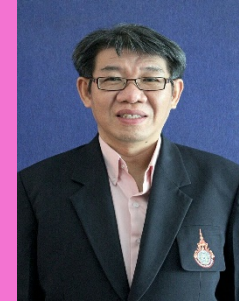
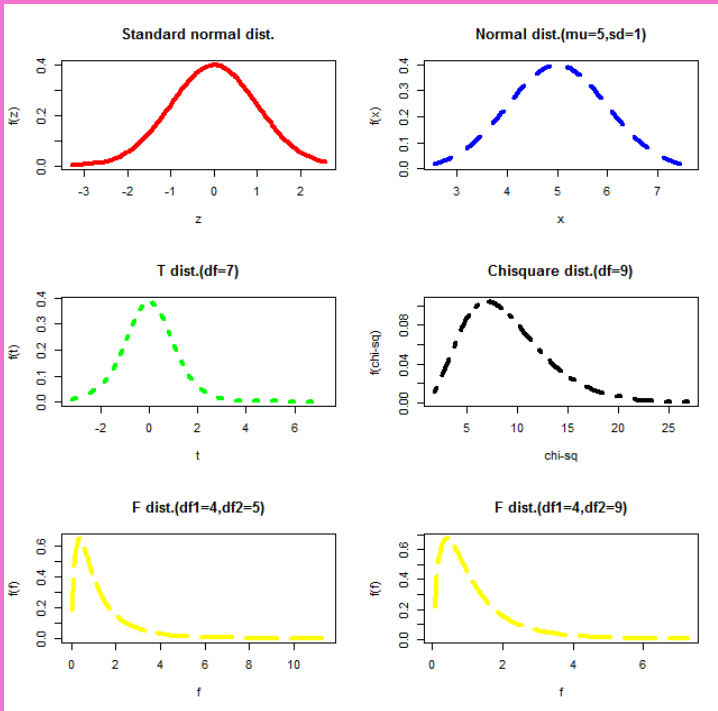


การวิเคราะห์ข้อมูลด้วยโปรแกรมอาร์

Data analytics with R



ผู้ช่วยศาสตราจารย์ ดร. อัจฉานท์ รัตนเลิศนุสรณ์

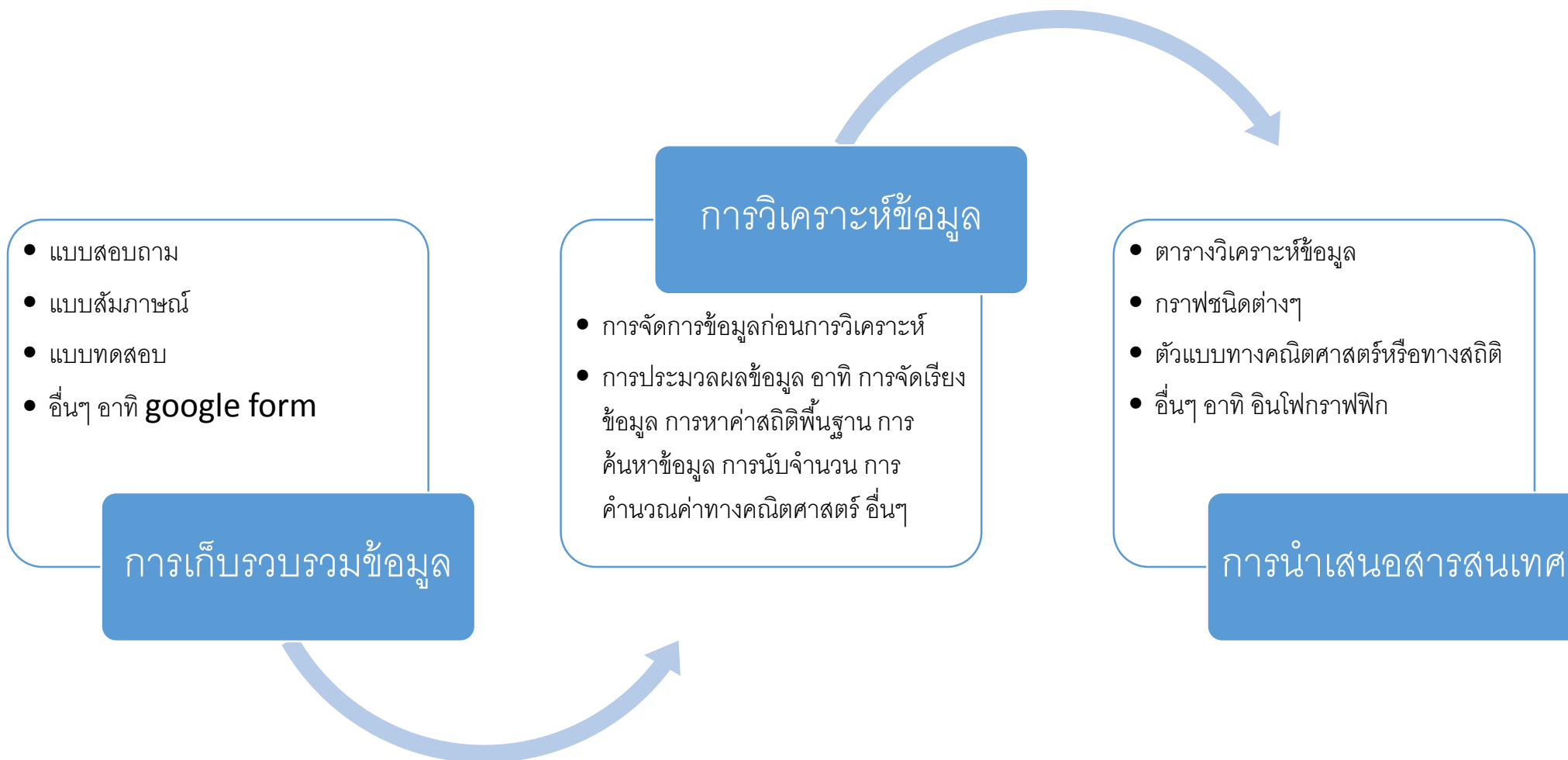
สาขาวิชาสถิติประยุกต์

คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยเทคโนโลยีราชมงคลธัญบุรี (มทร.ธัญบุรี)

ปทุมธานี ประเทศไทย

กระบวนการจัดการข้อมูล



การเก็บรวบรวมข้อมูลตามมาตรการวัดข้อมูล

ระดับการวัดข้อมูลทางสถิติมี 4 ระดับ คือ ข้อมูลระดับนามบัญญัติ ข้อมูลระดับเรียงลำดับ ข้อมูลระดับอันตรภาค และข้อมูลระดับอัตราส่วน

- **ข้อมูลระดับนามบัญญัติ** ข้อมูลระดับนี้ไม่สามารถเปรียบเทียบกันในลักษณะเชิงตัวเลข

อาทิ เพศ (ชาย, หญิง) สีต่างๆ(สีแดง สีเขียว สีเหลือง อื่นๆ) สัญลักษณ์ต่างๆ(พอใจ ไม่พอใจ เฉยๆ)

