

การใช้โปรแกรม R (R- Base Language)

โปรแกรม R เป็นระบบที่ใช้สำหรับการคำนวณทางสถิติและกราฟฟิก โดยใช้ภาษา R ซึ่งได้รับการพัฒนามาเพื่อทดแทนภาษา S ตัวภาษาจะเป็นแบบ FPL(functional programming language) และเป็นแบบ case sensitivity เราสามารถเขียนฟังก์ชันผ่านเข้าไปตอนอินพุตได้ ซึ่งจะทำให้สามารถนำไปใช้ประโยชน์สำหรับการสร้างโมเดลสถิติและโมเดลกราฟฟิกได้ ทั้งนี้ด้วยการที่โปรแกรมนี้เป็นโปรแกรมประเภท freeware อีกด้วยการติดตั้งโปรแกรม

การติดตั้งแบ่งออกเป็น 2 ระบบปฏิบัติการที่จะกล่าวในรายงานคือแบบบน Window และบน Ubuntu ได้ดังนี้

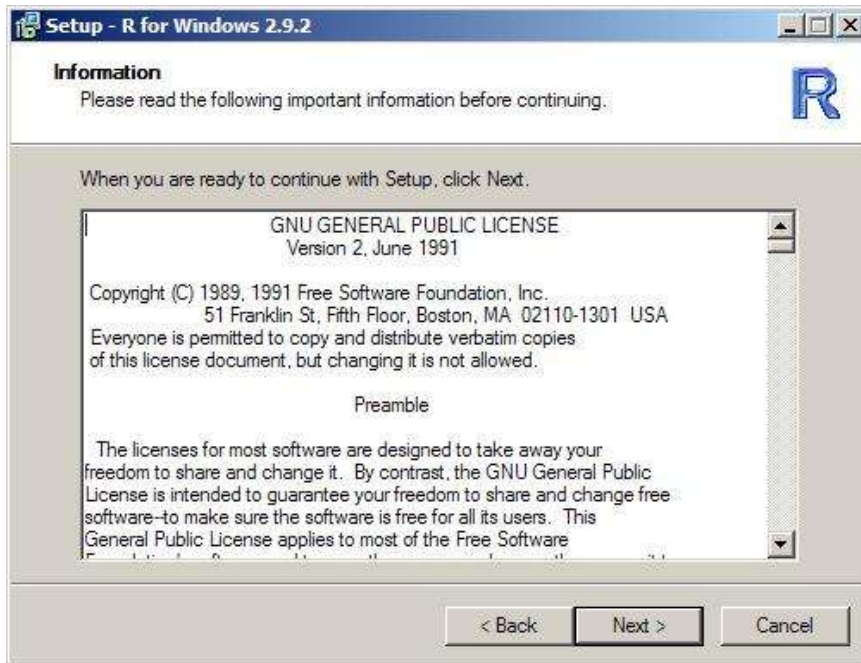
ในบน Window การติดตั้งต้องไปโหลดตัวติดตั้งจากเว็บไซต์ <http://mirror.kapook.com/cran/> หลังจากที่โหลดมาแล้วก็ทำการติดตั้งซึ่งสามารถแจ้งเป็นรายละเอียดการติดตั้งได้ดังนี้

