



สถิติและการใช้โปรแกรม R และ R studio

ณัฐพร ภัคดี

เขตอุตสาหกรรมซอฟต์แวร์ภาคตะวันออก มหาวิทยาลัยบูรพา

14 – 15 มกราคม พ.ศ. 2560



Why use R ?

- ▶ FREE
- ▶ Fast
- ▶ Good for statistical modelings
- ▶ Strong community
(R-blogger, Twitter(#rstats), Stackoverflow)
- ▶ Great online learning resources



การติดตั้งโปรแกรม (Installation) R

- ▶ การติดตั้ง R สำหรับ Linux <https://cran.r-project.org/bin/linux/>
- ▶ การติดตั้ง R สำหรับ MacOS X <https://cran.r-project.org/bin/macosx/>
- ▶ การติดตั้ง R สำหรับ windows <https://cran.r-project.org/bin/windows/>

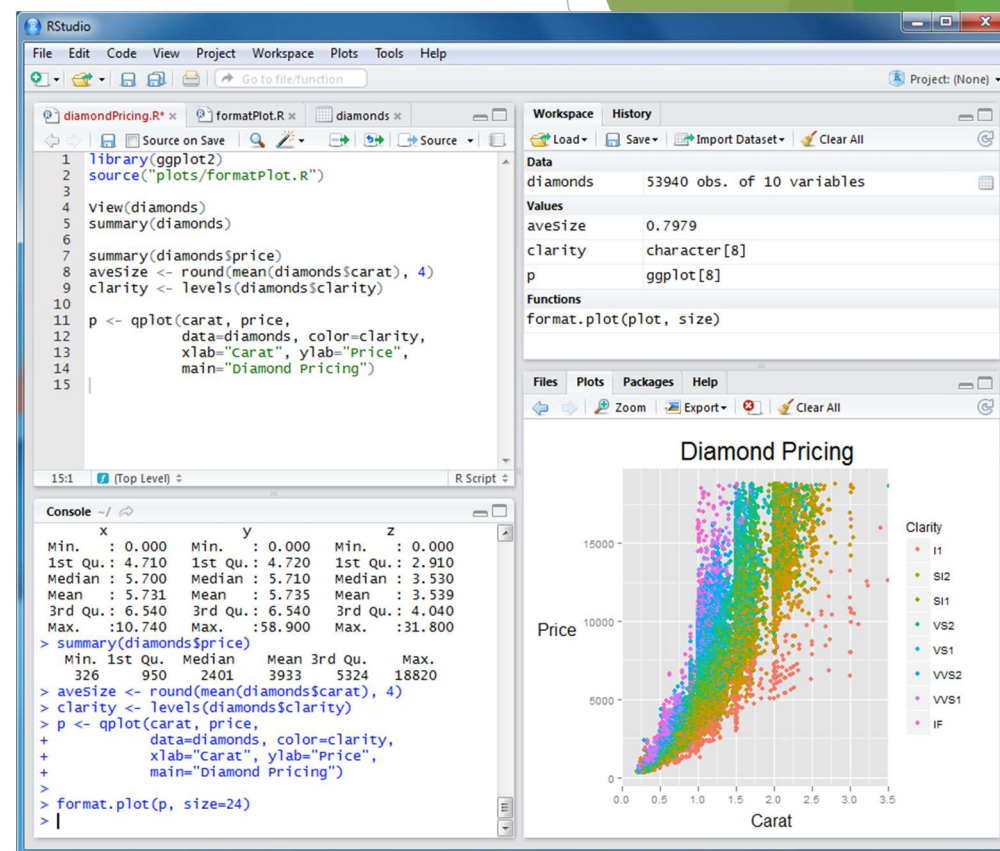
การติดตั้งโปรแกรม (Installation) RStudio