- **ข้อมูลระดับเรียงลำดับ** ข้อมูลระดับนี้สามารถเปรียบเทียบกันในลักษณะเชิงตัวเลขว่ามีค่ามากกว่า น้อยกว่า เท่ากัน หรือสามารถนำมาจัดเรียงลำดับจากน้อยไปมาก หรือมากไปน้อยได้ แต่ไม่สามารถบอกความแตกต่างระหว่างข้อมูลแต่ละค่าได้

อาทิ ความคิดเห็นหรือความพึงพอใจ (โดยทั่วไปแทนด้วยตัวเลข 5,4,3,2,1)

โดยที่ 5=ระดับความคิดเห็นด้วยมากที่สุด 4= มาก 3= ปานกลาง 2=น้อย 1=น้อยที่สุด

การเก็บรวบรวมข้อมูลตามมาตรการวัดข้อมูล

- **ข้อมูลระดับอันตรายภาค** ข้อมูลระดับนี้สามารถเปรียบเทียบกันในลักษณะเชิงตัวเลขว่ามีค่ามากกว่า น้อยกว่า เท่ากัน จัดเรียงลำดับ และหาความแตกต่างของข้อมูลแต่ละค่ารวมถึงค่าสถิติพื้นฐานของข้อมูลได้ โดยที่ข้อมูลระดับอันตรายภาคแต่ละค่าไม่ได้เริ่มต้นที่ศูนย์ (0)

อาทิ คะแนนสอบวิชาสถิติทั่วไปของนักศึกษา 3 คน (11, 15, 17 คะแนน) **คะแนนของนศ.แต่ละคนไม่ได้มีจุดเริ่มต้นที่ศูนย์**

หรือส่วนสูงของคนไข้ จำนวน 5 คน (160,157,170,164,158 เซนติเมตร) เป็นต้น **ส่วนสูงของคนไข้แต่ละคนไม่ได้มีจุดเริ่มต้นที่ศูนย์**

- **ข้อมูลระดับอัตราส่วน** ข้อมูลระดับนี้สามารถเปรียบเทียบกันในลักษณะเชิงตัวเลขที่ครอบคลุมข้อมูลระดับอันตรายภาคและยังสามารถเปรียบเทียบข้อมูลในรูปของอัตราส่วนได้เพราะมีจุดเริ่มต้นของการวัดที่ตำแหน่งเดียวกันคือตำแหน่งศูนย์ (0)

อาทิ ปริมาณสารเคมีในหลอดทดลองที่ 1,2,3 (5, 10,15 ml) **ปริมาณสารเคมีในแต่ละหลอดมีจุดเริ่มต้นที่ศูนย์**

หรือ ปริมาณยาพาราเซตามอลที่คนไข้ 5 คนรับประทาน (100,200,300,400,500 มิลลิกรัม ตามลำดับ) **ปริมาณยาพาราเซตามอลของคนไข้แต่ละคนมีจุดเริ่มต้นที่ศูนย์**

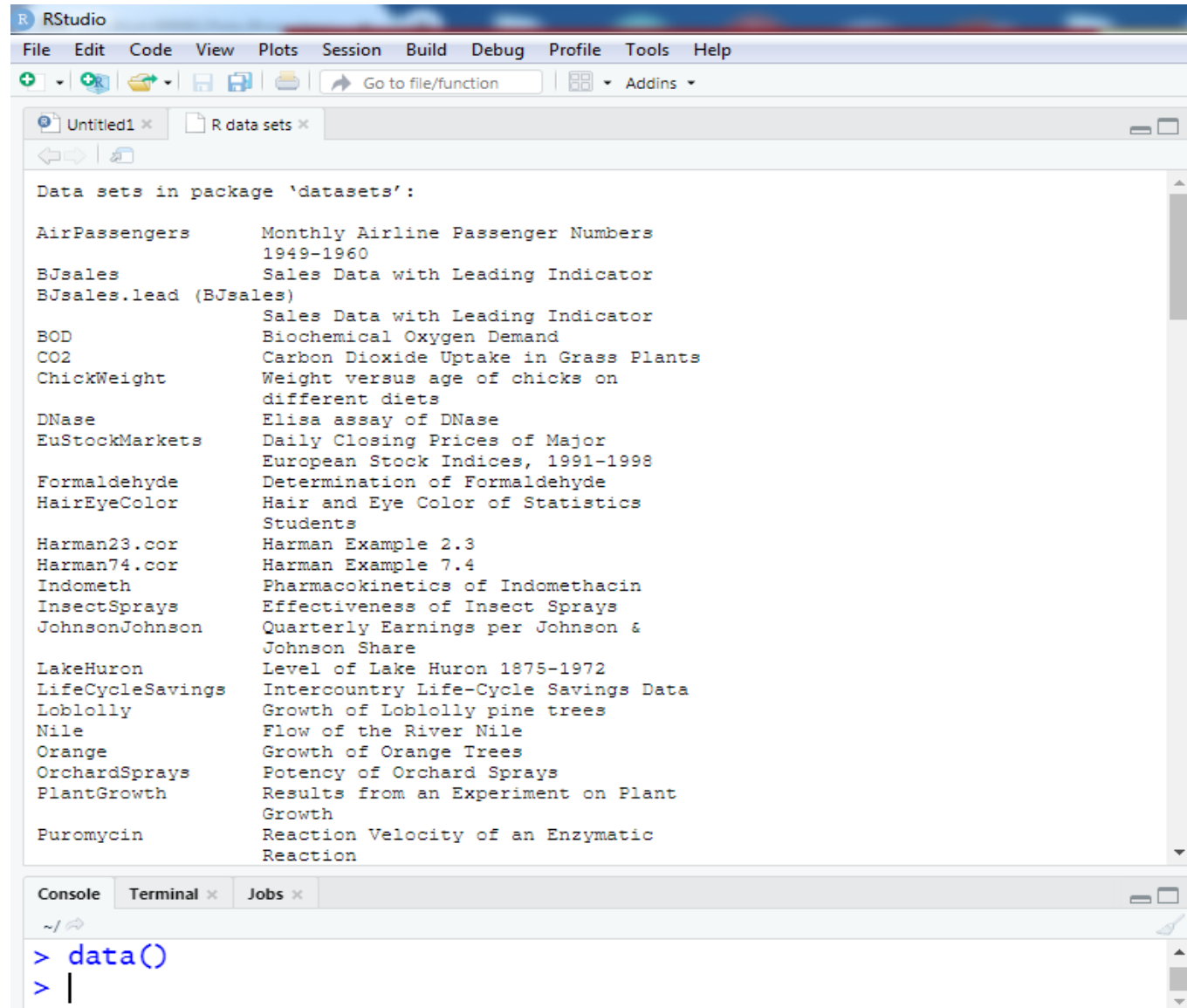
แหล่งข้อมูลที่น่าสนใจ ในโปรแกรม R

- โปรแกรมอาร์ มีชุดข้อมูล

build in มาให้

โดยสามารถเรียกดูจากคำสั่ง

>data()



The screenshot shows the RStudio environment. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for file operations and a search bar labeled 'Go to file/function'. The main editor window displays a list of data sets in the 'datasets' package. The console at the bottom shows the command '> data()' being entered.