ขั้นตอนการติดตั้ง R บน Window

1. ทำการโหลดไฟล์ R-2.9.2-win32.exe มายังเครื่องที่ต้องการติดตั้ง



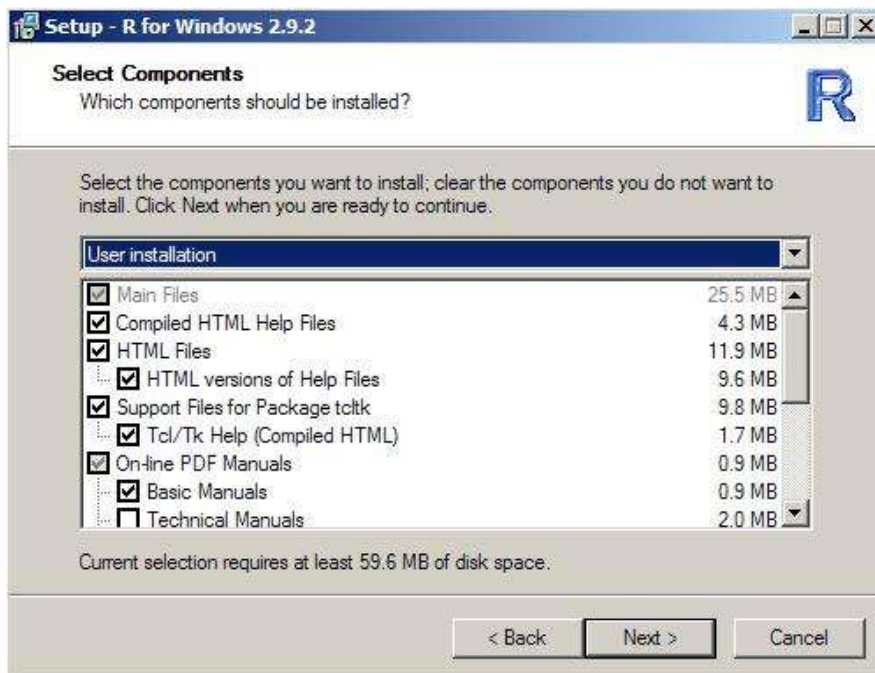
2. เมื่อรันตัวติดตั้ง แล้วจะเข้าสู่หน้านี้ ให้กด Next



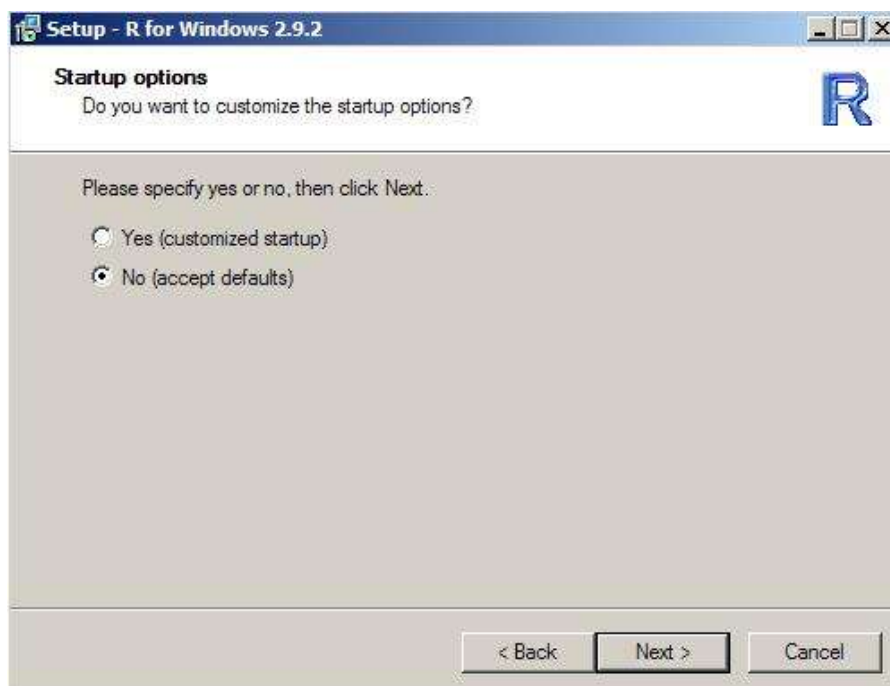
3. หน้านี้จะเป็นการบอกรายละเอียดของโปรแกรม เช่น ชื่อโปรแกรม, ชื่อรุ่น, ชื่อผู้สร้าง, และ ลิขสิทธิ์ ถ้ายอมรับให้กด Next



4. หน้านี้จะป็นตำแหน่งที่ต้องการติดตั้ง ในเครื่องของเรา และบอกขนาดพื้นที่ ที่ต้องการ ให้กด Next