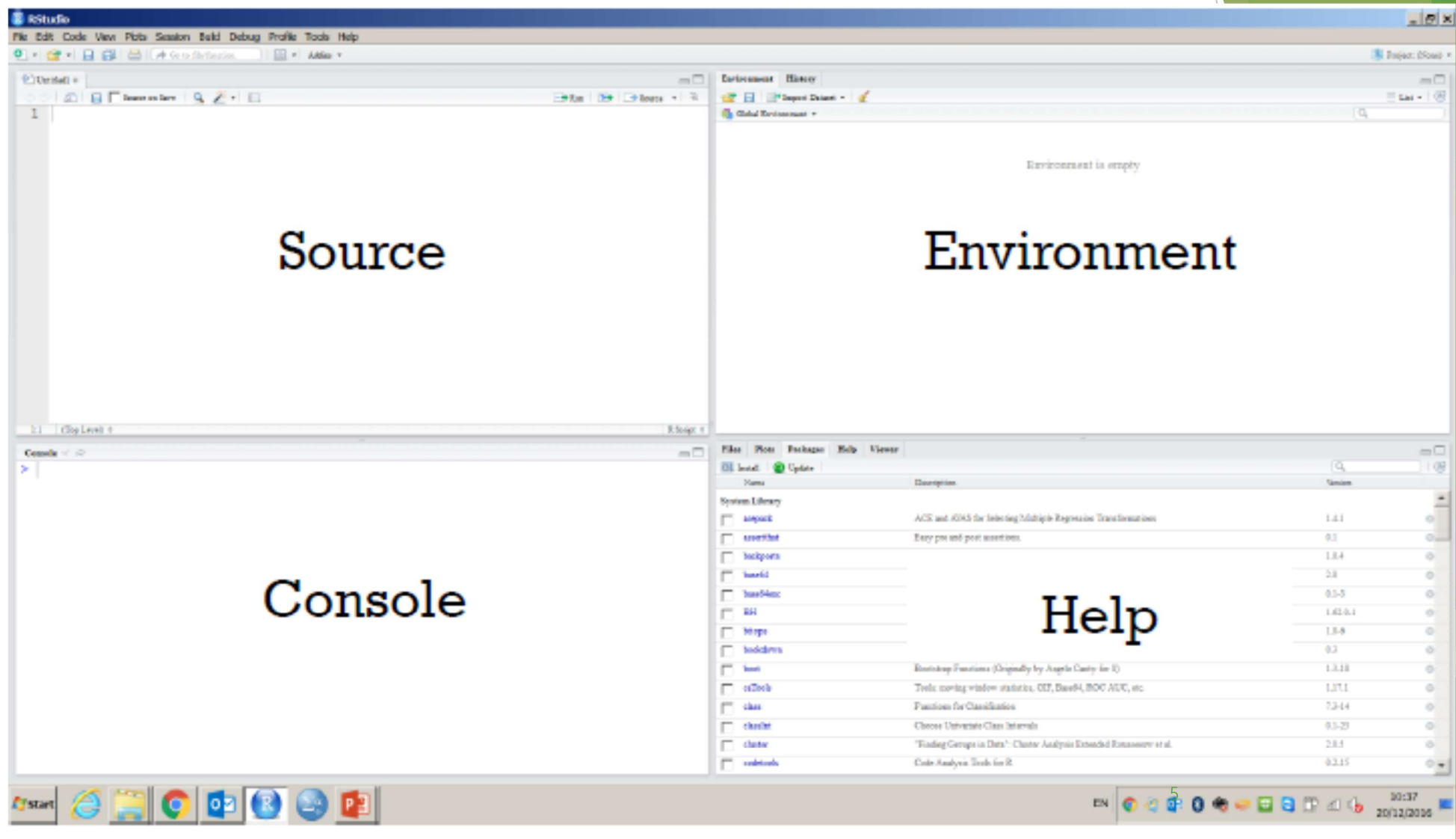
<http://www.rstudio.org/>

- ▶ เลือก RStudio for Desktop

Rstudio

- ▶ การกำหนด directory สำหรับการทำงาน
 - ▶ setwd()
 - ▶ setwd("c:/RFiles")
- ▶ ตรวจสอบ Working directory ปัจจุบัน
 - ▶ getwd()
- ▶ คำสั่งช่วยเหลือใน R
 - ▶ help(...)





Source

Environment

Console

Help

การนำเข้าข้อมูล

► Excel

read in the first worksheet from the workbook myexcel.xlsx

first row contains variable names

```
library(xlsx)
```

```
mydata <- read.xlsx("c:/myexcel.xlsx", 1)
```

read in the worksheet named mysheet

```
mydata <- read.xlsx("c:/myexcel.xlsx", sheetName = "mysheet")
```

การนำเข้าข้อมูล

► SPSS

save SPSS dataset in trasport format

```
get file="c:\mydata.sav".
```

```
export outfile="c:\mydata.por".
```

in R

```
library(Hmisc)
```

```
mydata <- spss.get("c:/mydata.por", use.value.labels=TRUE)
```


การนำเข้าข้อมูล

► SAS

save SAS dataset in trasport format

```
libname out xport "c:/mydata.xpt";  
data out.mydata;  
set sasuser.mydata;  
run;
```

in R

```
library(Hmisc)  
mydata <- sasxport.get("c:/mydata.xpt")
```


การนำเข้าข้อมูล

► Stata

input Stata file

```
library(foreign)
```

```
mydata<-read.dta("c:/mydata.dta")
```

► systat

input Systat file

```
library(foreign)
```

```
mydata <- read.systat("c:/mydata.dta")
```

► การนำเข้าข้อมูล (Import data)

► R เป็นโปรแกรมที่เหมาะสมสำหรับการวิเคราะห์ข้อมูล การป้อนข้อมูลลงใน R นั้นมักจะเก็บอยู่ในรูปแบบอื่นๆ มาก่อน เช่น spreadsheet (MS Excel) หรือ SPSS แล้วค่อยนำเข้า ข้อมูลดังกล่าวเข้ามาวิเคราะห์ใน R โดยสามารถนำเข้าข้อมูลด้วยคำสั่ง Command ด้านล่างนี้

► นำเข้าไฟล์ข้อมูลนามสกุล .csv เช่น data.csv

► `data = read.table ("c:/data.csv", header=TRUE, sep="," , row.names="id")`

► นำเข้าไฟล์ข้อมูลนามสกุล .txt

► `data = read.table ("c:/data.txt", header=FALSE, sep=" " , row.names="id")`

การส่งออกข้อมูลใน R

► Text File

```
write.table(mydata, "c:/mydata.txt", sep="\t")
```

► Excel Spreadsheet

```
library(xlsx)
```

```
write.xlsx(mydata, "c:/mydata.xlsx")
```

การส่งออกข้อมูลใน R

► SPSS

```
library(foreign)
```

```
write.foreign(mydata, "c:/mydata.txt",  
              "c:/mydata.sps", package="SPSS")
```

► Stata

```
library(foreign)
```

```
write.dta(mydata, "c:/mydata.dta")
```

การจัดการข้อมูล (Data Management)

▶ การส่งออกข้อมูล (Export data)

▶ นำเข้าไฟล์ข้อมูลนามสกุล .csv เช่น data.csv

▶ **write.csv** (ชื่อตัวแปร, “c:/ชื่อไฟล์.csv”, header=TRUE, sep=“,”,
row.names = TRUE, col.names =TRUE)

▶ **write.table** (ชื่อตัวแปร, “ชื่อไฟล์”, header=TRUE, sep=“ ”,
row.names=TRUE, col.names =TRUE)



การติดตั้ง และ Update Packages ใน R

```
install.packages("package name")
```

```
update.packages("package name")
```

ประเภทของข้อมูลใน R

- ▶ Numeric
- ▶ Integer
- ▶ Character
- ▶ Logical (True/False)
- ▶ Complex (rarely used)

การคำนวณพื้นฐานใน R

- ▶ การบวก (+)
- ▶ การลบ (-)
- ▶ การคูณ (*)
- ▶ การหาร (/)
- ▶ ยกกำลัง (^)
- ▶ รากที่สอง ($\sqrt{a}$)
- ▶ ค่าสัมบูรณ์ ($|a|$)

```
Console --/ ↺
> 3+3
[1] 6
> 5-2
[1] 3
> 5*4
[1] 20
> 10/2
[1] 5
> 5^2
[1] 25
> sqrt(4)
[1] 2
> abs(-5)
[1] 5
```

ข้อมูลที่เป็น Character ต้องเขียนภายใต้ “....”