Data sets in package 'datasets':	
AirPassengers	Monthly Airline Passenger Numbers 1949-1960
BJsales	Sales Data with Leading Indicator
BJsales.lead (BJsales)	Sales Data with Leading Indicator
BOD	Biochemical Oxygen Demand
CO2	Carbon Dioxide Uptake in Grass Plants
ChickWeight	Weight versus age of chicks on different diets
DNase	Elisa assay of DNase
EuStockMarkets	Daily Closing Prices of Major European Stock Indices, 1991-1998
Formaldehyde	Determination of Formaldehyde
HairEyeColor	Hair and Eye Color of Statistics Students
Harman23.cor	Harman Example 2.3
Harman74.cor	Harman Example 7.4
Indometh	Pharmacokinetics of Indomethacin
InsectSprays	Effectiveness of Insect Sprays
JohnsonJohnson	Quarterly Earnings per Johnson & Johnson Share
LakeHuron	Level of Lake Huron 1875-1972
LifeCycleSavings	Intercountry Life-Cycle Savings Data
Loblolly	Growth of Loblolly pine trees
Nile	Flow of the River Nile
Orange	Growth of Orange Trees
OrchardSprays	Potency of Orchard Sprays
PlantGrowth	Results from an Experiment on Plant Growth
Puromycin	Reaction Velocity of an Enzymatic Reaction

```
> data()
> |
```

การวิเคราะห์ข้อมูล (การจัดการกับข้อมูล)

โปรแกรมอาร์มีคำสั่งหรือฟังก์ชันที่ใช้จัดการกับข้อมูลดังนี้

คำสั่ง/ฟังก์ชัน	คำอธิบายแบบย่อ
<code>c()</code>	สร้างเวกเตอร์โดยนำข้อมูลมาเรียงต่อกัน
<code>seq(from,to)</code> <code>seq(from,to,by)</code> <code>seq(from,to,length)</code>	สร้างเวกเตอร์ตัวเลขเรียงลำดับ โดยกำหนดค่าเริ่มต้น และค่าสุดท้าย สร้างเวกเตอร์ตัวเลขเรียงลำดับโดยกำหนดค่าเริ่มต้น ค่าสุดท้าย และค่าที่เพิ่ม(หรือที่ลด)ในครั้งละ สร้างเวกเตอร์ตัวเลขเรียงลำดับโดยกำหนดค่าเริ่มต้น ค่าสุดท้าย และจำนวนสมาชิก
<code>rep(x,times,...)</code>	สร้างเวกเตอร์ตัวเลขหรือเวกเตอร์ข้อความที่สามารถกำหนดจำนวนซ้ำของข้อมูลแต่ละค่า
<code>as.integer(x,...)</code> <code>as.factor(x,...)</code> <code>as.numeric(x,...)</code>	ทำให้เวกเตอร์ x ให้เป็น integer ทำให้เวกเตอร์ x ให้เป็น factor ทำให้เวกเตอร์ x ให้เป็น numeric
<code>cbind()</code> <code>rbind()</code>	สร้างเมทริกซ์หรือ data frame จากเวกเตอร์ โดยนำข้อมูลเวกเตอร์มาเรียงต่อกันในแนวนอน คอลัมน์ หรือแนวแถว

การวิเคราะห์ข้อมูล (การจัดการกับข้อมูล)

คำสั่ง/ฟังก์ชัน	คำอธิบายแบบย่อ
<code>install.packages(...)</code> <code>library(...)</code>	ติดตั้ง package ที่ต้องการใช้ อาทิ <code>install.packages(readxl)</code> เรียกใช้หรือ load package ที่ต้องการใช้งาน อาทิ <code>library(readxl)</code>
<code>read.table(...)</code> <code>read.csv(...)</code> <code>read.spss(...)</code>	อ่านแฟ้มข้อมูลชนิด text file เข้ามาในโปรแกรมอาร์ อ่านแฟ้มข้อมูลชนิด csv file เข้ามาในโปรแกรมอาร์ อ่านแฟ้มข้อมูลชนิด spss เข้ามาในโปรแกรมอาร์
<code>read_excel(...)</code>	อ่านแฟ้มข้อมูล Microsoft Excel (*.xlsx หรือ *.xls) เข้ามาในโปรแกรมอาร์ ก่อนใช้ฟังก์ชันนี้ต้องเรียกใช้คำสั่ง <code>library(readxl)</code>
<code>write_xlsx(...)</code>	บันทึกแฟ้มข้อมูลเป็นไฟล์ *.xlsx สามารถไปเปิดโดย Microsoft Excel ก่อนใช้ฟังก์ชันนี้ต้องเรียกใช้คำสั่ง <code>library(writexl)</code>
<code>getwd()</code> <code>setwd()</code>	เรียกดู working directory ตั้งค่า working directory ใหม่

การวิเคราะห์ข้อมูล (การจัดการกับข้อมูล)

คำสั่ง/ฟังก์ชัน	คำอธิบายแบบย่อ
mean() median()	หาค่าเฉลี่ยเลขคณิต หาค่ามัธยฐาน
sd() var()	หาค่าส่วนเบี่ยงเบนมาตรฐานของข้อมูล หาค่าความแปรปรวนของข้อมูล
min() max()	หาค่าต่ำสุดในชุดข้อมูล หาค่าสูงสุดในชุดข้อมูล
length()	นับจำนวนข้อมูลในวัตถุ
table()	นับความถี่ของข้อมูลแต่ละค่า หรือการแจกแจงความถี่
summary()	สรุปค่าสถิติพื้นฐานของวัตถุ 6 ค่า คือ min, 1 st -Q, Median, Mean, 3 rd -Q, Max

การวิเคราะห์ข้อมูล (การจัดการกับข้อมูล)

คำสั่ง/ฟังก์ชัน	คำอธิบายแบบย่อ
<code>sample(x, replace=..., prob=...)</code>	ใช้สุ่มตัวอย่างข้อมูลจากเวกเตอร์ x แบบใส่คืน หรือแบบไม่ใส่คืน และสามารถกำหนดความน่าจะเป็นของการสุ่มข้อมูลแบบเท่ากัน หรือไม่เท่ากัน
<code>data.frame(...)</code> <code>is.data.frame(x)</code> <code>as.data.frame(x)</code>	ใช้สร้างวัตถุชนิด data frame ใช้ตรวจสอบวัตถุ x ว่าเป็น data frame หรือไม่ (ผลลัพธ์ที่ได้คือ TRUE หรือ FALSE) ใช้เปลี่ยนวัตถุ x ให้เป็นวัตถุชนิด data frame
<code>str(x)</code>	ใช้ตรวจสอบโครงสร้างข้อมูลภายในวัตถุ x
<code>round(x, digit)</code>	ใช้ปัดเศษตัวเลขที่อยู่ในเวกเตอร์ x ให้มีจำนวนทศนิยมตามที่ระบุใน digit
<code>replace()</code>	แทนที่ข้อมูลในเวกเตอร์หรือ data frame ด้วยค่าที่กำหนด