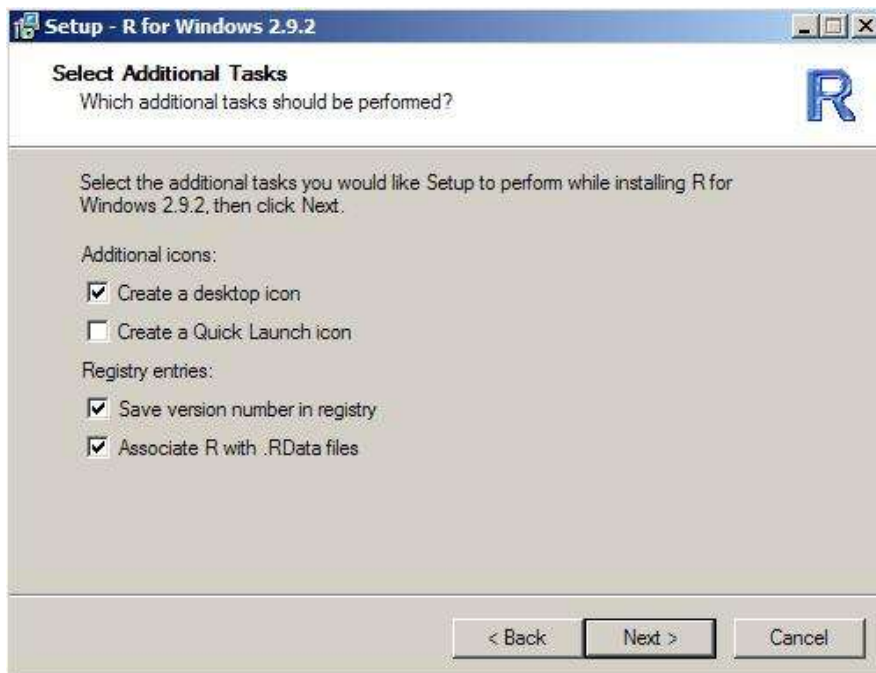
5. หน้านี้จะเป็นการเลือกรายละเอียดของโปรแกรม ที่ต้องการติดตั้ง ให้กด Next



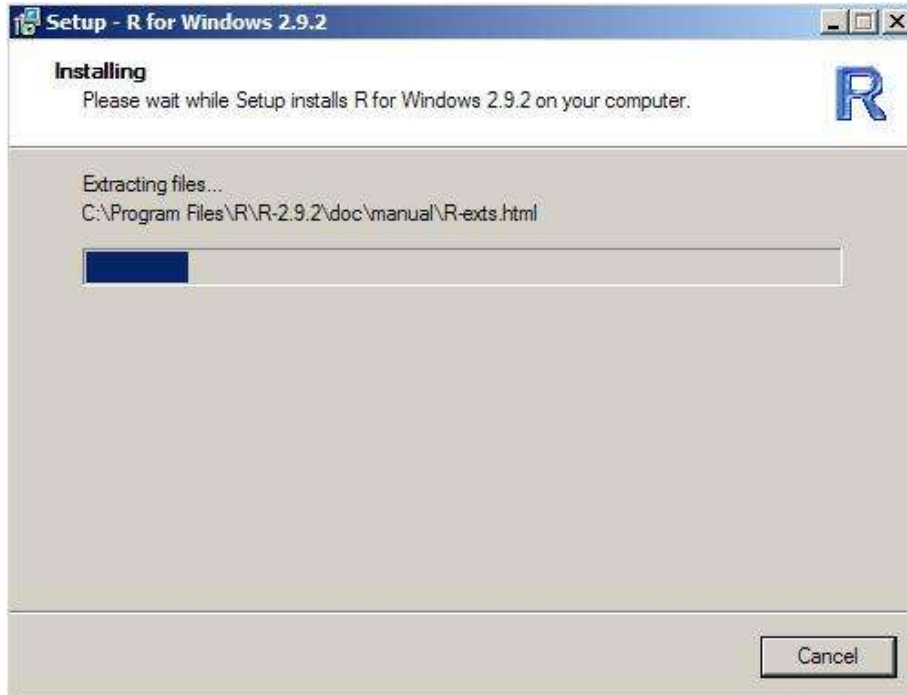
6. เป็นการเลือก option ถ้าต้องการให้ลงแบบปกติ ก็กด Next



7. เป็นการสร้างโฟลเดอร์ใน Start Menu ถ้าไม่ต้องการให้กดในช่อง Don't create a Start Menu folder หลังจากนั้นให้กด Next



8. เป็นการเลือกสร้าง Short Cut และตั้งค่า Registry จากนั้นกด Next



9. เป็นสถานะแสดงว่าเรากำลังแตกไฟล์ + กำลังติดตั้ง



10. หน้าต่างเป็นการบอกว่าได้ติดตั้งเสร็จสมบูรณ์แล้ว ให้กด Finish เพื่อออกจากการติดตั้ง

