ชัน)
```

ตัวอย่างเช่น

```
> is.function(sum)
[1] TRUE
```

6) is.matrix() ใช้ตรวจสอบชนิดข้อมูลของตัวแปรว่าเป็นข้อมูลชนิดเมทริกซ์หรือไม่ มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> is.matrix (ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> x <- matrix(1:9, nrow = 3, dimnames = list(c("X","Y","Z"),
c("A","B","C")))
> x
  A B C
X 1 4 7
Y 2 5 8
Z 3 6 9
> is.matrix(x)
[1] TRUE
```

2.2.5 คำสั่งสำหรับแปลงชนิดข้อมูล

1) as.character **เป็นคำสั่งที่ใช้สำหรับแปลงข้อมูลเป็นตัวอักษร** มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> as.character (ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> Y=10
> typeof(Y)
[1] "double"
> C<-as.character(Y)
> typeof(C)
[1] "character"
```

2) as.integer **เป็นคำสั่งที่ใช้สำหรับแปลงข้อมูลเป็นตัวเลขจำนวนเต็ม** มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> as.integer (ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> Y=10
> C<-as.character(Y)
> typeof(C)
[1] "character"
> Z=as.integer(C)
> typeof(Z)
[1] "integer"
```

3) as.double เป็นคำสั่งที่ใช้สำหรับแปลงข้อมูลเป็นตัวเลขจำนวนจริง มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> as.double(ชื่อตัวแปร)
```

ตัวอย่างเช่น

```
> Y=20L
> X=as.double(Y)
> typeof(X)
[1] "double"
```

4) as.numeric เป็นคำสั่งที่ใช้สำหรับแปลงข้อมูลเป็นตัวเลข มีรูปแบบการเขียนดังนี้

รูปแบบการใช้งาน:

```
> as.numeric(ชื่อฟังก์ชัน)
```

ตัวอย่างเช่น

```
> H<-"10"
> typeof(H)
[1] "character"
> I=as.numeric(H)
> typeof(I)
[1] "double"
> is.numeric(I)
[1] TRUE
```

2.3 ตัวดำเนินการ

ตัวดำเนินการเป็นสัญลักษณ์ที่ใช้บอกตัวแปลภาษาให้ดำเนินการคำนวณหรือจัดการด้านตรรกศาสตร์ให้กับโปรแกรม ซึ่งในภาษา R ประกอบด้วยตัวดำเนินการหลายประเภท ได้แก่ 1) ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators) 2) ตัวดำเนินการด้านความสัมพันธ์ (Relational Operators) 3) ตัวดำเนินการด้านตรรกศาสตร์ (Logical Operators) 4) ตัวดำเนินการ

ในการกำหนดค่า (Assignment Operators) และ 5) ตัวดำเนินการอื่นๆ (Miscellaneous Operators) ซึ่งจะแสดงรายละเอียดในหัวข้อถัดไป

2.3.1 ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators)

ตัวดำเนินการทางคณิตศาสตร์ (Arithmetic Operators) เป็นตัวดำเนินการที่ใช้ในการคำนวณทางคณิตศาสตร์ ดังแสดงในตารางที่ 2.1 ซึ่งจะแสดงสัญลักษณ์ตัวดำเนินการทางคณิตศาสตร์และตัวอย่างผลลัพธ์ที่ได้การทำงานของตัวดำเนินการที่ใช้ในภาษา R

ตารางที่ 2.1 สัญลักษณ์ตัวดำเนินการทางคณิตศาสตร์และตัวอย่างผลลัพธ์

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
+	การบวก	v <- c(2,5.5,6) t <- c(8, 3, 4) print(v+t) ผลลัพธ์ [1] 10.0 8.5 10.0
–	การลบ	v <- c(2,5.5,6) t <- c(8, 3, 4) print(v-t) ผลลัพธ์ [1] -6.0 2.5 2.0
*	การคูณ	v <- c(2,5.5,6) t <- c(8, 3, 4)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
		print(v*t) ผลลัพธ์ [1] 16.0 16.5 24.0
/	การหาร	v <- c(2,5.5,6) t <- c(8, 3, 4) print(v/t) ผลลัพธ์ [1] 0.250000 1.833333 1.500000
^ หรือ **	การยกกำลัง	v <- c(2,5.5,6) t <- c(8, 3, 4) print(v^t) ผลลัพธ์ [1] 256.000 166.375 1296.000
%%	การเก็บเศษที่ได้จากการหาร	v <- c(2,5.5,6) t <- c(8, 3, 4) print(v%%t)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
		ผลลัพธ์ [1] 2.0 2.5 2.0
%/%	การหารที่ได้จำนวนเต็ม	v <- c(2,5.5,6) t <- c(8, 3, 4) print(v%/%t) ผลลัพธ์ [1] 0 1 1

2.3.2 ตัวดำเนินการความสัมพันธ์ (Relational Operators)

ตัวดำเนินการความสัมพันธ์ (Relational Operators) เป็นตัวดำเนินการที่ใช้สำหรับการเปรียบเทียบ โดยจะแสดงการทำงานในตารางที่ 2.2 ซึ่งจะแสดงสัญลักษณ์ที่ใช้สำหรับตัวดำเนินการความสัมพันธ์และผลลัพธ์ที่ได้จากการทำงาน ตัวดำเนินการความสัมพันธ์ในภาษา R จะใช้ตัวอย่างของข้อมูลประเภทเวกเตอร์ 2 เวกเตอร์ และแสดงผลลัพธ์เป็นค่าตรรกศาสตร์ (TRUE หรือ FALSE)

ตารางที่ 2.2 สัญลักษณ์ดำเนินการความสัมพันธ์และตัวอย่างผลลัพธ์

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
>	การมากกว่า	v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v>t) ผลลัพธ์ [1] FALSE TRUE FALSE FALSE

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
<	การน้อยกว่า	v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v < t) ผลลัพธ์ [1] TRUE FALSE TRUE FALSE
==	การเท่ากับ	v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v == t) ผลลัพธ์ [1] FALSE FALSE FALSE TRUE
<=	การน้อยกว่าเท่ากับ	v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v<=t) ผลลัพธ์ [1] TRUE FALSE TRUE TRUE
>=	การมากกว่าเท่ากับ	v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v>=t)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
		ผลลัพธ์ [1] FALSE TRUE FALSE TRUE
!=	การปฏิเสธไม่เท่ากับ	v <- c(2,5.5,6,9) t <- c(8,2.5,14,9) print(v!=t) ผลลัพธ์ [1] TRUE TRUE TRUE FALSE

2.3.3 ตัวดำเนินการเชิงตรรกะ (Logical Operators)

ตัวดำเนินการเชิงตรรกะ (Logical Operators) เป็นตัวดำเนินการที่ใช้สำหรับเปรียบเทียบข้อมูล 2 เวกเตอร์และใช้กับข้อมูลประเภทตรรกะ ข้อมูลจำนวนเต็ม ข้อมูลจำนวนจริง ซึ่งผลลัพธ์ของการเปรียบเทียบจะแสดงเป็นค่าตรรกะ (TRUE หรือ FALSE) ดังแสดงในตาราง 2.3 ซึ่งสัญลักษณ์ตัวดำเนินการดำเนินการเชิงตรรกะและตัวอย่างผลลัพธ์ที่ได้การทำงานของตัวดำเนินการที่ใช้ในภาษา R

ตารางที่ 2.3 สัญลักษณ์ตัวดำเนินการทางตรรกะและตัวอย่างผลลัพธ์