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

```
x <- c(1,2,4,6,9)
```

```
x
```

```
seq(1,10)
```

```
seq(1,20,by=3)
```

```
seq(1,10,length=30)
```

```
rep(c(2,4,6),3)
```

```
x <- c(1,3,5,7,9)
```

```
is.double(x)
```

```
xnew<- as.integer(x)
```

```
is.integer(xnew)
```

```
gender <- c("Male","Female")
```

```
is.character(gender)
```

```
genderfac<- as.factor(gender)
```

```
genderfac
```

```
is.factor(genderfac)
```

```
xnum <- c("1","2","3","4","5")
```

```
xnum
```

```
is.numeric(xnum)
```

```
xnumber<- as.numeric(xnum)
```

```
xnumber
```

```
A <- matrix(1:8,nrow=4,byrow=TRUE)
```

```
A
```

```
A <- cbind(A,c(9,10,11,12))
```

```
A
```

```
A <- rbind(A,c(13,14,15))
```

```
A
```

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

ผลลัพธ์ที่ได้

```
> x <- c(1,2,4,6,9)
> x
[1] 1 2 4 6 9
> seq(1,10)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(1,20,by=3)
[1] 1 4 7 10 13 16 19
> seq(1,10,length=30)
[1] 1.000000 1.310345 1.620690 1.931034 2.241379
[6] 2.551724 2.862069 3.172414 3.482759 3.793103
[11] 4.103448 4.413793 4.724138 5.034483 5.344828
[16] 5.655172 5.965517 6.275862 6.586207 6.896552
[21] 7.206897 7.517241 7.827586 8.137931 8.448276
[26] 8.758621 9.068966 9.379310 9.689655 10.000000
> rep(c(2,4,6),3)
[1] 2 4 6 2 4 6 2 4 6
> x <- c(1,3,5,7,9)
> is.double(x)
[1] TRUE
> xnew<- as.integer(x)
> is.integer(xnew)
[1] TRUE
> gender <- c("Male","Female")
> is.character(gender)
[1] TRUE
```

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

ผลลัพธ์ที่ได้

```
> genderfac<- as.factor(gender)
> genderfac
[1] Male   Female
Levels: Female Male
> is.factor(genderfac)
[1] TRUE
> xnum <- c("1","2","3","4","5")
> xnum
[1] "1" "2" "3" "4" "5"
> is.numeric(xnum)
[1] FALSE
> xnumber<- as.numeric(xnum)
> xnumber
[1] 1 2 3 4 5
> A <- matrix(1:8,nrow=4,byrow=TRUE)
> A
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
[4,]    7    8
```

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

ผลลัพธ์ที่ได้

```
> A <- cbind(A,c(9,10,11,12))
> A
      [,1] [,2] [,3]
[1,]     1     2     9
[2,]     3     4    10
[3,]     5     6    11
[4,]     7     8    12
> A <- rbind(A,c(13,14,15))
> A
      [,1] [,2] [,3]
[1,]     1     2     9
[2,]     3     4    10
[3,]     5     6    11
[4,]     7     8    12
[5,]    13    14    15
> |
```

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

```
#install.packages(ggplot2)
#library(ggplot2)
dfpop<- read.table("population.txt",header=TRUE)
dfpop
dfstdsco<- read.csv("stdscore.csv",header = TRUE)
dfstdsco
getwd()
setwd("C:/Users/Statistics")
getwd()
setwd("C:/Users/Statistics/Documents")
```

ตัวอย่างการใช้ฟังก์ชันการจัดการข้อมูล

ผลลัพธ์ที่ได้

```
> dfpop<- read.table("population.txt",header=TRUE)
> dfpop
  province population
1   Bangkok   8700000
2 Pathumthani   1500000
3 Nonthaburi   2000000
4 Samutsakorn   2500000
> dfstdsco<- read.csv("stdscore.csv",header = TRUE)
> dfstdsco
  stdname gender mtscore
1     AAA   Male      20
2     BBB Female      35
3     CCC   Male      40
4     DDD   Male      36
5     EEE Female      48
6     FFF Female      50
7     GGG Female      32
8     HHH   Male      16
9     III Female      28
10    JJJ Female      33
> getwd()
[1] "C:/Users/Statistics/Documents"
> setwd("C:/Users/Statistics")
> getwd()
[1] "C:/Users/Statistics"
> setwd("C:/Users/Statistics/Documents")
```

การนำเสนอข้อมูลด้วยกราฟ

- กราฟแสดงการกระจายข้อมูล (scatter plot)
- กราฟเส้น (line plot)
- กราฟแท่ง (histogram plot)
- กราฟวงกลม (pie plot)
- กราฟ box plot

การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ฟังก์ชัน plot มีรูปแบบการใช้งานดังนี้

```
plot(x, y, ...)
```

โดยที่

x เป็นข้อมูลบนแกน x

y เป็นข้อมูลบนแกน y

... เป็น argument อื่นๆของฟังก์ชัน plot อาทิ

type เป็นชนิดของกราฟสามารถระบุเป็น

“p”, “l”, “b”, “h”, “o”, “s” “n”

main เป็นชื่อของรูปกราฟสามารถระบุตามที่ต้องการได้

xlab, ylab เป็นชื่อของแกน x และแกน y ตามลำดับ สามารถระบุตามที่ต้องการได้

การนำเสนอข้อมูลด้วยฟังก์ชัน plot

`xlim, ylim` เป็นการกำหนดช่วงข้อมูลที่ต้องการแสดงบนแกน `x` และแกน `y` ตามลำดับ

`color` เป็นการกำหนดสีของกราฟสามารถระบุเป็นตัวเลข `1,2,3,4,5,...`

หรือ `“black”, “red”, “green”, “yellow”, “pink”, etc.`

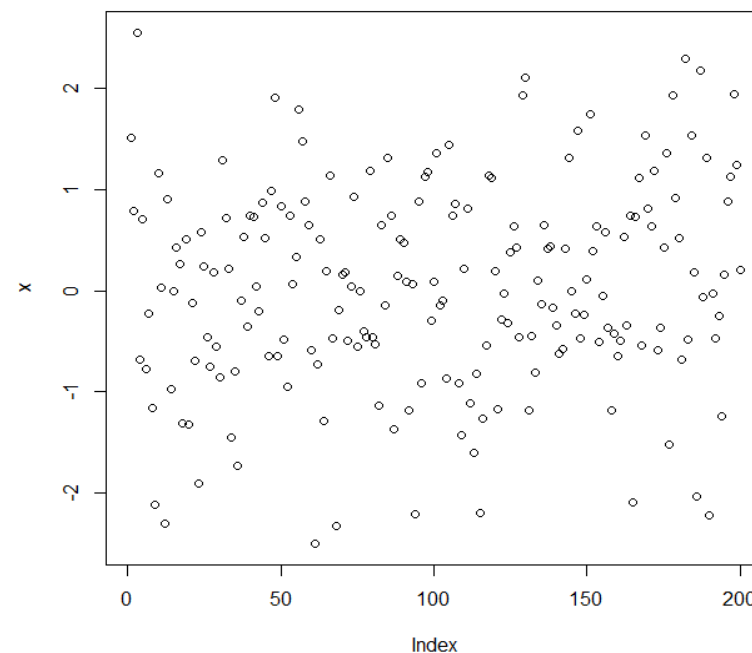
`sub` เป็นการตั้งชื่อ subtitle ของรูปภาพ