การติดตั้ง R บน Ubuntu

1. ทำการเรียกหน้าต่าง Terminal
2. ทำการอัปเดต โดยใช้คำสั่ง

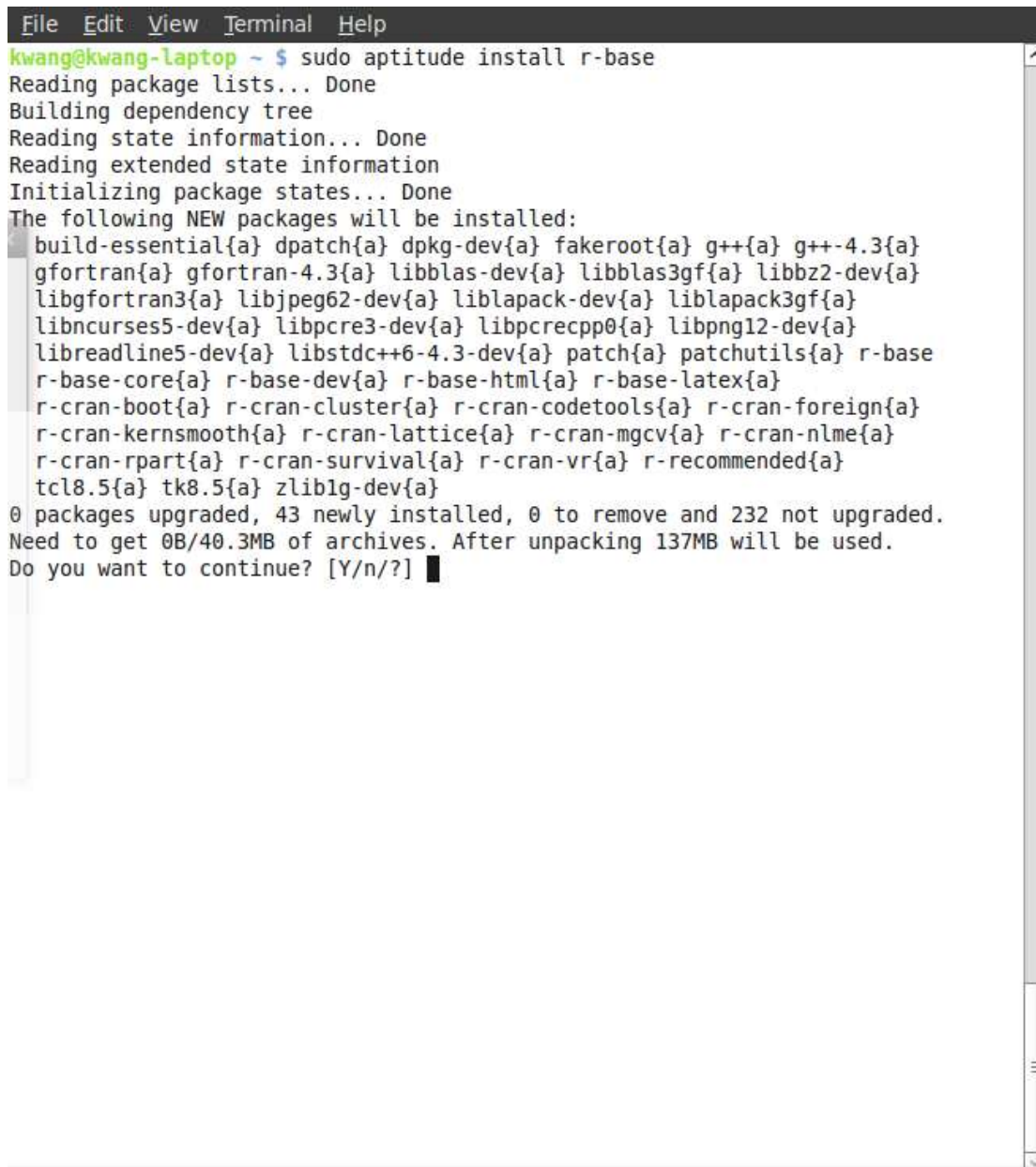
```
sudo apt-get update
```



```
kwang@kwang-Laptop ~ $ sudo aptitude update
Hit http://archive.canonical.com jaunty Release.gpg
Get:1 http://packages.linuxmint.com gloria Release.gpg [198B]
Hit http://security.ubuntu.com jaunty-security Release.gpg
Hit http://archive.ubuntu.com jaunty Release.gpg
Ign http://archive.canonical.com jaunty/partner Translation-en_US
Ign http://security.ubuntu.com jaunty-security/main Translation-en_US
Ign http://archive.ubuntu.com jaunty/main Translation-en_US
Ign http://packages.linuxmint.com gloria/main Translation-en_US
Hit http://archive.canonical.com jaunty Release
Ign http://security.ubuntu.com jaunty-security/restricted Translation-en_US
Ign http://archive.ubuntu.com jaunty/restricted Translation-en_US
Hit http://archive.canonical.com jaunty/partner Packages
Ign http://packages.linuxmint.com gloria/upstream Translation-en_US
Ign http://security.ubuntu.com jaunty-security/universe Translation-en_US
Ign http://archive.ubuntu.com jaunty/universe Translation-en_US
Ign http://security.ubuntu.com jaunty-security/multiverse Translation-en_US
Ign http://archive.ubuntu.com jaunty/multiverse Translation-en_US
Ign http://packages.linuxmint.com gloria/import Translation-en_US
Hit http://security.ubuntu.com jaunty-security Release
Hit http://archive.ubuntu.com jaunty-updates Release.gpg
Get:2 http://packages.linuxmint.com gloria Release [9189B]
Ign http://archive.ubuntu.com jaunty-updates/main Translation-en_US
Hit http://security.ubuntu.com jaunty-security/main Packages
Hit http://packages.medibuntu.org jaunty Release.gpg
Ign http://archive.ubuntu.com jaunty-updates/restricted Translation-en_US
Hit http://security.ubuntu.com jaunty-security/restricted Packages
Ign http://archive.ubuntu.com jaunty-updates/universe Translation-en_US
Hit http://security.ubuntu.com jaunty-security/universe Packages
Ign http://packages.medibuntu.org jaunty/free Translation-en_US
Ign http://packages.linuxmint.com gloria/main Packages
Ign http://archive.ubuntu.com jaunty-updates/multiverse Translation-en_US
Hit http://security.ubuntu.com jaunty-security/multiverse Packages
Ign http://packages.linuxmint.com gloria/upstream Packages
Ign http://packages.medibuntu.org jaunty/non-free Translation-en_US
Ign http://packages.linuxmint.com gloria/import Packages
Hit http://archive.ubuntu.com jaunty Release
Hit http://packages.medibuntu.org jaunty Release
Hit http://packages.linuxmint.com gloria/main Packages
Hit http://archive.ubuntu.com jaunty-updates Release
Hit http://archive.ubuntu.com jaunty/main Packages
```

