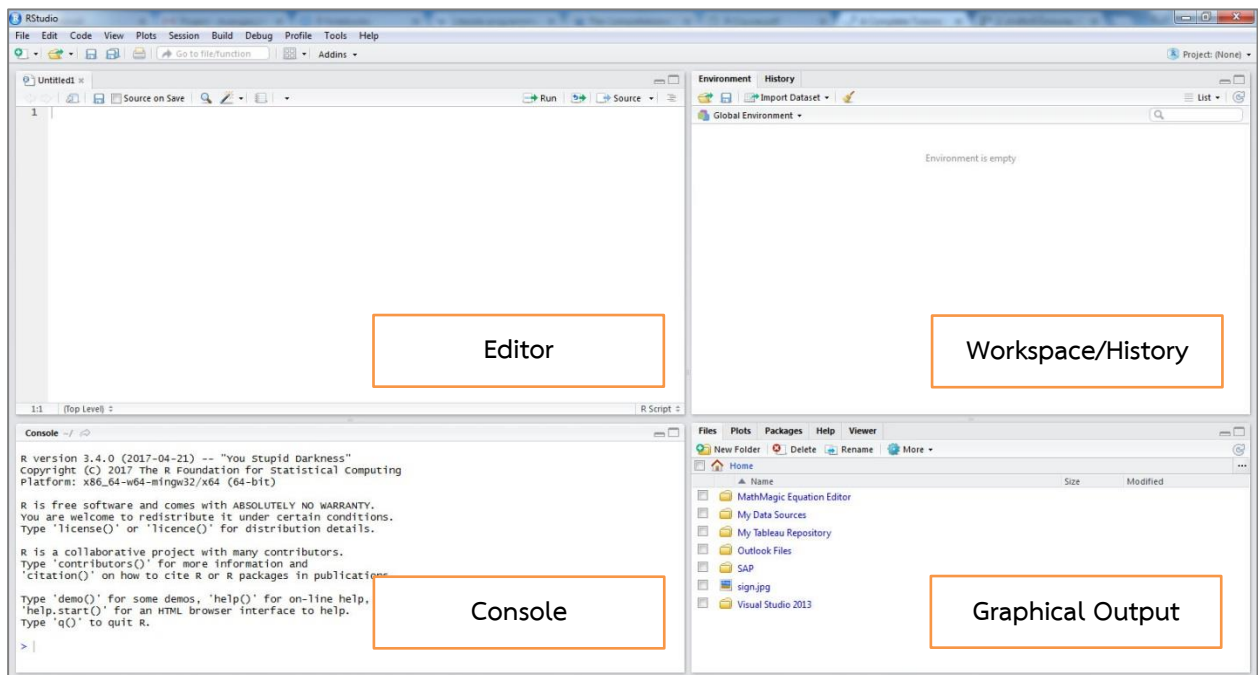


Programming with R for Beginners: Part I

1. โปรแกรม R Studio



2. เริ่มต้นทำการทดสอบการทำงานด้วยการตรวจสอบเวอร์ชันของ R โดยใช้คำสั่ง version (พิมพ์คำสั่งในหน้าต่าง Console)

```
> version

platform      x86_64-w64-mingw32
arch           x86_64
os             mingw32
system         x86_64, mingw32
status
major          3
minor          4.0
year           2017
month          04
day            21
svn rev        72570
language       R
version.string  R version 3.4.0 (2017-04-21)
nickname       You Stupid Darkness
```

3. หากต้องการเคลียร์หน้าต่าง Console ใช้คำสั่ง Ctrl+L

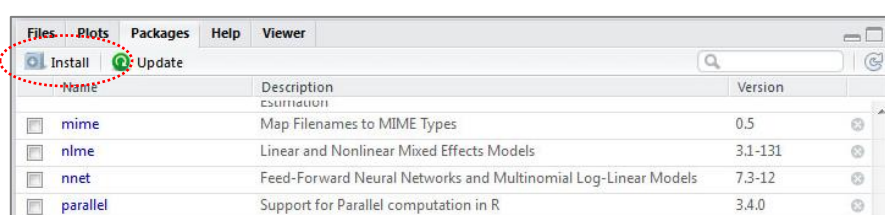
4. การใช้งานแพ็คเกจ (Package)

วิธีที่ 1

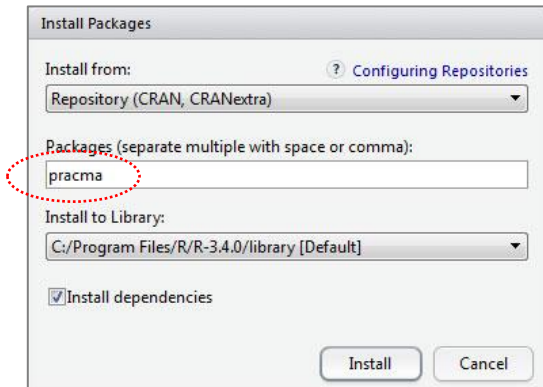
- ติดตั้งแพ็คเกจที่ต้องการโดยใช้คำสั่ง `install.packages("ชื่อแพ็คเกจ")`
- โหลดแพ็คเกจขึ้นมาใช้งานโดยใช้คำสั่ง `library("ชื่อแพ็คเกจ")`