Console D:/Generate Code By R/Code_gen/Code_gen/ ↻

```
> Hello world!  
Error: unexpected symbol in "Hello world"  
> "Hello world!"  
[1] "Hello world!"  
>
```

Logical Operators

- ▶ เท่ากับ
- ▶ ไม่เท่ากับ
- ▶ มากกว่า
- ▶ น้อยกว่า
- ▶ มากกว่าหรือเท่ากับ
- ▶ น้อยกว่าหรือเท่ากับ

Console D:/Generate Code By R/Code_gen/Cod

```
> 17==16  
[1] FALSE  
> 17!=16  
[1] TRUE  
> 17>16  
[1] TRUE  
> 17<16  
[1] FALSE  
> 17>=16  
[1] TRUE  
> 17<=16  
[1] FALSE  
> |
```

สัญลักษณ์พิเศษ

- ▶ inf (Infinity)
- ▶ NaN (Not a Number)
- ▶ NA (Not Available)

```
Console D:/Generate Code By R/Code_gen/Code_gen/
> 1/0
[1] Inf
> 0/0
[1] NaN
> new=c(1,2,3,NA,5,6,10,NA)
> new
[1] 1 2 3 NA 5 6 10 NA
> |
```

ข้อมูลใน R

► vectors

- ข้อมูลพื้นฐานสำหรับโปรแกรม R เป็นข้อมูลแบบเวกเตอร์ มีข้อมูลตัวเดียวหรือหลายตัวก็ได้ แต่จะต้องเป็นข้อมูลประเภทเดียวกัน เช่น character, numeric, Integer, complex number และ logical(True/False)

- คำสั่งนำเข้าข้อมูลในรูปเวกเตอร์

- `X=c(10,5,23,18)`

- `Y=c(TRUE,FALSE,FALSE,TRUE)`

- `Z=c("Hello", "World", "Thanks")`

► นำเข้าข้อมูลแบบเวกเตอร์โดยใช้คำสั่ง `scan()`

```
>x=scan()
```

```
>1 3 2 3 3 3 4 2 1 2 2 1 1 4 3 3 2 1 2 3
```

(การเว้นวรรคมีความหมายเช่นเดียวกับ , และการขึ้นบรรทัดใหม่หมายถึงการสิ้นสุดการนำเข้าข้อมูลของตัวแปร x)

► หากต้องการนำเข้าข้อมูลต่างประเภทกัน เช่น เป็นตัวเลขบ้าง ข้อความบ้าง สามารถนำเข้าได้โดยใช้คำสั่ง `list` และอ้างข้อมูลเฉพาะตำแหน่งได้

```
> l=list('male',34, 'single', 67.8)
```

```
> l[3]
```

การกระทำของเวกเตอร์

► การบวก

► การลบ

► การคูณ

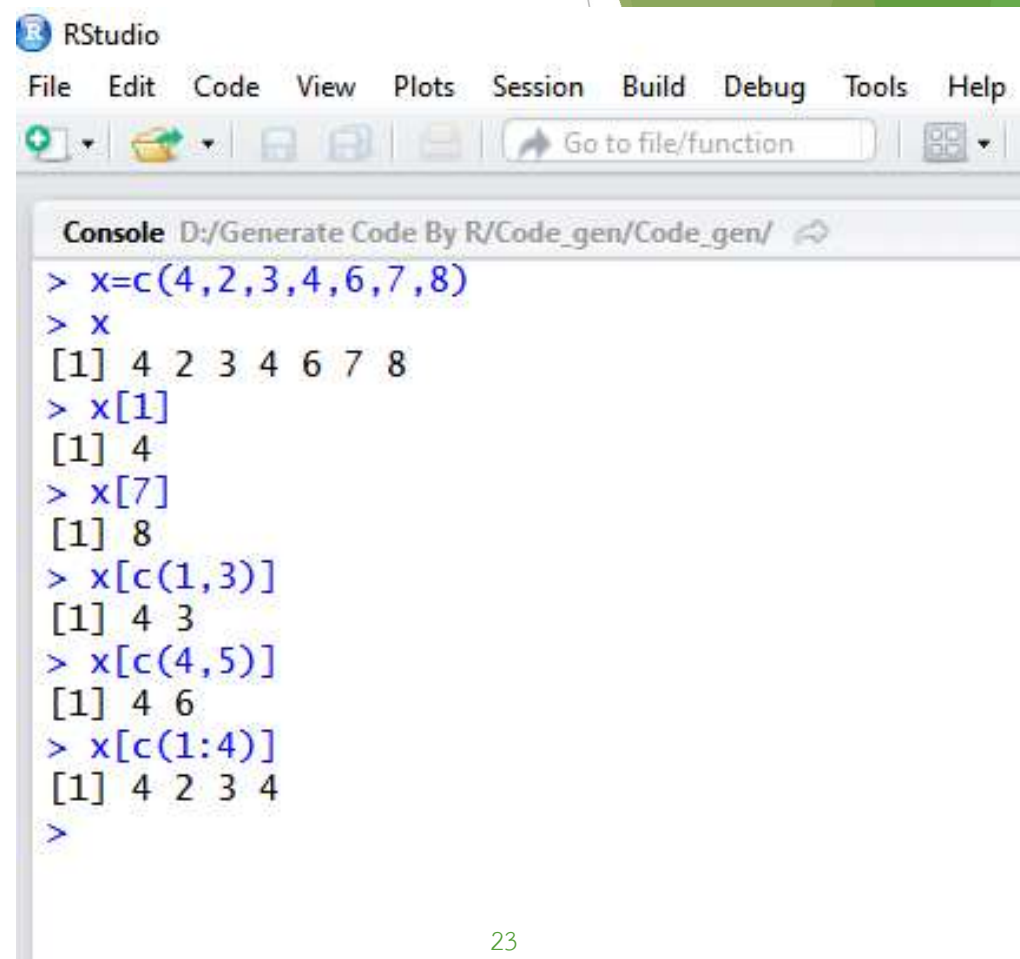
► การหาร

Console D:/Generate Code By R/Code_gen/Code_gen/ ↻

```
> x=c(1:4)
> y=c(6:9)
> x
[1] 1 2 3 4
> y
[1] 6 7 8 9
> x+y
[1] 7 9 11 13
> x-y
[1] -5 -5 -5 -5
> x*y
[1] 6 14 24 36
> x/y
[1] 0.1666667 0.2857143 0.3750000
[4] 0.4444444
> |
```