3. ทำการติดตั้งโดยใช้คำสั่ง

```
Sudo apt-get install r-base
```



```
File Edit View Terminal Help
kwang@kwang-laptop ~ $ sudo aptitude install r-base
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
The following NEW packages will be installed:
build-essential{a} dpkg-dev{a} fakeroot{a} g++{a} g++-4.3{a}
gfortran{a} gfortran-4.3{a} libblas-dev{a} libblas3gf{a} libbz2-dev{a}
libgfortran3{a} libjpeg62-dev{a} liblapack-dev{a} liblapack3gf{a}
libncurses5-dev{a} libpcre3-dev{a} libpcrecpp0{a} libpng12-dev{a}
libreadline5-dev{a} libstdc++6-4.3-dev{a} patch{a} patchutils{a} r-base
r-base-core{a} r-base-dev{a} r-base-html{a} r-base-latex{a}
r-cran-boot{a} r-cran-cluster{a} r-cran-codetools{a} r-cran-foreign{a}
r-cran-kernsmooth{a} r-cran-lattice{a} r-cran-mgcv{a} r-cran-nlme{a}
r-cran-rpart{a} r-cran-survival{a} r-cran-vr{a} r-recommended{a}
tcl8.5{a} tk8.5{a} zlib1g-dev{a}
0 packages upgraded, 43 newly installed, 0 to remove and 232 not upgraded.
Need to get 0B/40.3MB of archives. After unpacking 137MB will be used.
Do you want to continue? [Y/n/?] █
```

จากภาพ ให้ กด Y เพื่อติดตั้ง แล้วรอกันกว่าจนเสร็จสมบูรณ์ ถือว่าเป็นการติดตั้งเสร็จสมบูรณ์