วิธีที่ 2

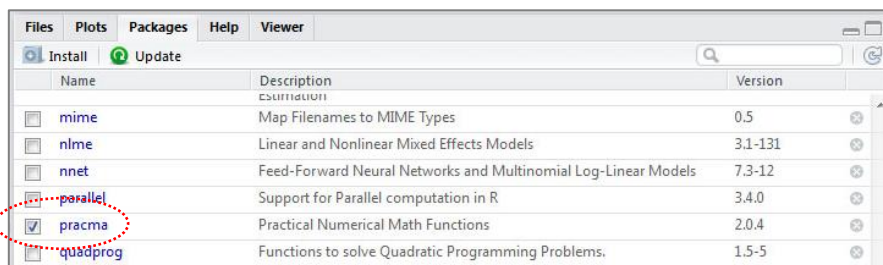
- ที่หน้าต่าง Graphical output คลิกเลือกแท็บ (Tab) Packages แล้วคลิกปุ่ม Install



- ที่ช่อง Packages ใส่ชื่อแพ็คเกจที่ต้องการติดตั้ง ในที่นี้ใส่เป็นแพ็คเกจที่ชื่อว่า pracma (Practical Numerical Math Functions) แล้วกดปุ่ม Install



- เมื่อต้องการโหลดแพ็คเกจขึ้นมาใช้งาน ให้ไปที่หน้าต่าง Graphical output คลิกเลือกแท็บ (Tab) Packages แล้วคลิกเลือกแพ็คเกจที่ต้องการโหลด

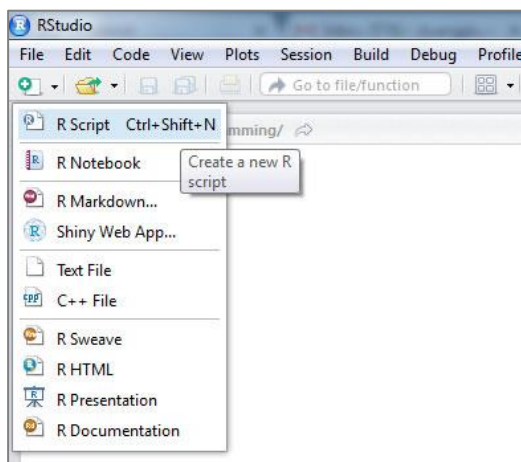


- ทดสอบการทำงานของแพ็คเกจ pracma

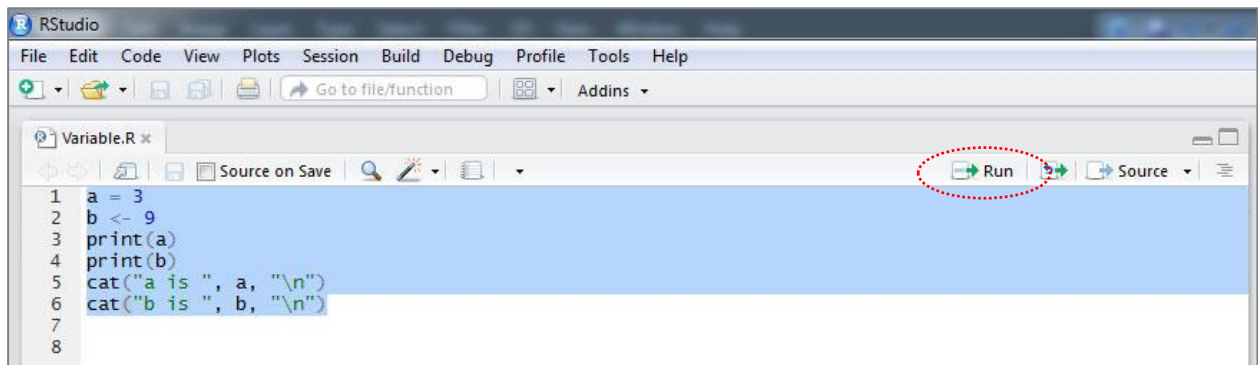
```
> idivide(7, 3)
> mod(7, 3)
```

* สามารถดูรายชื่อแพ็คเกจทั้งหมดได้ที่ https://cran.r-project.org/web/packages/available_packages_by_name.html

- การตั้งชื่อตัวแปร ชื่อตัวแปรประกอบไปด้วยตัวอักษร (Letters), ตัวเลข (Numbers), เครื่องหมายจุด (Dot) (.) และเครื่องหมายขีดล่าง (Underline) (_) โดยที่ตัวแปรสามารถขึ้นต้นด้วยตัวอักษรหรือเครื่องหมายจุดที่ไม่ตามด้วยตัวเลขได้
- การใช้งานหน้าต่าง R Script ไปที่ทูลบาร์ (Tool Bar) คลิกที่ไอคอนแรก (รูปกระดาษว่าง) แล้วเลือกเมนู R Script



7. ตัวอย่างคำสั่งในเรื่องของการกำหนดค่าตัวแปร



The screenshot shows the RStudio interface with a script editor containing the following R code:

```
1 a = 3
2 b <- 9
3 print(a)
4 print(b)
5 cat("a is ", a, "\n")
6 cat("b is ", b, "\n")
7
8
```

The 'Run' button in the top right corner of the script editor is circled in red.