Subsetting a vector

- ▶ แสดงค่าบางตัวในเวกเตอร์
 - ▶ ชื่อเวกเตอร์[ลำดับ]
 - ▶ ชื่อเวกเตอร์[c(ลำดับ1, ลำดับ2)]
 - ▶ ชื่อเวกเตอร์[c(ลำดับ 1: ลำดับ n)]



RStudio

File Edit Code View Plots Session Build Debug Tools Help

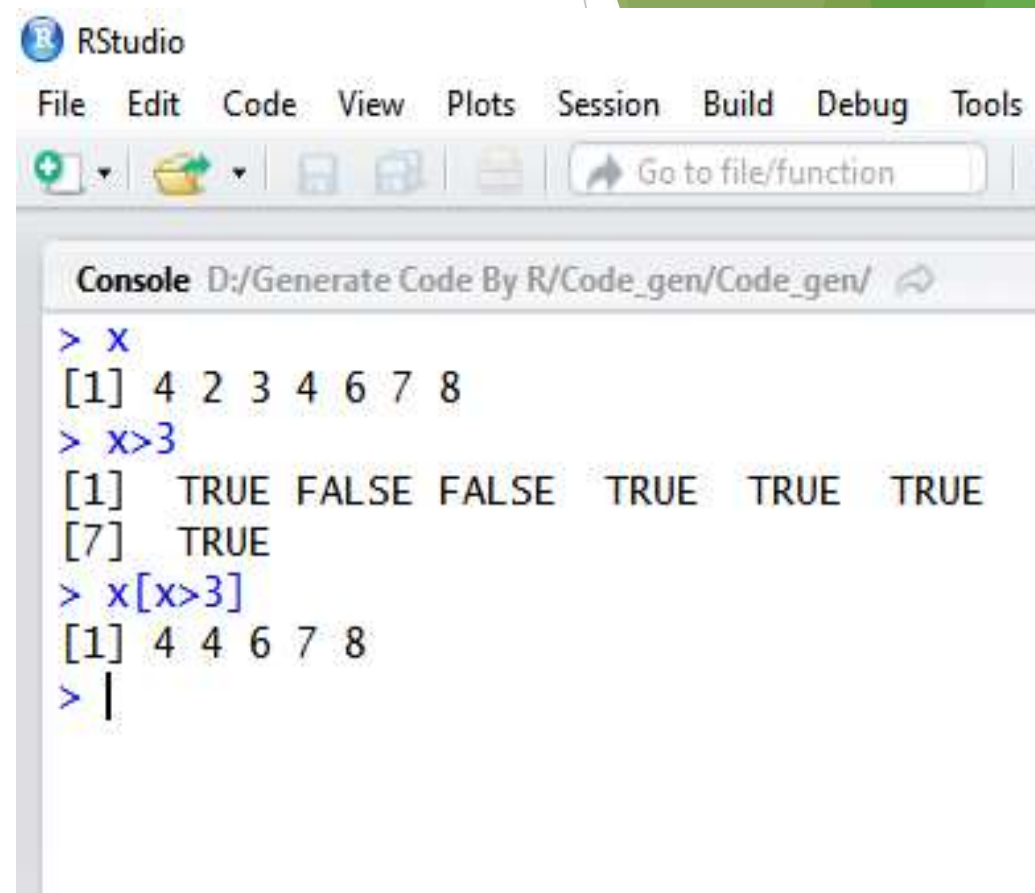
Go to file/function

Console D:/Generate Code By R/Code_gen/Code_gen/

```
> x=c(4,2,3,4,6,7,8)
> x
[1] 4 2 3 4 6 7 8
> x[1]
[1] 4
> x[7]
[1] 8
> x[c(1,3)]
[1] 4 3
> x[c(4,5)]
[1] 4 6
> x[c(1:4)]
[1] 4 2 3 4
>
```


Subsetting a vector

- ▶ แสดงค่าบางตัวในเวกเตอร์ด้วยเงื่อนไข
 - ▶ เวกเตอร์ เงื่อนไข
 - ▶ เวกเตอร์[เวกเตอร์ เงื่อนไข]



RStudio

File Edit Code View Plots Session Build Debug Tools

Go to file/function

Console D:/Generate Code By R/Code_gen/Code_gen/

```
> x  
[1] 4 2 3 4 6 7 8  
> x>3  
[1] TRUE FALSE FALSE TRUE TRUE TRUE  
[7] TRUE  
> x[x>3]  
[1] 4 4 6 7 8  
> |
```

ประเภทของข้อมูลใน R (ต่อ)

► Matrices

- ข้อมูลสำหรับการวิเคราะห์เชิงสถิติข้อมูลมีหลายตัวแปร และมีลักษณะเป็นตาราง ข้อมูลแต่ละคอลัมน์แทนตัวแปรแต่ละตัว ข้อมูลหลายมิตินี้จะเรียกว่าข้อมูลแบบ matrix หรือเรียกว่า data frame

- matrix ข้อมูลจะต้องมีลักษณะเดียวกันทั้งหมด

Matrices

matrix (data, nrow, ncol, byrow)

Subsetting

- ▶ แสดงค่าบางตัวในเมทริกซ์
 - ▶ ชื่อเมทริกซ์[แถว, คอลัมน์]
 - ▶ ชื่อเวกเตอร์[แถว,)]
 - ▶ ชื่อเวกเตอร์[, คอลัมน์]

```
Console ~1 ↺
>
> mat=matrix(data=c(9,2,3,4,5,6),ncol=3)
> mat
      [,1] [,2] [,3]
[1,]    9    3    5
[2,]    2    4    6
>
> mat=matrix(data=c(9,2,3,4,5,6),nrow=3)
> mat
      [,1] [,2]
[1,]    9    4
[2,]    2    5
[3,]    3    6
> mat=matrix(data=c(9,2,3,4,5,6),nrow=2,ncol=2)
> mat
      [,1] [,2]
[1,]    9    3
[2,]    2    4
> mat[,2]
[1] 3 4
> mat[1,1]
[1] 9
> mat[2,]
[1] 2 4
> mean(mat)
[1] 4.5
>
```