ฟังก์ชันพื้นฐาน ในการใช้ภาษา R - base

ฟังก์ชัน c() เพื่อสร้าง vector หรือ list

รูปแบบการใช้งาน :

```
ชื่อตัวแปร <- c(1,2)
```

ฟังก์ชัน scan() เป็นการรับค่าข้อมูล

รูปแบบการใช้งาน :

```
ชื่อตัวแปร <- scan() จะปรากฏส่วนที่ให้ทำการกำหนดค่า
```

ฟังก์ชัน edit() เป็นการแก้ไขข้อมูล

รูปแบบการใช้งาน :

```
ชื่อตัวแปร<-edit(ชื่อตัวแปร) จะปรากฏหน้าต่างข้อมูลของตัวแปรนั้น ซึ่ง  
สามารถทำการแก้ไขค่าได้
```

ฟังก์ชัน plot() เป็นการ plot กราฟแบบจุด

รูปแบบการใช้งาน :

```
plot(x,y) โดย x เป็นค่าตามแนวนอนและ y เป็นค่าตามแนว x
```

ฟังก์ชัน lines() เป็นการลากเส้นเชื่อมจุด

รูปแบบการใช้งาน :

```
lines(x,y) โดย x เป็นค่าตามแนวนอนและ y เป็นค่าตามแนว x
```

ฟังก์ชัน barplot() เป็นการ plot กราฟแบบกราฟแท่ง

รูปแบบการใช้งาน :

`barplot(x,y)` โดย x เป็นค่าตามแนวนอนและ y เป็นค่าตามแนว x

ฟังก์ชัน `pie()` เป็นการ plot กราฟแบบ `piechart`

รูปแบบการใช้งาน :

`pie(x,y)` โดย x เป็นค่าตามแนวนอนและ y เป็นค่าตามแนว x

ฟังก์ชัน `cbind()` เป็นการนำคอลัมน์มารวมกัน

รูปแบบการใช้งาน :

`cbind(ชื่อตัวแปร)`

ฟังก์ชัน `rnorm()` เป็นการสร้างค่าแบบสุ่ม

รูปแบบการใช้งาน :

`rnorm(มิติ)` เราสามารถกำหนดขนาดของมิติได้ภายในวงเล็บ

ฟังก์ชัน `data.entry()` เป็นการกำหนดค่าแบบแสดงเป็นตาราง โดยต้องมีการกำหนดค่าของตัวแปรนั้นอยู่ก่อนแล้วด้วย

รูปแบบการใช้งาน :

`data.entry(ชื่อตัวแปร)` จะปรากฏตารางเก็บค่าข้อมูลขึ้นมาให้กำหนดค่า

การนำเข้าข้อมูลจากไฟล์

การนำข้อมูลเข้ามาใช้ในโปรแกรมต้องกำหนด directory ให้โปรแกรม R เข้าถึงได้ เรียกว่า “working directory” (wd) ฟังก์ชันต่อไปนี้จะใช้จัดการกับ directory getwd() พิมพ์ โดยไม่ต้องมี argument จะได้ working directory ปัจจุบัน setwd(“dir”) กำหนดให้ working directory เป็น dir เช่น setwd(“D:/Documents and Setting/data1”) จะเป็นการเซตให้อยู่ที่ D:/Documents and Setting/data1 จากนั้นสามารถดำเนินการกับข้อมูลได้โดยใช้ฟังก์ชัน read.table() จะได้ผลลัพธ์เป็นดังนี้

men.size	female.size	childrens
167	158	2
165	162	NA
153	167	5

ฟังก์ชันเพิ่มเติม

ฟังก์ชัน typeof()

```
> x <- 1:3  
> typeof(x)  
[1] "integer"
```

เป็นฟังก์ชันที่ใช้ตรวจสอบค่า ของ x ว่าเป็นชนิดอะไร เช่น
"NULL" คือ เป็นค่าว่าง
"environment" คือ เป็นตัวแปรแบบเวดล้อม
"integer" คือ เป็นตัวแปรแบบเลขจำนวนเต็ม
"double" คือ เป็นตัวแปรแบบเลขจำนวนจริง
"expression" คือ เป็นสิ่งที่ใช้ดำเนินการกับตัวแปรต่างๆ เช่น เครื่องหมายบวก ลบ เป็นต้น