ตัวอย่างการใช้ฟังก์ชัน `plot`

```
x <- rnorm(200)
```

```
x
```

```
plot(x)
```



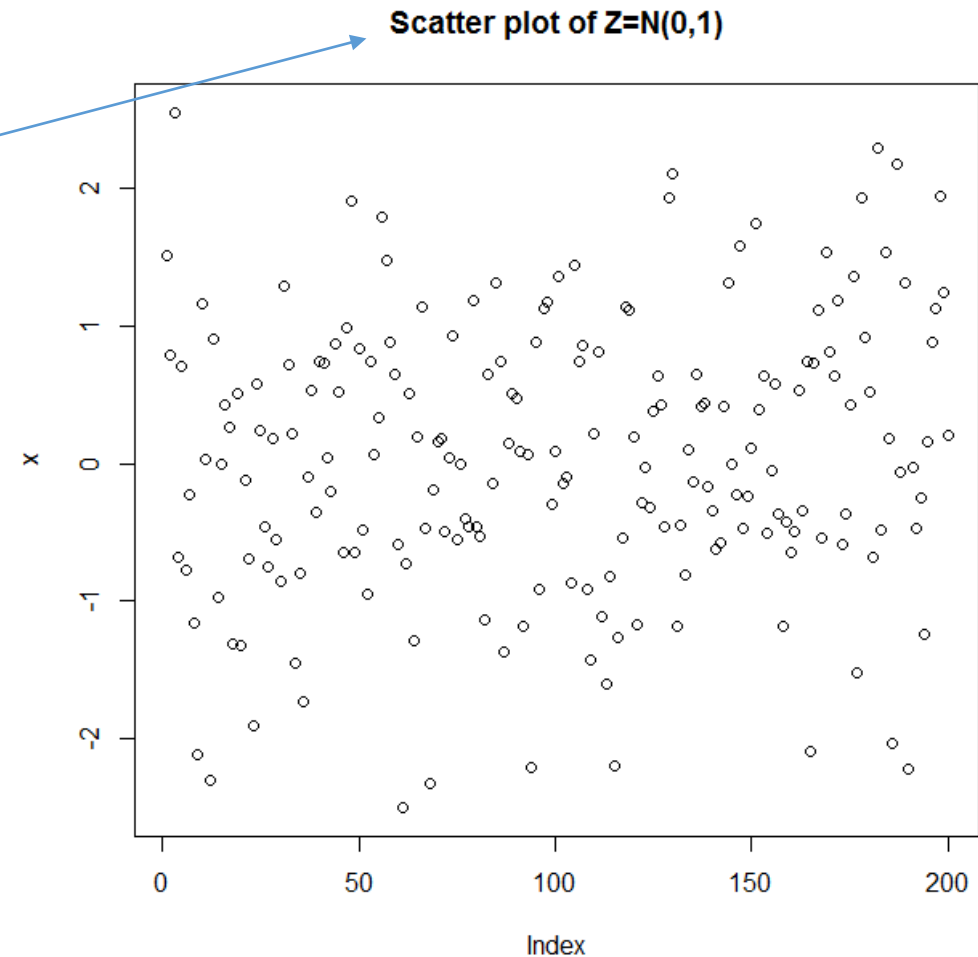
การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

```
x <- rnorm(200)
```

```
x
```

```
plot(x,main="Scatter plot of Z=N(0,1)")
```



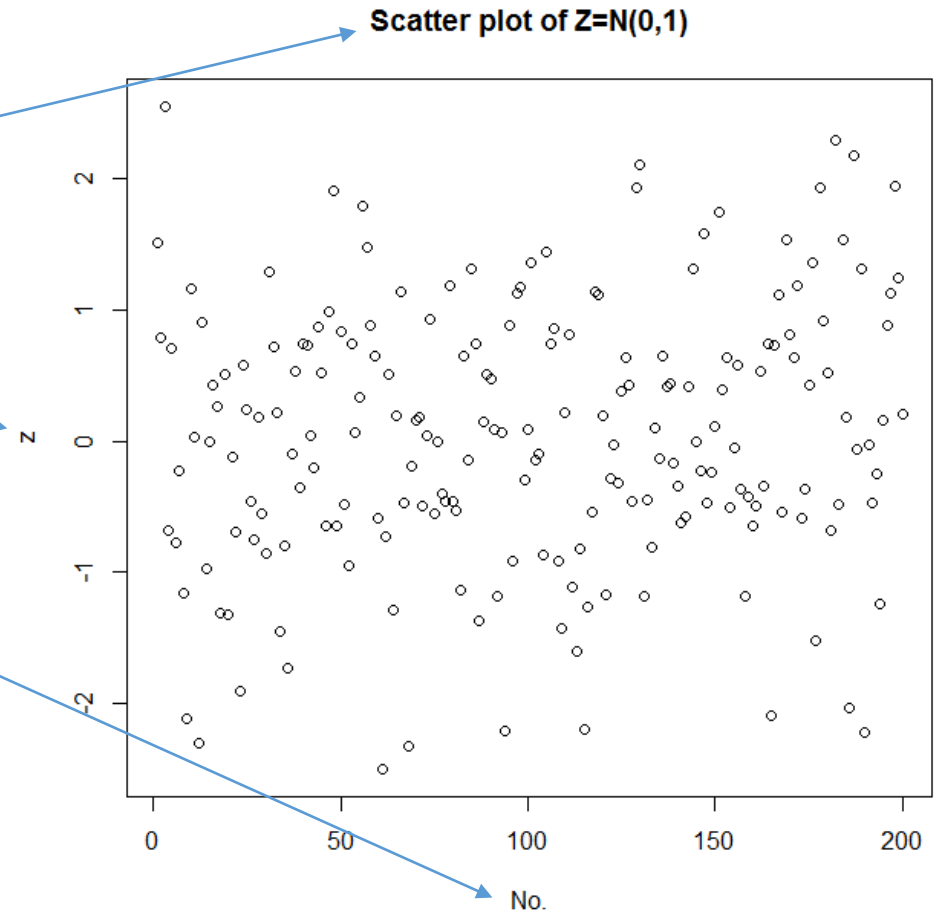
การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

```
x <- rnorm(200)
```

```
x
```

```
plot(x,main="Scatter plot of Z=N(0,1)",  
      xlab="No.",ylab="z")
```