การคำนวณ Matrix ใน R

- ▶ การบวก (Addition)
- ▶ การลบ (Subtraction)
- ▶ อินเวอร์ส (Inverse)
- ▶ ทรานสโพส (Transpose)
- ▶ การคูณ (Multiplication) (หาค่าได้เมื่อเมทริกซ์มีคุณสมบัติที่สามารถคูณกันได้)
- ▶ การคูณด้วยจำนวน

input Matrix

```
mat1=matrix(1:6,2)
```

```
mat1
```

```
mat2=matrix(c(rep(1,3),rep(2,3)), nrow = 2,byrow=T)
```

```
mat2
```

##addition

```
mat1+mat2
```

```
mat2+3
```

##subtraction

```
mat1-mat2
```

##inverse

```
solve(mat1[,2:3])
```

##transpose

t(mat2)

##multiplication

mat1%*%t(mat2)

##element -wise multiplication

mat1*mat2

mat1*4

##division

mat1/mat2

mat2/2

Data frame

- ▶ data frame ข้อมูลเป็นตารางที่ข้อมูลต่างคอลัมน์เป็นข้อมูลต่างชนิดได้ data frame เป็นข้อมูลที่ใช้บ่อยสำหรับการวิเคราะห์ข้อมูล

ตัวอย่างคำสั่งการรวมข้อมูลจาก vector เป็น data frame

- ▶ คำสั่ง data.frame data.frame (เวกเตอร์1, เวกเตอร์2,....., เวกเตอร์ n)

Console ~/

```
> gender = c("male", "female", "female", "male", "female", "female", "male", "male", "female")
> age=c(15, 35, 18, 25, 28, 30, 32, 19, 23)
> pref=c("red", "green", "blue", "yellow", "red", "blue", "green", "blue", "red")
> x=data.frame(gender, age, pref)
> x
```

	gender	age	pref
1	male	15	red
2	female	35	green
3	female	18	blue
4	male	25	yellow
5	female	28	red
6	female	30	blue
7	male	32	green
8	male	19	blue
9	female	23	red

```
>
```

- ▶ ในทางปฏิบัติเราควรเตรียมข้อมูลโดยพิมพ์ไว้ใน Excel ก่อน จากนั้นเก็บไฟล์ในรูปแบบที่สามารถนำเข้า R แล้วใช้คำสั่งในการนำเข้าข้อมูล read.table

Control flow

- ▶ If
- ▶ If...else
- ▶ ifelse(test, yes, no)
- ▶ loop for
- ▶ While-loop
- ▶ repeat

องค์ประกอบของโปรแกรม R



การวิเคราะห์ข้อมูลด้วย RStudio

เรียกใช้ข้อมูลชื่อ data.txt

- ▶ คำสั่งสำหรับเรียกดูสารสนเทศของข้อมูล

▶ **ls**(ชื่อข้อมูล) } ใช้สำหรับเรียกดูชื่อตัวแปรในชุดข้อมูล

▶ **names**(ชื่อข้อมูล)

▶ **str**(ชื่อข้อมูล) ใช้สำหรับการเรียกดูโครงสร้างของชุดข้อมูล เช่น จำนวนตัวแปร
จำนวนค่าสังเกตทั้งหมด และชนิดของตัวแปรแต่ละตัว

▶ **dim**(ชื่อข้อมูล) เรียกดูมิติของ เมทริกซ์

▶ **class**(ชื่อข้อมูล) เรียกดูคุณลักษณะของชุดข้อมูล เช่น numeric, matrix, data
frame เป็นต้น

▶ **levels**(ชื่อข้อมูล) เรียกดูค่าในข้อมูล

Subsetting data

► selecting(Keeping) Variable

```
##select variables read, write, math, science, socst
```

```
myvars=c("read", "write", "math", "science", "socst")
```

```
newdata=mydata[myvars]
```

```
##select 7th to 11th
```

```
newdata <- mydata[c(7:11)]
```

Subsetting data

exclude variables ID, female, race, ses, schtype, prog

```
myvars <- names(mydata) %in% c("ID", "female",  
"race", "race", "ses", "schtype", "prog")
```

```
newdata <- mydata[!myvars]
```