- การกำหนดค่าให้กับตัวแปร ใช้เครื่องหมาย = หรือ <-
- คำสั่ง cat() ใช้ในการแสดงผลหลายๆ ค่า
- ถ้าต้องการเรียกดูตัวแปรทั้งหมด ใช้คำสั่ง print(ls())
- ถ้าต้องการลบตัวแปร ใช้คำสั่ง rm(ชื่อตัวแปร)
- ถ้าต้องการลบตัวแปรทั้งหมด ใช้คำสั่ง rm(list = ls())

8. ชนิดข้อมูล (Data Types)

(18.1) ชนิดข้อมูลแบบพื้นฐาน (Atomic)

(18.2) ชนิดข้อมูลแบบเวกเตอร์ (Vector) เป็นชุดข้อมูลประเภทเดียวกันมากกว่า 1 ตัวขึ้นไป

(18.3) ชนิดข้อมูลแบบแฟกเตอร์ (Factor) เป็นชุดข้อมูลสำหรับข้อมูลเชิงคุณภาพ (Categorical data) จะประกอบไปด้วย level หรือ category ของข้อมูลว่ามีทั้งหมดกี่ประเภท

(18.4) ชนิดข้อมูลแบบลิสต์ (List) เป็นชุดข้อมูลเวกเตอร์ชนิดพิเศษที่สามารถบรรจุข้อมูลประเภทใดก็ได้ โดยไม่จำเป็นต้องเป็นข้อมูลชนิดเดียวกัน

(18.5) ชนิดข้อมูลแบบเมตริกซ์ (Matrix) เป็นชุดข้อมูลประเภทเดียวกันที่เก็บไว้เป็น 2 มิติ คือเป็นแถว (row) และคอลัมน์ (column)

(18.6) ชนิดข้อมูลแบบอาร์เรย์ (Array) เป็นชุดข้อมูลเมตริกซ์ที่มีจำนวนมิติมากกว่า 2 มิติ

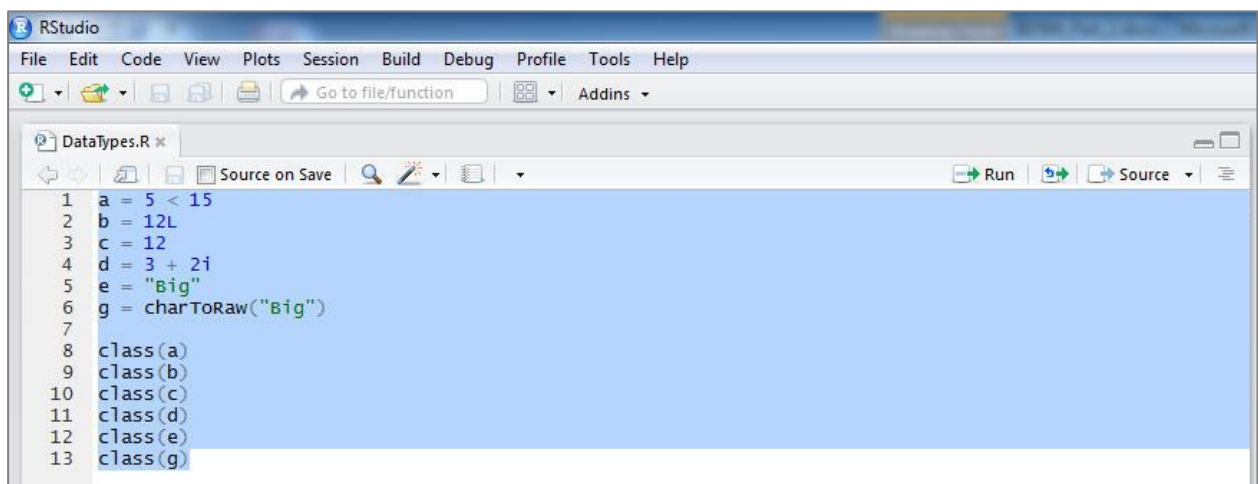
(18.7) ชนิดข้อมูลแบบดาต้าเฟรม (Data Frame) เป็นชุดข้อมูลเมตริกซ์ โดยแต่ละคอลัมน์จะเป็นตัวแปรซึ่งมีข้อมูลชนิดเดียวกัน และทุกๆ คอลัมน์จะมีจำนวนแถวเท่ากัน

(18.8) ชนิดข้อมูลที่สูญหาย (Missing values) แทนค่าได้ด้วยตัวแปรพิเศษ NA มาจากคำว่า Not Available ส่วนค่าที่ไม่นิยามเป็นตัวเลขจะแทนด้วย NaN มาจากคำว่า Not a Number

(18.1) ชนิดข้อมูลแบบพื้นฐาน (Atomic)

ชนิดข้อมูล	คำอธิบาย
Logical	ข้อมูลชนิดตรรกะ มีค่าเป็น True หรือ False
Integer	ข้อมูลตัวเลขชนิดจำนวนเต็ม
Numeric	ข้อมูลตัวเลขชนิดจำนวนจริง
Complex	ข้อมูลชนิดจำนวนเชิงซ้อน
Character, String	ข้อมูลชนิดตัวอักษร
Raw	ข้อมูลชนิด bitstream แสดงข้อมูลในรูปเลขฐาน 16

- R จะประมวลผลตัวเลขในลักษณะของจำนวนจริงเสมอ หากต้องการกำหนดให้ตัวเลขเป็นจำนวนเต็มจะต้องเติม L ไว้ด้านหลัง เช่น 2L
- Inf (Infinity) หมายถึงมีค่าอนันต์ แต่สามารถนำมาคำนวณได้
- NaN (Not a Number) หมายถึงหาค่าไม่ได้
- ตัวอย่างคำสั่ง