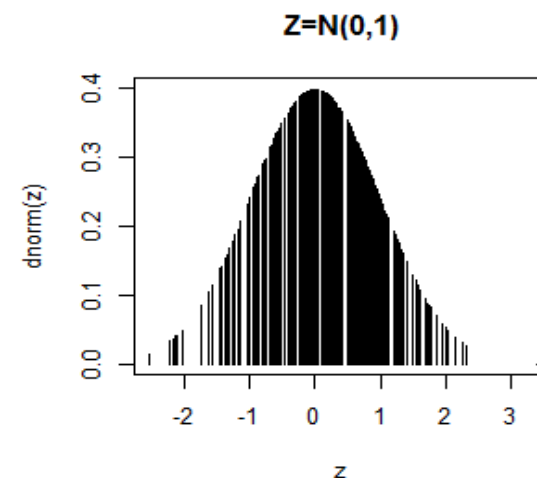
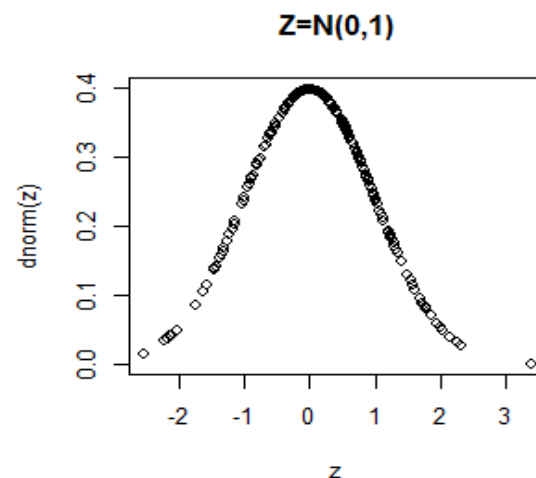
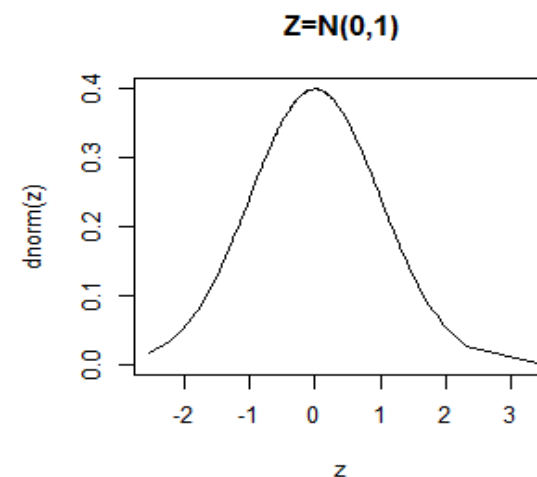
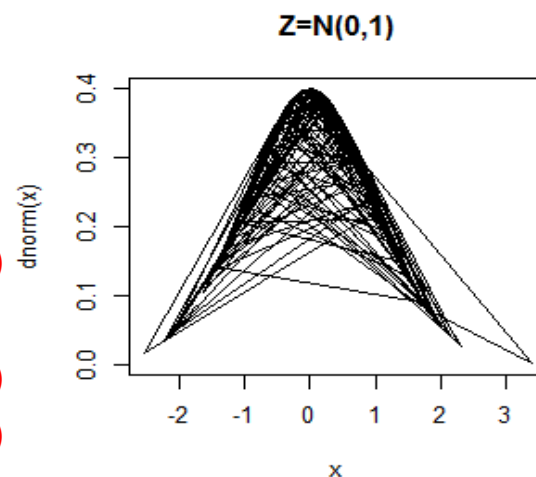
การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

```
par(mfrow=c(2,2))  
x <- rnorm(200)  
plot(x,dnorm(x),type="l",main="Z=N(0,1)")  
z<- sort(x)  
plot(z,dnorm(z),type="l",main="Z=N(0,1)")  
plot(z,dnorm(z),type="p",main="Z=N(0,1)")  
plot(z,dnorm(z),type="h",main="Z=N(0,1)")
```

คำสั่งเซตรูปภาพให้มีค่า 1 รูป (default)

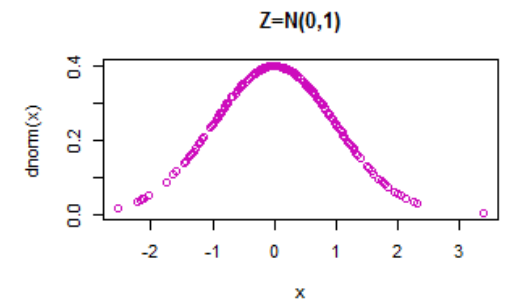
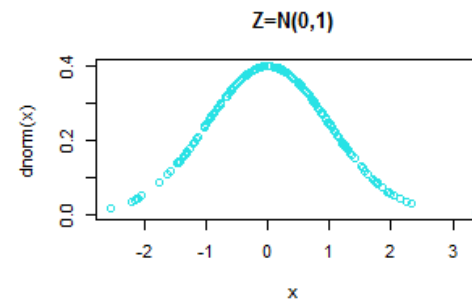
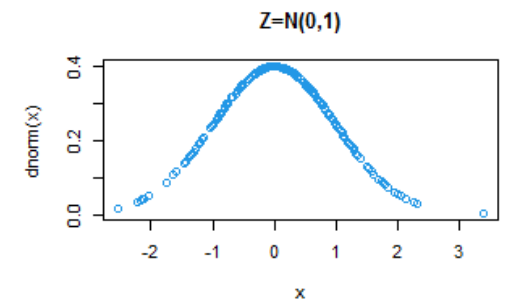
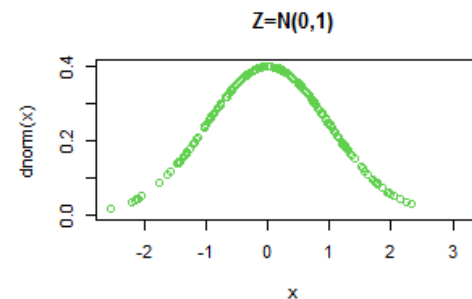
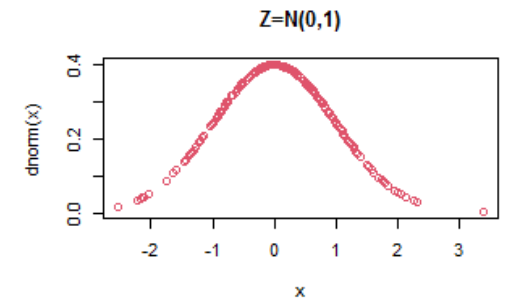
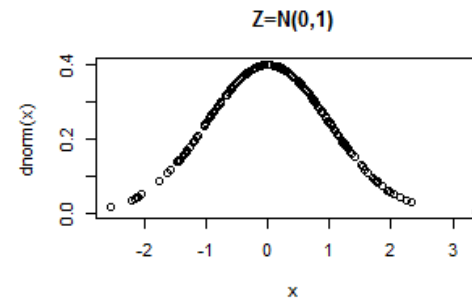
```
par()  
# par(mfrow=c(1,1)) setting 1 figure
```



การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

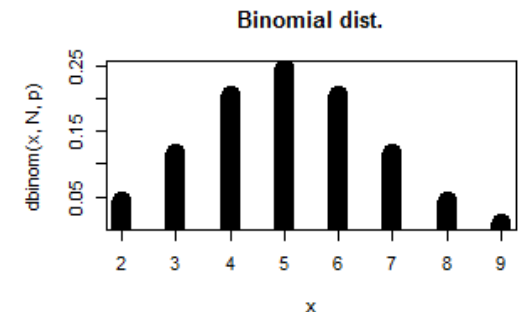
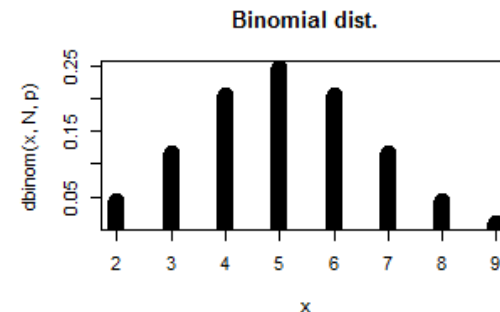
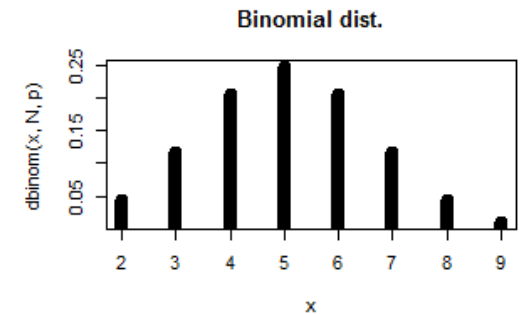
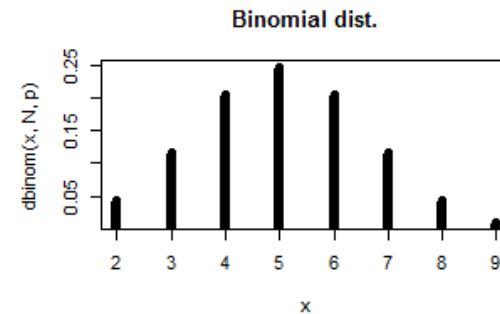
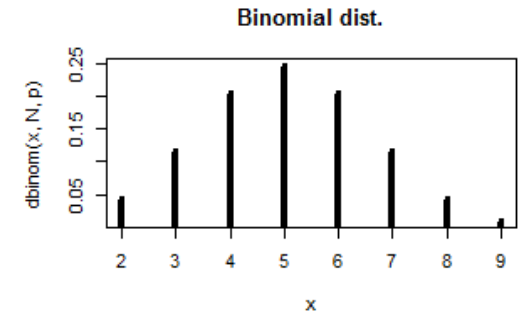
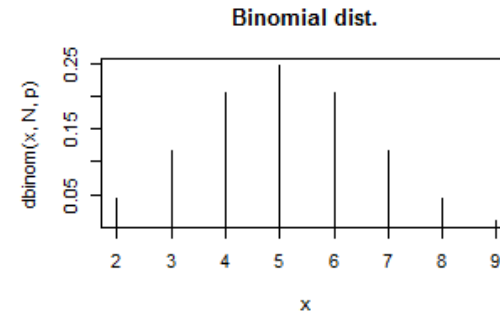
```
par(mfrow=c(3,2))      # 6 figures
plot(x,dnorm(x),type="p",main="Z=N(0,1)",col=1)
plot(x,dnorm(x),type="p",main="Z=N(0,1)",col=2)
plot(x,dnorm(x),type="p",main="Z=N(0,1)",col=3)
plot(x,dnorm(x),type="p",main="Z=N(0,1)",col=4)
plot(x,dnorm(x),type="p",main="Z=N(0,1)",col=5)
plot(x,dnorm(x),type="p",main="Z=N(0,1)",col=6)
```



การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

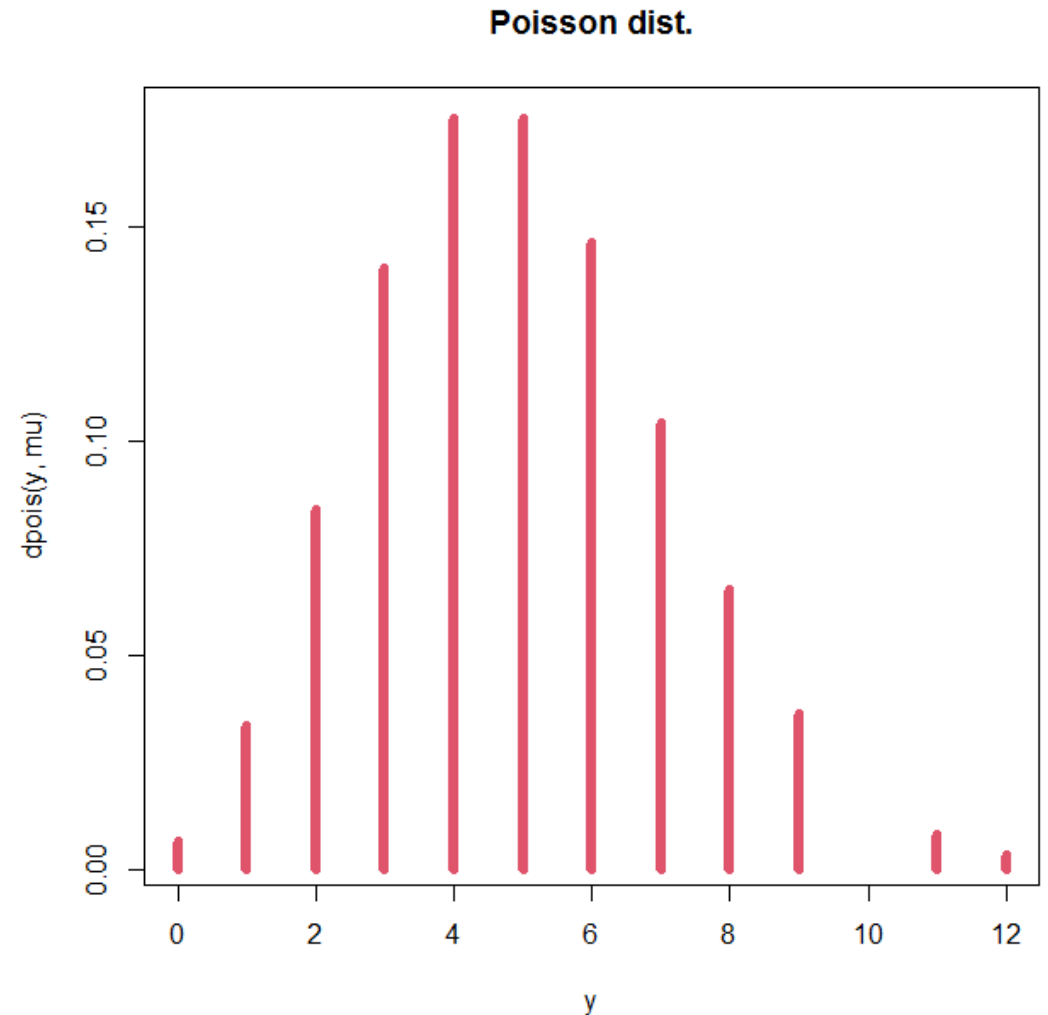
```
par(mfrow=c(3,2))      # 6 figures
p <- 0.5               # p=Prob of success
N <- 10                # N = number of trials
x<-rbinom(100,N,prob = p )
plot(x,dbinom(x,N,p),type="h",main="Binomial
dist.",col=1)
plot(x,dbinom(x,N,p),type="h",main="Binomial
dist.",col=1,lwd=4)
plot(x,dbinom(x,N,p),type="h",main="Binomial
dist.",col=1,lwd=6)
plot(x,dbinom(x,N,p),type="h",main="Binomial
dist.",col=1,lwd=8)
plot(x,dbinom(x,N,p),type="h",main="Binomial
dist.",col=1,lwd=10)
plot(x,dbinom(x,N,p),type="h",main="Binomial
dist.",col=1,lwd=12)
```



การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

```
mu <- 5  
y <- rpois(100,mu)  
plot(y,dpois(y,mu),type="h",main="Poisson  
dist.",col=2,lwd=6)
```



การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

```
par(mfrow=c(3,2)) # 6 figures
z<-rnorm(200)
zsort <- sort(z)
plot(zsort,dnorm(zsort),main="Standard normal dist.",
     type="l",col="red",xlab="z",ylab="f(z)",lty=1,lwd=4)
x1 <- rnorm(200,5,1)
x1s <- sort(x1)
plot(x1s,dnorm(x1s,5,1),main="Normal dist.(mu=5,sd=1)",
     type="l",col="blue",xlab="x",ylab="f(x)",lty=2,lwd=4)
x2 <- rt(200,df=7)
x2s <- sort(x2)
plot(x2s,dt(x2s,df=7),main="T dist.(df=7)",
     type="l",col="green",xlab="t",ylab="f(t)",lty=3,lwd=4)
x3 <- rchisq(200,df=9)
x3s <- sort(x3)
plot(x3s,dchisq(x3s,df=9),main="Chisquare dist.(df=9)",
     type="l",col="black",xlab="chi-sq",ylab="f(chi-sq)",lty=4,lwd=4)
```

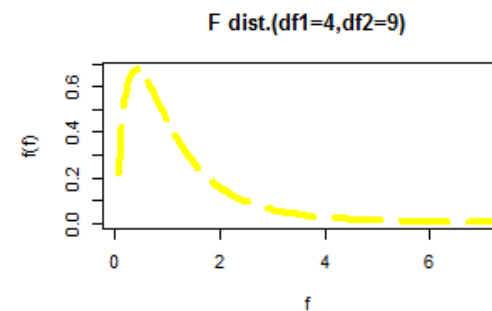
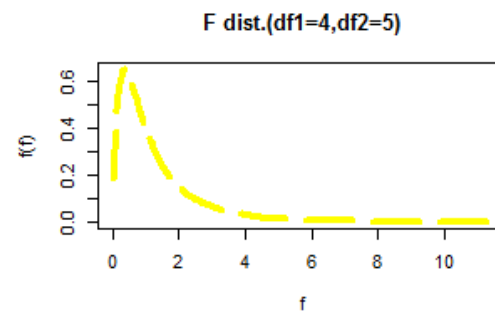
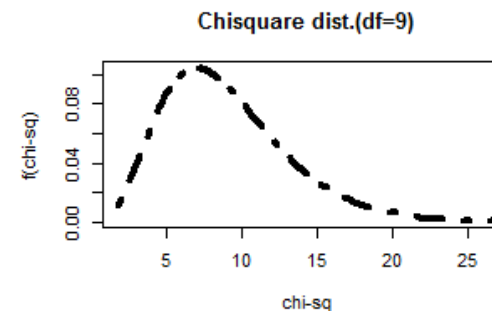
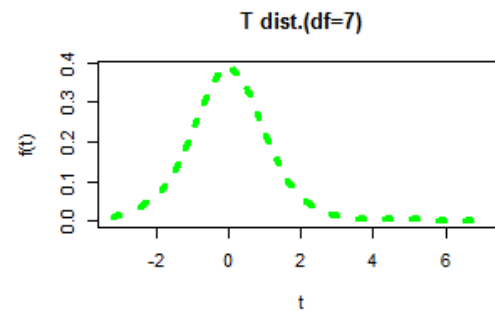
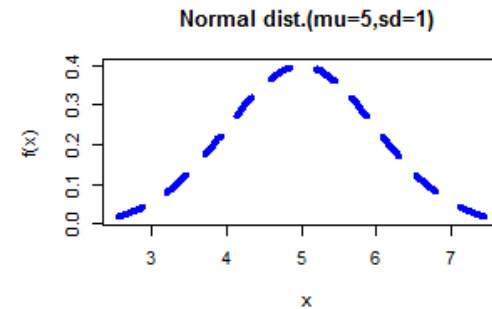
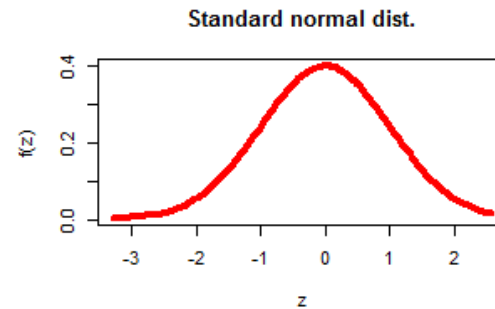
การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ตัวอย่างการใช้ฟังก์ชัน plot

```
x4 <- rf(200,df1=4,df2=5)
x4s <- sort(x4)
plot(x4s,df(x4s,df1=4,df2=5),main="F dist.(df1=4,df2=5)",
     type="l",col="yellow",xlab="f",ylab="f(f)",lty=5,lwd=4)
x5 <- rf(200,df1=4,df2=9)
x5s <- sort(x5)
plot(x5s,df(x5s,df1=4,df2=9),main="F dist.(df1=4,df2=9)",
     type="l",col="yellow",xlab="f",ylab="f(f)",lty=5,lwd=4)
```

การนำเสนอข้อมูลด้วยฟังก์ชัน plot

ผลลัพธ์ที่ได้จากฟังก์ชัน plot





Q&A