exclude 1rd to 6th variable

```
newdata <- mydata[c(-1:-6)]
```

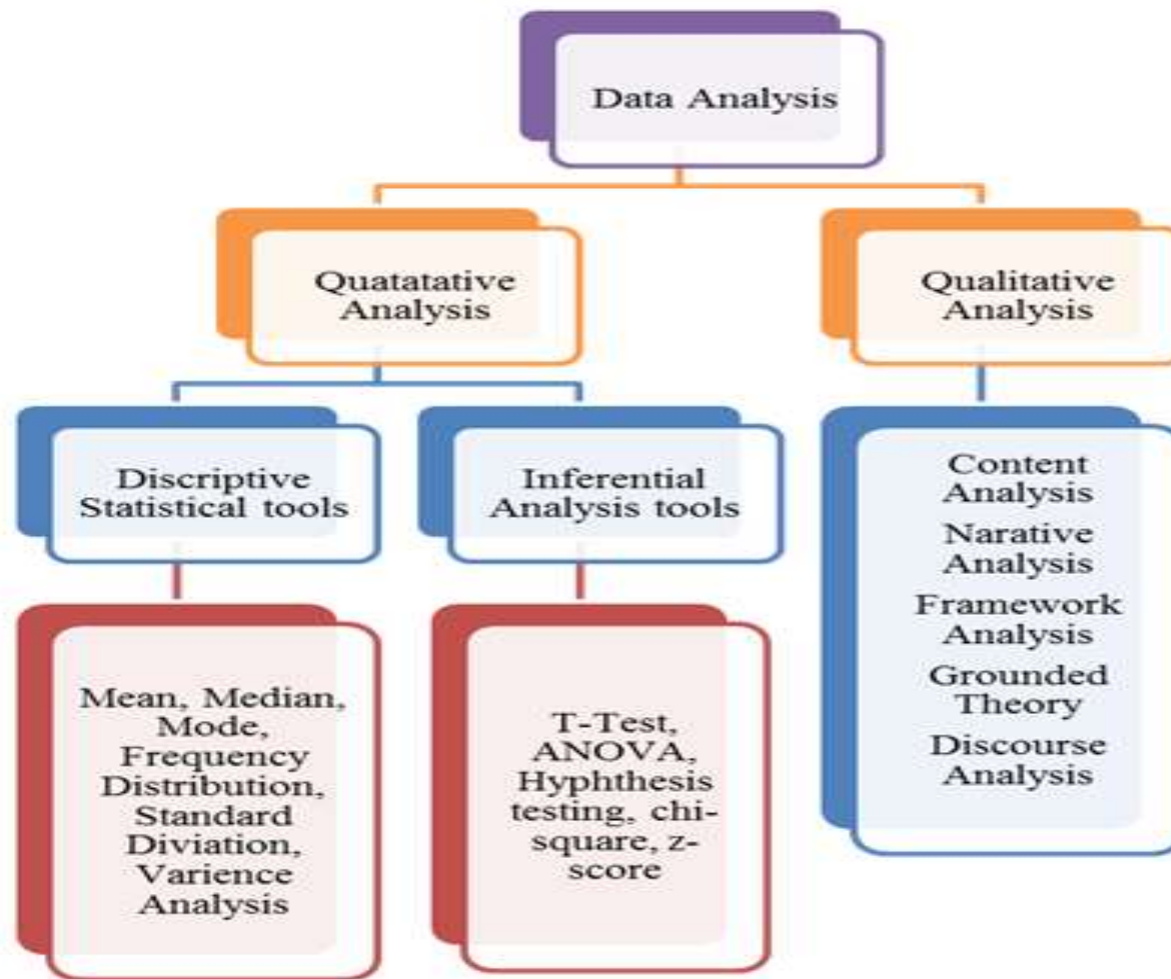
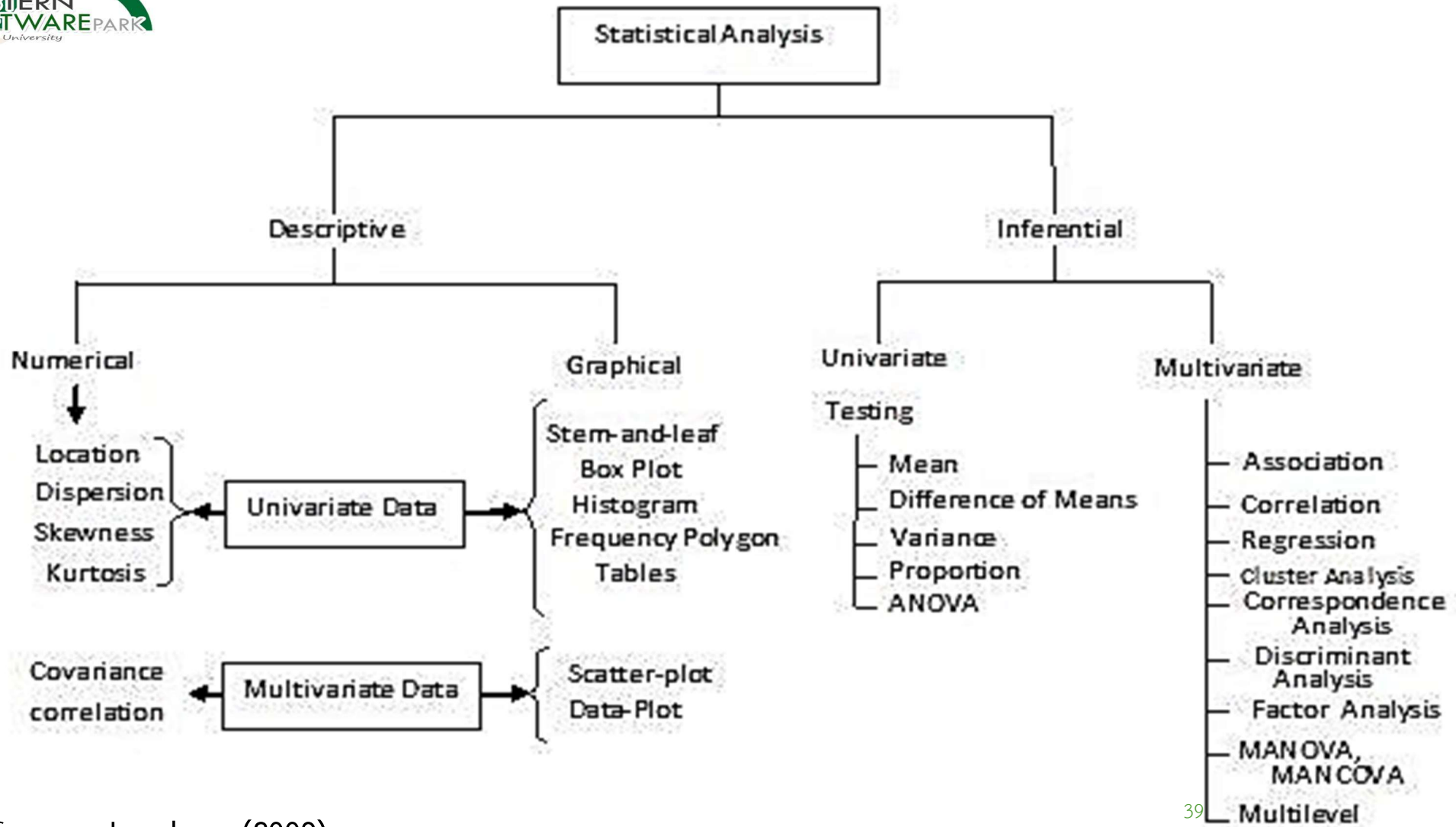


Figure 1 : Clasification of Data Analysis
Source: Saunders et al, (2012)



Source : Jowaheer (2009)

ระดับการวัดของข้อมูล

Scale	Definition	Example
Nominal	Categorical variable, its values cannot be ranked	gender (male, female)
Ordinal	Qualitative variable, its values can be ranked	aggression (weak, moderate, strong intensity)
Interval	The values of the variable can be ranked, and the differences of the values show the distances between the values. This scale does not have a true zero point.	temperature (e.g. $40^{\circ}\text{C} - 20^{\circ}\text{C} = 20^{\circ}\text{C}$, but $40^{\circ}\text{C} / 20^{\circ}\text{C} \neq 2$)
Ratio	The differences of the values show the distances between the values and also the ratio of values is defined, as the variable has a true zero point	height