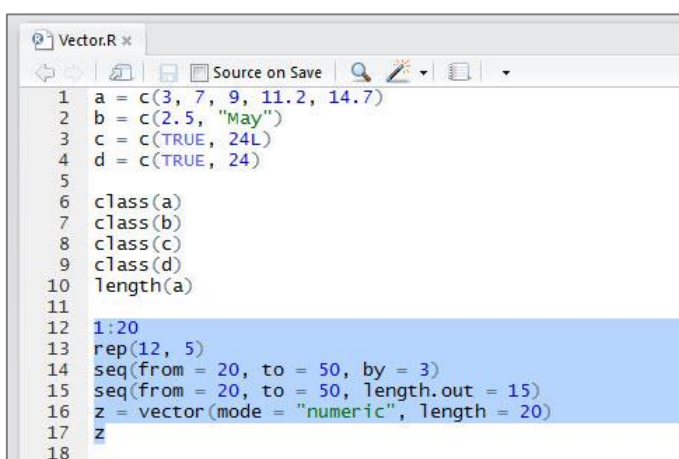
```

RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
DataTypes.R
Source on Save Run Source
1 a = 5 < 15
2 b = 12L
3 c = 12
4 d = 3 + 2i
5 e = "Big"
6 g = charToRaw("Big")
7
8 class(a)
9 class(b)
10 class(c)
11 class(d)
12 class(e)
13 class(g)

```

(18.2) ชนิดข้อมูลแบบเวกเตอร์ (Vector) เป็นชุดข้อมูลประเภทเดียวกันมากกว่า 1 ตัวขึ้นไป คำสั่งในการสร้างเวกเตอร์ คือ c() และคำสั่งในการกำหนดเวกเตอร์ว่างๆ ไว้เพื่อใช้งาน คือ vector()

- ตัวอย่างคำสั่ง



```

Vector.R
Source on Save
1 a = c(3, 7, 9, 11.2, 14.7)
2 b = c(2.5, "May")
3 c = c(TRUE, 24L)
4 d = c(TRUE, 24)
5
6 class(a)
7 class(b)
8 class(c)
9 class(d)
10 length(a)
11
12 1:20
13 rep(12, 5)
14 seq(from = 20, to = 50, by = 3)
15 seq(from = 20, to = 50, length.out = 15)
16 z = vector(mode = "numeric", length = 20)
17 z
18

```

(18.3) ชนิดข้อมูลแบบแฟกเตอร์ (Factor) เป็นชุดข้อมูลสำหรับข้อมูลเชิงคุณภาพ (Categorical data) จะประกอบไปด้วย level หรือ category ของข้อมูลว่ามีทั้งหมดกี่ประเภท

■ ตัวอย่างคำสั่ง

```
Factor.R *
1 x = factor(c("A", "B", "A", "A", "C"))
2 x
3
4 y = factor(c(T, TRUE, T, F, FALSE))
5 y
6
```

(18.4) ชนิดข้อมูลแบบลิสต์ (List) เป็นชุดข้อมูลเวกเตอร์ชนิดพิเศษที่สามารถบรรจุข้อมูลประเภทใดก็ได้ โดยไม่จำเป็นต้องเป็นข้อมูลชนิดเดียวกัน

■ ตัวอย่างคำสั่ง (คำสั่ง str() ใช้เรียกดูโครงสร้างข้อมูล)

```
List.R *
1 myList = list(
2   one = 1:5,
3   two = 1,
4   three = c(3, 4),
5   four = matrix(11:16, ncol = 3),
6   five = c("a", "b")
7 )
8
9 myList
10
11 str(myList)
12 summary(myList)
13
14 myList$one
15 myList[[1]]
```


(18.5) ชนิดข้อมูลแบบเมตริกซ์ (Matrix) เป็นชุดข้อมูลประเภทเดียวกันที่เก็บไว้เป็น 2 มิติ คือเป็นแถว (row) และคอลัมน์ (column) ในการสร้างเมตริกซ์จะต้องกำหนดจำนวนแถวด้วยคำสั่ง nrow และกำหนดจำนวนคอลัมน์ด้วยคำสั่ง ncol

■ ตัวอย่างคำสั่ง

```
Matrix.R *
1 data = c(1, 2, 3, 4, 5, 6)
2 A_mat = matrix(data, nrow = 2, byrow = T)
3
4 A_mat
5 A_mat[1,]
6 A_mat[, 2]
7 A_mat[2, 3]
8
9 diag(4)
10 matrix(0, nrow = 3, ncol = 4)
```

(18.6) ชนิดข้อมูลแบบอาร์เรย์ (Array) เป็นชุดข้อมูลเมตริกซ์ที่มีจำนวนมิติมากกว่า 2 มิติ โดยใช้คำสั่ง `dim()` ในการกำหนดจำนวนมิติ

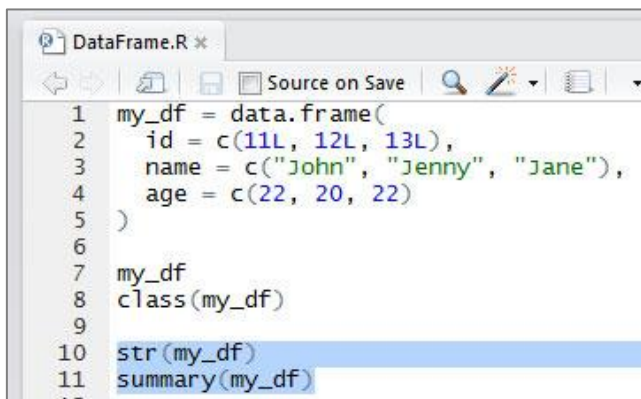
■ ตัวอย่างคำสั่ง



```
1 a = array(  
2   c("green", "yellow"),  
3   dim = c(3, 3, 2)  
4 )  
5  
6 a
```

(18.7) ชนิดข้อมูลแบบดาต้าเฟรม (Data Frame) เป็นชุดข้อมูลเมตริกซ์ โดยแต่ละคอลัมน์จะเป็นตัวแปรซึ่งมีข้อมูลชนิดเดียวกัน และทุกๆ คอลัมน์จะมีจำนวนแถวเท่ากัน