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
&	AND	v <- c(3,1,TRUE,2+3i) t <- c(4,1,FALSE,2+3i) print(v&t) ผลลัพธ์ [1] TRUE TRUE FALSE TRUE

ตารางที่ 2.3 สัญลักษณ์ตัวดำเนินการทางตรรกะและตัวอย่างผลลัพธ์ (ต่อ)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
|	OR	v <- c(3,0,TRUE,2+2i) t <- c(4,0,FALSE,2+3i) print(v|t) ผลลัพธ์ [1] TRUE FALSE TRUE TRUE
!	NOT	v <- c(3,0,TRUE,2+2i) print(!v) ผลลัพธ์ [1] FALSE TRUE FALSE FALSE

ตัวดำเนินการเชิงตรรกะ AND (&&) และ OR (||) จะพิจารณาค่าแรกของเวกเตอร์และแสดงผลลัพธ์ออกมา ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 ตัวดำเนินการเชิงตรรกะ AND (&&) และ OR (||)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
&&	AND	v <- c(3,0,TRUE,2+2i) t <- c(1,3,TRUE,2+3i) print(v&&t) ผลลัพธ์ [1] TRUE

ตารางที่ 2.4 ตัวดำเนินการเชิงตรรกะ AND (&&) และ OR (||) (ต่อ)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
||	OR	v <- c(0,0,TRUE,2+2i) t <- c(0,3,TRUE,2+3i) print(v||t) ผลลัพธ์ [1] FALSE

2.3.4 ตัวดำเนินการในการกำหนดค่า (Assignment Operators) ใช้เพื่อกำหนดค่าให้กับตัวแปร สามารถแสดงรายละเอียดและการทำงานได้ดังตารางที่ 2.5

ตารางที่ 2.5 ตัวดำเนินการเชิงตรรกะ AND (&&) และ OR (||)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
<– or = or <<–	กำหนดค่าทางซ้าย	v1 <- c(3,1,TRUE,2+3i) v2 <<- c(3,1,TRUE,2+3i) v3 = c(3,1,TRUE,2+3i) print(v1) print(v2) print(v3) ผลลัพธ์ [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i

ตารางที่ 2.5 ตัวดำเนินการเชิงตรรกะ AND (&&) และ OR (||) (ต่อ)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
-> or ->>	กำหนดค่าทางขวา	c(3,1,TRUE,2+3i) -> v1 c(3,1,TRUE,2+3i) ->> v2 print(v1) print(v2) ผลลัพธ์ [1] 3+0i 1+0i 1+0i 2+3i [1] 3+0i 1+0i 1+0i 2+3i

2.3.5 ตัวดำเนินการอื่น ๆ (Miscellaneous Operators)

นอกเหนือจากตัวดำเนินการที่ได้กล่าวไปข้างต้น ยังมีตัวดำเนินการอื่น ๆ ที่ใช้สำหรับการเขียนโปรแกรมในภาษา R ซึ่งมีวัตถุประสงค์การใช้งานพิเศษ ไม่ได้ใช้กับการคำนวณทางคณิตศาสตร์หรือตรรกะทั่วไป ดังแสดงรายละเอียดในตารางที่ 2.6

ตารางที่ 2.6 ตัวดำเนินการอื่น ๆ ในภาษา R

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
:	ใช้เพื่อกำหนดลำดับชุดข้อมูล	v <- 2:8 print(v) ผลลัพธ์ [1] 2 3 4 5 6 7 8

ตารางที่ 2.6 ตัวดำเนินการอื่น ๆ ในภาษา R (ต่อ)

ตัวดำเนินการ	คำอธิบาย	ตัวอย่างการทำงาน
%in%	เปรียบเทียบค่าในเวกเตอร์	v1 <- 8 v2 <- 12 t <- 1:10 print(v1 %in% t) print(v2 %in% t) ผลลัพธ์ [1] TRUE [1] FALSE
%*%	คูณเมทริกซ์	M = matrix(c(2,6,5,1,10,4), nrow = 2,ncol = 3,byrow = TRUE) t = M %*% t(M) print(t) ผลลัพธ์ [,1] [,2] [1,] 65 82 [2,] 82 117

2.4 สรุป

ชนิดข้อมูลในภาษา R นั้น ค่อนข้างแตกต่างจากภาษาอื่น โดยสามารถแบ่งเป็น 2 ประเภท คือ ข้อมูลเดี่ยว และข้อมูลชุด ซึ่งข้อมูลเดี่ยวจะเหมือนกับการเขียนโปรแกรมภาษาอื่นทั่วไปที่ประกอบด้วยข้อมูลตัวเลข ตัวอักษระ ตรรกะ เป็นต้น แต่ส่วนที่แตกต่างจากภาษาอื่นคือ ข้อมูลชุดที่ประกอบด้วยข้อมูลเวกเตอร์ ลิสต์ ดาต้าเฟรม เมตริกซ์ และแฟคเตอร์ ซึ่งทำให้การเขียนโปรแกรมภาษา R มีความยืดหยุ่นเหมาะสมสำหรับการวิเคราะห์ข้อมูลทางสถิติเป็นอย่างมาก

สำหรับรูปแบบการใช้งานตัวแปรและตัวดำเนินการของโปรแกรมภาษา R นั้น ไม่แตกต่างจากภาษาอื่นมากนัก สามารถทำได้ง่ายไม่ยุ่งยาก นอกจากนี้ภาษา R ยังได้เตรียมตัวดำเนินการเพิ่มเติมอื่น ๆ นอกเหนือจากตัวดำเนินการหลักซึ่งทำให้เกิดความคล่องตัวในการเขียนโปรแกรมอย่างมาก

2.5 แบบฝึกหัดท้ายบท

2.5.1 จงอธิบายความหมายของประเภทข้อมูลเดี่ยว พร้อมทั้งยกตัวอย่างมาให้เข้าใจ

2.5.2 จงอธิบายความหมายของประเภทข้อมูลชุด พร้อมทั้งยกตัวอย่างมาให้เข้าใจ

2.5.3 จงอธิบายความหมายของคำสงวน พร้อมยกตัวอย่างมาอย่างน้อย 5 คำ

2.5.4 จงอธิบายความหมายของตัวแปร พร้อมกฎการตั้งชื่อ และการประกาศตัวแปรมาให้เข้าใจ

2.5.5 จงอธิบายความหมายและประเภทของตัวดำเนินการในภาษา R มาให้เข้าใจ พร้อมยกตัวอย่างประกอบ