การกำหนดค่าให้กับตัวแปร

```
> y <- 4  
> y  
[1] 4
```

จากตัวอย่างข้างบนนี้ เป็นการกำหนดค่าให้กับตัวแปร y คือ ให้ y มีค่าเท่ากับ 4

การใช้ตัวดำเนินการกับตัวแปร

ตัวอย่างที่ 1

```
> 34+6  
[1] 40  
> 48*3  
[1] 144  
> 45/5  
[1] 9  
> 14-7  
[1] 7  
> (56+3)*7  
[1] 413
```

จากตัวอย่างข้างบนนี้ เป็นการบวก คูณ หาร ลบ กับค่าต่างๆ

ตัวอย่าง operators ต่างๆ เช่น

- < เป็นเครื่องหมายที่ใช้เปรียบเทียบค่าน้อยกว่า
- > เป็นเครื่องหมายที่ใช้เปรียบเทียบค่ามากกว่า
- == เป็นเครื่องหมายที่ใช้เปรียบเทียบค่าเท่ากับ
- >= เป็นเครื่องหมายที่ใช้เปรียบเทียบค่ามากกว่าหรือเท่ากับ
- <= เป็นเครื่องหมายที่ใช้เปรียบเทียบค่าน้อยกว่าหรือเท่ากับ
- ! เป็นเครื่องหมายที่ใช้เปรียบเทียบค่าที่ไม่เท่ากับ
- ? เป็นเครื่องหมายที่ใช้เรียกตัว help

ตัวอย่างที่ 2

```
> x <- 10; x + 56  
[1] 66
```

จากตัวอย่างข้างบนนี้ เป็นการบวกค่า โดยมีการกำหนดค่าให้กับ x ก่อน แล้วจึงนำ x ไปดำเนินการกับค่าอื่นๆ

การใช้คำสั่ง switch

```
> y <- "fruit"
> switch(y, fruit = "banana", vegetable = "broccoli", meat = "beef")
[1] "banana"
> y <- "meat"
> switch(y, fruit = "banana", vegetable = "broccoli", meat = "beef")
[1] "beef"
> y <- "vegetable"
> switch(y, fruit = "banana", vegetable = "broccoli", meat = "beef")
[1] "broccoli"
```

จากตัวอย่างข้างบนนี้เป็นการกำหนดตัวเลือก แล้วให้เราเลือกว่าจะเอาตัวเลือกอะไร เช่น ตัวเลือก fruit ถ้าเราเลือกผลลัพธ์ที่ออกมาจะเป็น banana

การบวกเลขหลายจำนวนพร้อมกัน

โดยใช้ตัวแปรรับค่า เช่น

```
> assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7)) หรือ
```

```
> c(10.4, 5.6, 3.1, 6.4, 21.7) -> x
```

```
> assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
> 1/x
[1] 0.09615385 0.17857143 0.32258065 0.15625000 0.04608295
> |
```

Index matrices

การสร้างเมทริกซ์

> x <- array(1:20, dim=c(4,5)) ใส่เลข 1-20 ลงในเมทริกซ์ขนาด 4*5

>x คือให้แสดงค่าตัวเลขใน array ออกมา เป็นเหมือนคำสั่ง print

```
> x <- array(1:20, dim=c(4,5))
> x
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    5    9   13   17
[2,]    2    6   10   14   18
[3,]    3    7   11   15   19
[4,]    4    8   12   16   20
> i <- array(c(1:3,3:1), dim=c(3,2))
> i
      [,1] [,2]
[1,]    1    3
[2,]    2    2
[3,]    3    1
```

คำนวณค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐาน

คำนวณค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของผลการไตเตรตดังนี้ 10.08, 10.11, 10.09,
10.10, 10.12 mL


```

> titrant <-c(10.08,10.11,10.09,10.10,10.12)
> mean(titrant)
[1] 10.1
> sd(titrant)
[1] 0.01581139
> titrant2 =c(10.08,10.11,10.09,10.10,10.12)
> mean(titrant2)
[1] 10.1
> sd(titrant2)
[1] 0.01581139
> |

```

จะเห็นว่าสามารถทำได้ ด้วยการกรอกข้อมูลลงไปพร้อมกับสร้างตัวแปรขึ้นมาเพื่อเก็บค่าไว้ ในที่นี้คือ titrant โดยใช้รูปแบบการกรอกดังตัวอย่างโดยใช้ฟังก์ชัน c() หลังจากนั้นจึงใช้คำสั่ง built-in สองอันคือ mean() กับ sd() เพื่อคำนวณค่าเฉลี่ย (arithmatic mean) และ ค่าเบี่ยงเบนมาตรฐาน (standard deviation) ในตัวอย่างด้านล่างเป็นการกรอกข้อมูลในอีกรูปแบบหนึ่งให้เก็บอยู่ในชื่อ titrant2 และรูปแบบการตั้ง โดยเปลี่ยนชื่อตัวแปร ซึ่งให้ผลเหมือนกัน