■ ตัวอย่างคำสั่ง



```
1 my_df = data.frame(  
2   id = c(11L, 12L, 13L),  
3   name = c("John", "Jenny", "Jane"),  
4   age = c(22, 20, 22)  
5 )  
6  
7 my_df  
8 class(my_df)  
9  
10 str(my_df)  
11 summary(my_df)
```

■ ตัวอย่างคำสั่ง



```
13 my_df$id  
14 my_df[["id"]]  
15 my_df[[1]]  
16  
17 my_df$name  
18 my_df[["name"]]  
19 my_df[, 2]  
20  
21 my_df[, 1:2]  
22 my_df[2:3, ]  
23 my_df[, c("id", "age")]  
24 my_df[c(1, 3), ]  
25  
26 my_df[my_df$age == 22, ]
```

(18.8) ชนิดข้อมูลที่สูญหาย (Missing values) แทนค่าได้ด้วยตัวแปรพิเศษ NA มาจากคำว่า Not Available ส่วนค่าที่ไม่นิยามเป็นตัวเลขจะแทนด้วย NaN มาจากคำว่า Not a Number ซึ่ง NaN ถือเป็นรูปแบบหนึ่งของ NA

■ ตัวอย่างคำสั่ง

```
1 df = data.frame(
2   name = c("John", "Jane", "Paul", "Mark"),
3   score = c(NA, NA, 87, 91)
4 )
5
6 df
7 is.na(df)
```

■ ตัวอย่างคำสั่ง

```
9 df[!complete.cases(df), ]  <----- เรียกดูแถวข้อมูลที่มีข้อมูลสูญหาย
10 mean(df$score)
11 mean(df$score, na.rm = T)  <----- หาค่าเฉลี่ยโดยไม่นับคอลัมน์ที่มีค่าข้อมูลสูญหาย
12
13 new_df = na.omit(df)       <----- ลบแถวข้อมูลที่มีข้อมูลสูญหาย
15 new_df
```

แบบฝึกหัด

1. จงสร้างตัวแปรเพื่อเก็บข้อมูลดังต่อไปนี้

name	gender	height	weight	age	bmi
Paul	Male	152	81	42	0
John	Male	171.5	50	38	0
Lisa	Female	165	59	26	0

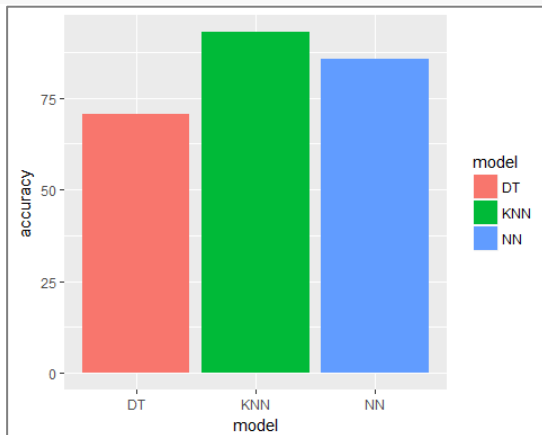
2. จงเขียนคำสั่งเพื่อเรียกดูโครงสร้างของตัวแปรที่สร้างขึ้นในข้อที่ 1.
3. จงเขียนคำสั่งเพื่อเรียกดูแถวข้อมูล ที่มีค่าส่วนสูงมากกว่า 160 ซม.
4. จงเขียนคำสั่งเพื่อหาค่าเฉลี่ยของอายุ
5. จงเขียนคำสั่งเพื่อคำนวณหาค่าดัชนีมวลกาย (bmi) โดยที่ ค่าดัชนีมวลกาย (bmi) = น้ำหนัก (กิโลกรัม) / ส่วนสูง (เมตร)² และเก็บค่าดัชนีมวลกายที่คำนวณได้ลงในตัวแปร bmi
6. จงเขียนคำสั่งเพื่อเรียกดูชื่อ (name) คนที่มีค่าดัชนีมวลกาย (bmi) มากกว่า 25

Programming with R for Beginners: Part II

1. การนำเสนอข้อมูล (Visualization with R)

- (1.1) ติดตั้งแพ็คเกจสำหรับการแสดงผลด้วยกราฟ แพ็คเกจชื่อ ggplot2 (โดยใช้คำสั่ง `install.packages("ชื่อแพ็คเกจ")` หรือใช้เครื่องมือของ R Studio)
- (1.2) ทำการโหลดแพ็คเกจขึ้นมาใช้งาน (โดยใช้คำสั่ง `library("ชื่อแพ็คเกจ")` หรือใช้เครื่องมือของ R Studio)
- (1.3) เพิ่มพื้นที่ (Chunk) ในการเขียนสคริปต์ใหม่ และเขียนคำสั่งดังต่อไปนี้ เสร็จแล้วกดปุ่มบันทึก (Save) (ไอคอนรูปแผ่นดิสก์) (วิเคราะห์ผลลัพธ์ที่เกิดขึ้น) แล้วกดปุ่มรัน

```
dat = data.frame(
  model = c("DT", "KNN", "NN"),
  accuracy = c(70.5, 93.0, 85.6)
)
dat
ggplot(data=dat, aes(x=model, y=accuracy, fill=model)) + geom_bar(stat="identity")
```



2. การนำเสนอข้อมูล (Visualization with R) (ต่อ)