วิเคราะห์สถิติพื้นฐานจำแนกตามกลุ่ม

```
desg=describeBy (score,mydata$female,mat=TRUE)
```

```
desgF
```

```
desFR=describeBy (score,lisy(mydata$female,mydata$race),mat=TRUE)
```

```
desFR
```

```
write.csv (desgF, file="DesScorbyFemale.csv")
```

```
write.csv(desFR, file="DesScorbyFemaleRace.csv")
```

วิเคราะห์ความถี่และร้อยละของตัวแปรเชิงคุณภาพ

▶ ตัวแปรเดียว

▶ table

`table`(ชื่อข้อมูล\$ตัวแปร)

▶ count

`library(plyr)`

`count`(mydata\$female)

วิเคราะห์ความถี่และร้อยละของตัวแปรเชิงคุณภาพ

▶ 2 way frequency table

▶ table

`table`(ตัวแปรที่ 1, ตัวแปรที่ 2)

▶ addmargins

`a=table`(ตัวแปรที่ 1, ตัวแปรที่ 2)

`addmargins(a)` -----> แสดงผลรวมของตัวแปร

วิเคราะห์ความถี่และร้อยละของตัวแปรเชิงคุณภาพ

▶ 3 way frequency table

▶ table

`table`(ตัวแปรที่ 1, ตัวแปรที่ 2, ตัวแปรที่ 3)

▶ addmargins

`a=table`(ตัวแปรที่ 1, ตัวแปรที่ 2, ตัวแปรที่ 3)

`addmargins(a)` -----> แสดงผลรวมของตัวแปร

วิเคราะห์ความถี่และร้อยละของตัวแปรเชิงคุณภาพ

▶ 2 way Cross tabulation

library(gmodel) -----> gmodel package

crosstable(ชื่อข้อมูล\$ชื่อตัวแปร1, ชื่อข้อมูล\$ชื่อตัวแปร2)

การคำนวณสถิติพื้นฐานสำหรับตัวแปรเชิงปริมาณ

- ▶ ค่าต่ำสุด (Minimize) >min(ชื่อข้อมูล\$ชื่อตัวแปร)
- ▶ ค่าสูงสุด (Maximize) >Max(ชื่อข้อมูล\$ชื่อตัวแปร)
- ▶ หาค่าเฉลี่ย (Mean) >mean(ชื่อข้อมูล\$ชื่อตัวแปร)
- ▶ หาค่าค่ามัธยฐาน (Median) >median(ชื่อข้อมูล\$ชื่อตัวแปร)
- ▶ ส่วนเบี่ยงเบนมาตรฐาน (Standardization) >sd(ชื่อข้อมูล\$ชื่อตัวแปร)
- ▶ ความแปรปรวน (Variance) >var(ชื่อข้อมูล\$ชื่อตัวแปร)
- ▶ พิสัย(Range) >range(ชื่อข้อมูล\$ชื่อตัวแปร)

คำนวณสถิติพื้นฐาน R

- ▶ กรณีต้องการคำนวณค่าสถิติพื้นฐานพร้อมกันหลายตัวแปร
 - ▶ **summary** (ชื่อข้อมูล)
 - ▶ **describe** (ชื่อข้อมูล) ภายใต้อัปเดต **psych** package เรียก **library (psych)**

คำสั่ง summary

```
mydata =read.table ("data.txt", header=TRUE)
```

```
score=mydata[c(7:11)]
```

```
sum=summary(score)
```

```
sum
```

```
write.csv(sum,file="SumScore.csv")
```

คำสั่ง describe

```
library(psych)
```

```
des=describe(score)
```

```
des
```

```
write.csv(des, file="DesScor.csv")
```

Bar plots เรียกข้อมูลชื่อ mtcars

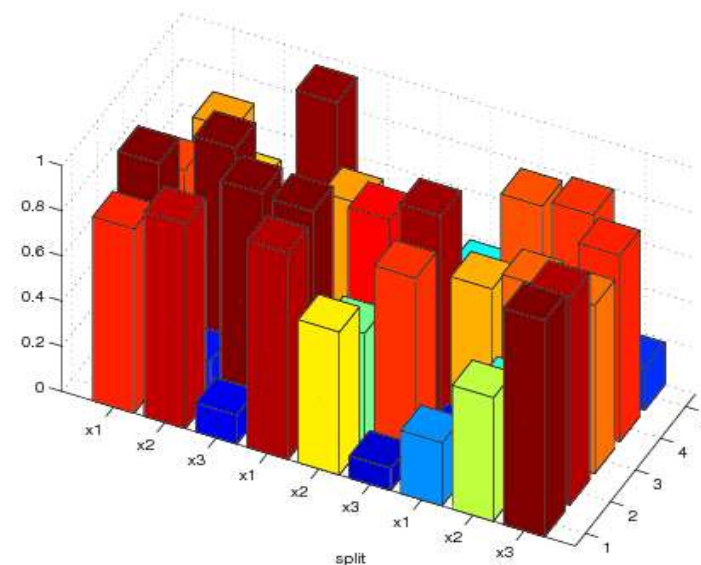
► เรียกข้อมูลชื่อ **mtcars**

► Simple Bar plot

```
counts <- table(mtcars$gear)
```

```
barplot(counts, main="Car Distribution",
```