สร้าง random number

ใส่ตัวเลขที่ต้องการสุ่มในที่นี้คือเลข 1-30

```

> sample(1:30)
[1] 7 24 13 29 8 6 26 30 9 16 5 19 12 3 15 17 18 21 28 1 4 2 25 20 22 14 11 23 10 27
> |

```

การสร้างกราฟมาตรฐาน

การคำนวณ linear regression เพื่อหาเส้นตรงความสัมพันธ์ระหว่างข้อมูลสองชุด ด้วยวิธี least squares method อย่างเช่นการเตรียม calibration curve ในการวิเคราะห์ทางเคมี สามารถทำได้ดังตัวอย่างการวัดความเข้มของสี (intensity) กับความเข้มข้น(conc)ของสาร ดังนี้

```
> conc=c(0,2,4,6,8,10,12)
> intensity=c(2.1,5.0,9.0,12.6,17.3,21,24.7)
> model=lm(intensity~conc)
> model
```

```
Call:
lm(formula = intensity ~ conc)
```

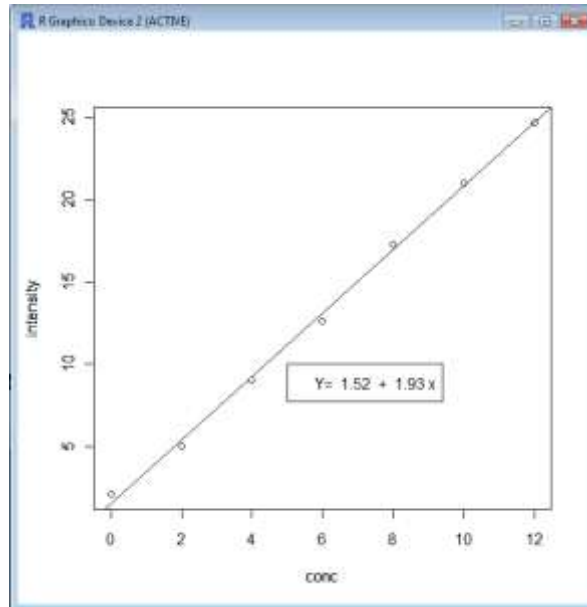
```
Coefficients:
(Intercept)      conc
      1.518      1.930
```

```
> |
```

นั่นคือเราได้สัมประสิทธิ์ตัวแรกพจน์แรกซึ่งเป็นจุดตัดแกน Y เป็น 1.518 และสัมประสิทธิ์ตัวที่สองคือความชันของเส้นกราฟเป็น 1.930

```
> plot(conc,intensity)
> abline(model$coef)
> txt<-paste("Y= ", format(model$coef[1],digits=3)," + ",format(model$coef[2],digits=3),"x")
> legend(5,10,txt)
> |
```

จากนั้นเราสามารถ Plot กราฟเส้นตรงนี้ได้ด้วยคำสั่ง plot () เพื่อแสดงจุดของข้อมูล แล้วใช้ abline() เพื่อแสดงเส้นตรงจากการคำนวณ จากนั้นแสดงสมการ $y=a+bx$ ได้โดยสร้างตัวแปรเช่น txt แล้วทำการแปะข้อความลงกราฟด้วยคำสั่ง legend()



การเปรียบเทียบค่าเฉลี่ยหลายค่า และการทำ box plot

ใส่ค่าให้กับตัวแปรที่ต้องการเปรียบเทียบ

```
> a<-c(102,100,101)
> b<-c(101,101,104)
> c<-c(97,95,99)
> d<-c(90,92,94)
```

ทำการวาด box-plot เพื่อดูข้อมูลการเปรียบเทียบค่าแบบคร่าวๆ

```
> signal=data.frame(a,b,c,d)
> signal
  a  b  c  d
1 102 101 97 90
2 100 101 95 92
3 101 104 99 94
> boxplot(signal, xlab="condition",ylab="Signal")
> |
```