- (2.1) ติดตั้งแพ็คเกจสำหรับการแปลงโครงสร้างของข้อมูล แพ็คเกจชื่อ reshape2 (โดยใช้คำสั่ง `install.packages("ชื่อแพ็คเกจ")` หรือใช้เครื่องมือของ R Studio)
- (2.2) ทำการโหลดแพ็คเกจขึ้นมาใช้งาน (โดยใช้คำสั่ง `library("ชื่อแพ็คเกจ")` หรือใช้เครื่องมือของ R Studio)
- (2.3) เพิ่มพื้นที่ (Chunk) ในการเขียนสคริปต์ใหม่ และเขียนคำสั่งดังต่อไปนี้ เสร็จแล้วกดปุ่มบันทึก (Save) (ไอคอนรูปแผ่นดิสก์) (วิเคราะห์ผลลัพธ์ที่เกิดขึ้น) แล้วกดปุ่มรัน

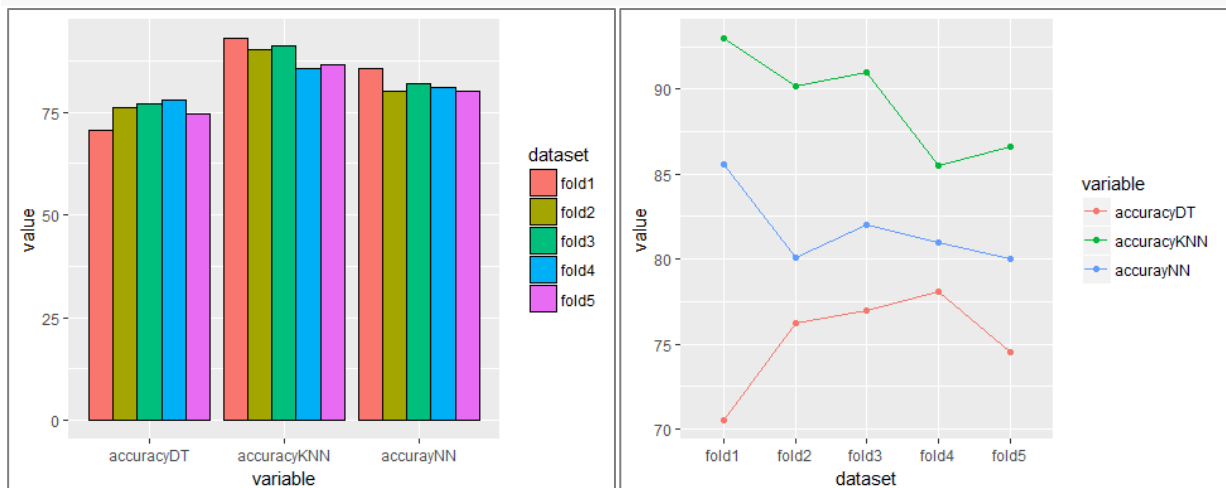
```
dat2 = data.frame(
  dataset = c("fold1", "fold2", "fold3", "fold4", "fold5"),
  accuracyDT = c(70.5, 76.2, 77.0, 78.1, 74.5),
  accuracyKNN = c(93.0, 90.2, 91.0, 85.5, 86.6),
  accuracyNN = c(85.6, 80.1, 82.0, 81.0, 80.0)
)
dat2
dat2 = melt(dat2, id.vars="dataset") ← ฟังก์ชันจากแพ็คเกจ reshape2
```



```
bar = ggplot(data=dat2, aes(x=variable, y=value, fill=dataset)) +  
  geom_bar(stat="identity", position=position_dodge(), colour="black")  
line = ggplot(data=dat2, aes(x=dataset, y=value, group=variable, colour=variable)) +  
  geom_line() + geom_point()
```

bar

line



Programming with R for Beginners: Part III

1. การวิเคราะห์ข้อมูลด้วยกระบวนการวิธี Linear Regression

(1.1) เขียนคำสั่งเพื่ออ่านไฟล์ข้อมูล

```
data.train = read.csv("heating_oil_train.csv")
```

```
data.train
```

ฟังก์ชัน read.csv() ทำการอ่านไฟล์ นามสกุล .csv

(1.2) โครงสร้างข้อมูลของ heating_oil_train.csv

Attribute	Type	Description
Insulation	Integer	ฉนวนกันความร้อนภายในบ้าน
Temperature	Integer	อุณหภูมิภายนอกบ้านในปีที่ผ่านมา
Heating_Oil	Integer	ปริมาณการสั่งซื้อน้ำมันเพื่อทำความร้อนเมื่อปีที่ผ่านมา
Num_Occupants	Integer	จำนวนผู้อยู่อาศัยภายในบ้าน
Avg_Age	Double (Real)	อายุเฉลี่ยของผู้อยู่อาศัยในบ้าน
Home_Size	Integer	ขนาดของบ้าน (ตัวเลขมากแสดงว่าขนาดใหญ่)

Insulation <int>	Temperature <int>	Heating_Oil <int>	Num_Occupants <int>	Avg_Age <dbl>	Home_Size <int>
6	74	132	4	23.8	4
10	43	263	4	56.7	4
3	81	145	2	28.0	6
9	50	196	4	45.1	3
2	80	131	5	20.8	2
5	76	129	3	21.5	3
5	72	131	4	23.5	3
6	88	161	2	38.2	6
5	77	184	3	42.5	3
10	42	225	3	51.1	1

1-10 of 1,218 rows Previous 1 2 3 4 5 6 ... 100 Next

(1.3) เขียนคำสั่งเรียกใช้งานฟังก์ชัน lm() เพื่อทำการสร้างโมเดลด้วยกระบวนการวิธี Linear Regression (วิธีพิมพ์เครื่องหมาย Tilde หรือ Accent (~) ใช้คำสั่ง Alt+126)

```
model = lm(Heating_Oil ~ ., data=data.train)
```

```
model
```

```
Call:
lm(formula = Heating_Oil ~ ., data = data.train)

Coefficients:
(Intercept)      Insulation  Temperature  Num_Occupants      Avg_Age
      135.4809         3.3255        -0.8699        -0.2690         1.9658
      Home_Size
       3.1715
```

(1.4) เขียนคำสั่งเรียกใช้งานฟังก์ชัน read.csv() เพื่อทำการอ่านไฟล์สำหรับพยากรณ์ข้อมูล (ทดสอบโมเดล)

```
data.test = read.csv("heating_oil_test.csv")
```

```
data.test
```

Insulation <int>	Temperature <int>	Num_Occupants <int>	Avg_Age <dbl>	Home_Size <int>
5	69	10	70.1	7
5	80	1	66.7	1
4	89	9	67.8	7
7	81	9	52.4	6
4	58	8	22.9	7
4	58	6	37.4	3
6	51	2	51.6	3
2	73	5	37.4	4
9	39	1	56.9	7
8	84	5	64.5	2

1-10 of 42,650 rows

Previous 1 2 3 4 5 6 ... 100 Next

ข้อสังเกต ข้อมูลสำหรับพยากรณ์ (ทดสอบโมเดล) จะไม่มีแอททริบิวต์ Heating_Oil

(1.5) เขียนคำสั่งเรียกใช้งานฟังก์ชัน predict() เพื่อทำการพยากรณ์ข้อมูล (การนำโมเดลมาใช้งานเพื่อพยากรณ์ค่าของข้อมูล)

```
p = predict(model, data.test)
```

```
data.test$predict = p
```

```
data.test
```

Insulation <int>	Temperature <int>	Num_Occupants <int>	Avg_Age <dbl>	Home_Size <int>	predict <dbl>
5	69	10	70.1	7	249.3982
5	80	1	66.7	1	216.5368
4	89	9	67.8	7	224.4218
7	81	9	52.4	6	207.9126
4	58	8	22.9	7	163.3933
4	58	6	37.4	3	179.7495
6	51	2	51.6	3	221.4805
2	73	5	37.4	4	163.4900
9	39	1	56.9	7	265.2700
8	84	5	64.5	2	220.8045

1-10 of 42,650 rows

Previous 1 2 3 4 5 6 ... 100 Next

(1.6) เขียนคำสั่งเรียกใช้งานฟังก์ชัน write.csv() เพื่อทำการบันทึกผลลัพธ์ที่ได้ออกเป็นไฟล์ .csv

```
write.csv(data.test, file="predict_heating.csv")
```

แบบฝึกหัดที่ 1

- (1) เป้าหมายของการวิเคราะห์คือประเมิน (พยากรณ์) ค่าใช้จ่ายในการรักษาพยาบาล (expenses) โดยใช้โมเดล Linear Regression
- (2) ไฟล์ข้อมูลสำหรับนำไปสร้างโมเดล คือ insurance_train.csv

age	sex	bmi	children	smoker	region	expenses
<dbl>	<fctr>	<dbl>	<dbl>	<fctr>	<fctr>	<dbl>
19	female	27.9	0	yes	southwest	16884.92
18	male	33.8	1	no	southeast	1725.55
28	male	33.0	3	no	southeast	4449.46
33	male	22.7	0	no	northwest	21984.47
32	male	28.9	0	no	northwest	3866.86
31	female	25.7	0	no	southeast	3756.62
37	male	29.8	2	no	northeast	6406.41
23	male	34.4	0	no	southwest	1826.84
56	female	39.8	0	no	southeast	11090.72
27	male	42.1	0	yes	southeast	39611.76
1-10 of 669 rows						
Previous 1 2 3 4 5 6 ... 67 Next						

- (3) ไฟล์ข้อมูลสำหรับพยากรณ์ (ทดสอบโมเดล) คือ insurance_test.csv
- (4) บันทึกผลลัพธ์ที่ได้จากการพยากรณ์เป็นไฟล์ กำหนดให้ชื่อว่า predict_insurance.csv

2. การวิเคราะห์ข้อมูลด้วยกระบวนการวิธี Decision Tree

- (2.1) ติดตั้งแพ็คเกจสำหรับใช้งานฟังก์ชันทางด้าน Decision Tree แพ็คเกจชื่อ rpart (โดยใช้คำสั่ง install.packages("ชื่อแพ็คเกจ") หรือใช้เครื่องมือของ R Studio)
- (2.2) ทำการโหลดแพ็คเกจขึ้นมาใช้งาน (โดยใช้คำสั่ง library("ชื่อแพ็คเกจ") หรือใช้เครื่องมือของ R Studio)
- (2.3) เขียนคำสั่งเรียกใช้งานฟังก์ชัน read.csv() เพื่อทำการอ่านไฟล์ข้อมูลสำหรับนำไปสร้างโมเดล และไฟล์ข้อมูลสำหรับพยากรณ์ข้อมูล (ทดสอบโมเดล)

```
kyp.train = read.csv("kyphosis_train.csv")
```

```
kyp.test = read.csv("kyphosis_test.csv")
```

- (2.4) โครงสร้างข้อมูลของ kyphosis_train.csv ข้อมูลเกี่ยวกับเด็กที่เคยผ่าตัดแก้ไขกระดูกสันหลัง

Attribute	Type	Description
Kyphosis	Factor	มีการเกิดการเสียรูปหลังการผ่าตัดหรือไม่ (absent/present)
Age	Integer	อายุ (เดือน)
Number	Integer	จำนวนกระดูกสันหลังที่เกี่ยวข้อง
Start	Integer	จำนวนกระดูกส่วนแรก (ส่วนบน) ที่ถูกดำเนินการ

X	Kyphosis	Age	Number	Start
<int>	<fctr>	<int>	<int>	<int>
1	absent	71	3	5
2	absent	158	3	14
3	present	128	4	5
4	absent	2	5	1
5	absent	1	4	15
6	absent	1	2	16
7	absent	61	2	17
8	absent	37	3	16
9	absent	113	2	16
10	present	59	6	12

1-10 of 57 rows

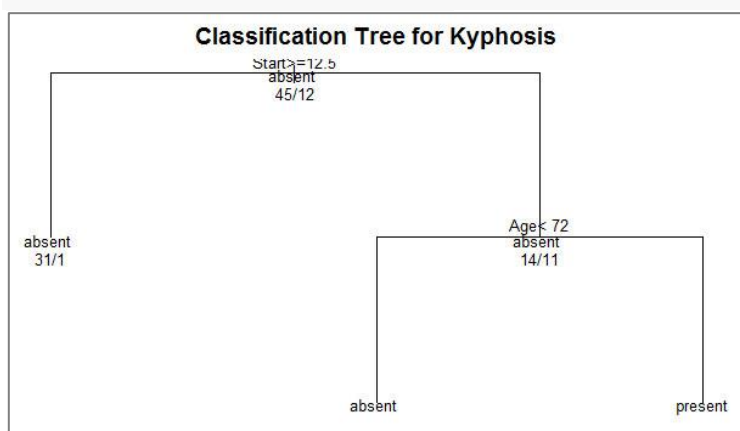
Previous 1 2 3 4 5 6 Next

(2.5) เขียนคำสั่งเรียกใช้งานฟังก์ชัน `rpart()` เพื่อทำการสร้างโมเดลด้วยกระบวนการวิธี Decision Tree โดยพิจารณาเลือกแอททริบิวต์ที่เหมาะสม (แอททริบิวต์ที่เป็นลำดับ หรือ ID ไม่นำมาพิจารณาในการสร้างโมเดล)

```
fit = rpart(Kyphosis ~ Age + Number + Start, method="class", data=kyp.train)
summary(fit)
```

(2.6) เขียนคำสั่งเรียกใช้งานฟังก์ชัน `plot()` และ `text()` เพื่อสร้างกราฟ Decision Tree

```
plot(fit, uniform=TRUE, main="Classification Tree for Kyphosis")
text(fit, use.n=TRUE, all=TRUE, cex=0.8)
```



(2.7) เขียนคำสั่งเรียกใช้งานฟังก์ชัน `predict()` เพื่อทำการพยากรณ์ข้อมูล (การนำโมเดลมาใช้งานเพื่อพยากรณ์ค่าของข้อมูล) แล้วบันทึกค่าที่พยากรณ์หรือทำนายได้ลงในชุดข้อมูล

```
p = predict(fit, kyp.test, type="class")
kyp.test$predict = p
kyp.test
```

X	Age	Number	Start	predict
<int>	<int>	<int>	<int>	<fctr>
58	120	5	8	present
59	51	7	9	absent
60	102	3	13	absent
61	130	4	1	present
62	114	7	8	present
63	81	4	1	present
64	118	3	16	absent
65	118	4	16	absent
66	17	4	10	absent
67	195	2	17	absent

1-10 of 24 rows

Previous 1 2 3 Next

แบบฝึกหัดที่ 2

- (1) เป้าหมายของการวิเคราะห์คือทำนายสาขาวิชาให้ตรงกับความสามารถของนักศึกษา (Major) [computer, hotel, marketing, management] โดยใช้โมเดล Decision Tree
- (2) ไฟล์ข้อมูลสำหรับนำไปสร้างโมเดล คือ student_train.csv

Student_ID	Study	Rank_study_group	Age_group	GPA_old_group	Rank_grade_major
<int>	<fctr>	<fctr>	<fctr>	<fctr>	<fctr>
1	BANGKOK	VOCATIONAL_EDU	<=20	Normal	Good
2	BANGKOK	VOCATIONAL_EDU	26-30	Normal	Good
3	UPCOUNTRY	VOCATIONAL_EDU	21-25	Good	Good
4	UPCOUNTRY	GENERAL_EDU	<=20	Bad	Bad
5	UPCOUNTRY	VOCATIONAL_EDU	21-25	Bad	Good
6	NA	GENERAL_EDU	26-30	Bad	Bad
7	NA	GENERAL_EDU	26-30	Good	Good
8	UPCOUNTRY	VOCATIONAL_EDU	21-25	Bad	Good
9	NA	GENERAL_EDU	26-30	Normal	Bad
10	NA	VOCATIONAL_EDU	<=20	Normal	Good

1-10 of 1,000 rows | 1-6 of 11 columns

Previous 1 2 3 4 5 6 ... 100 Next

จากชุดข้อมูลจะเห็นว่ามีแอททริบิวต์ Student_ID ซึ่งเป็นแอททริบิวต์ที่เป็นลำดับ และไม่นำมาพิจารณาในการสร้างโมเดล จึงต้องใช้ฟังก์ชัน subset() ในการเลือกแอททริบิวต์อีกครั้ง

```
stu.train = read.csv("student_train.csv")
stu.train = subset(stu.train, select = -c(Student_ID))
```

- (3) แสดงกราฟ Decision Tree ของโมเดลที่สร้างขึ้น
- (4) ไฟล์ข้อมูลสำหรับพยากรณ์ (ทดสอบโมเดล) คือ student_test.csv
- (5) บันทึกค่าที่พยากรณ์หรือทำนายได้ลงในชุดข้อมูล