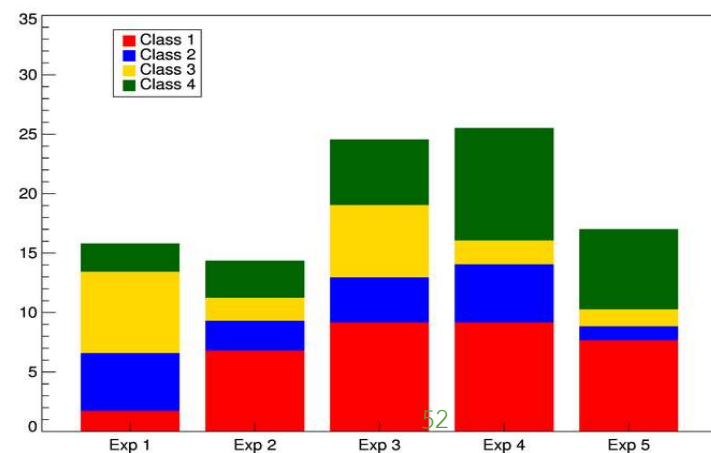
```
  xlab="Number of Gears")
```



Stracked Bar Plot

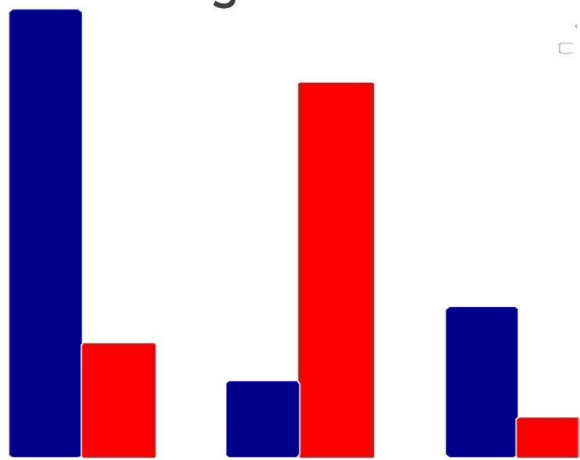
```
► counts <- table(mtcars$vs, mtcars$gear)

barplot(counts, main="Car Distribution by Gears and VS",
        xlab="Number of Gears", col=c("darkblue","red"),
        legend = rownames(counts))
```



Grouped Bar Plot

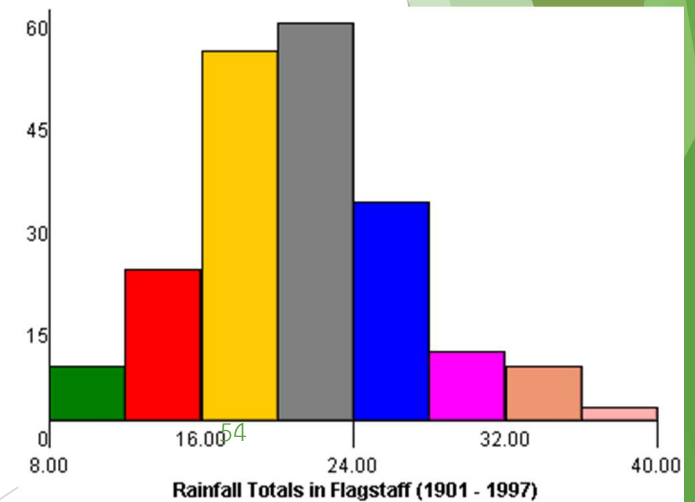
```
► counts <- table(mtcars$vs, mtcars$gear)
barplot(counts, main="Car Distribution by Gears and VS",
xlab="Number of Gears", col=c("darkblue","red"),
legend = rownames(counts), beside=TRUE)
```



การสร้างกราฟในโปรแกรม R

► Histogram

hist(ชื่อข้อมูล\$ชื่อตัวแปร, col="สี", xlab="ชื่อแกน x",
ylab = "ชื่อแกน y", main="ชื่อกราฟ")



Histogram with Normal Curve

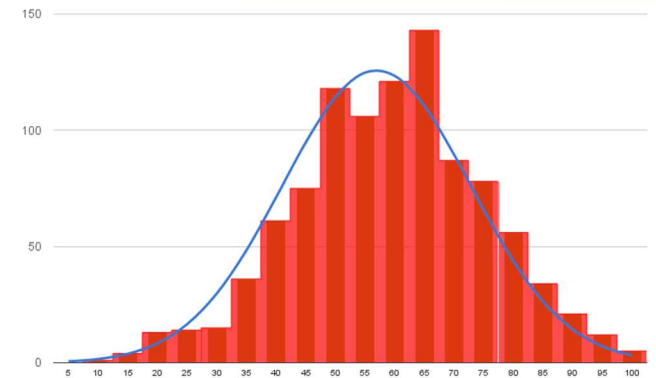
```
x <- mydata$read
```

```
m=mean(mydata$read)
```

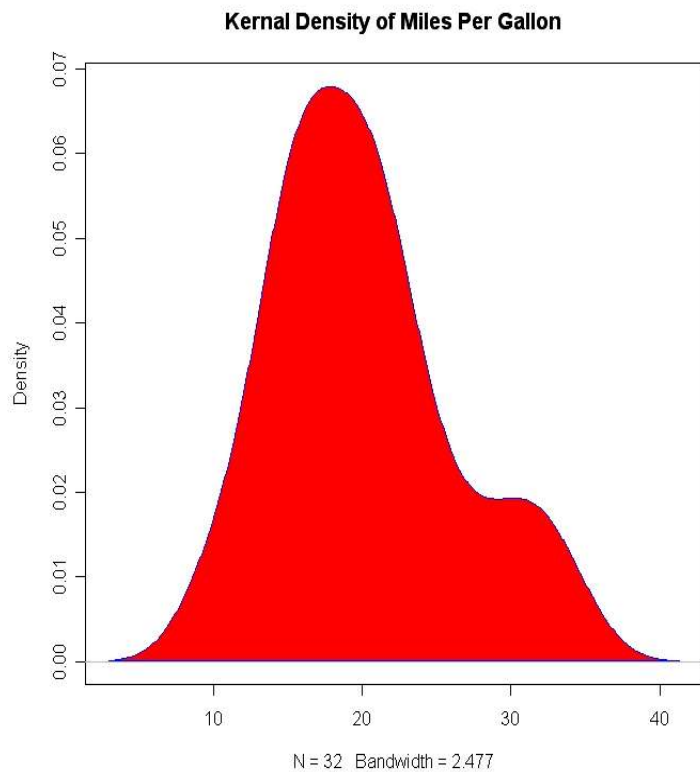
```
std=sd(mydata$read)
```

```
hist(x, col="red", prob=TRUE, xlab="Score", main="Histogram with Normal  
Curve")
```

```
curve(dnorm(x, mean=m, sd=std), col="darkblue", lwd=2, add=TRUE)
```



Kernel Density Plots



► `density`(ชื่อข้อมูล\$ชื่อตัวแปร)

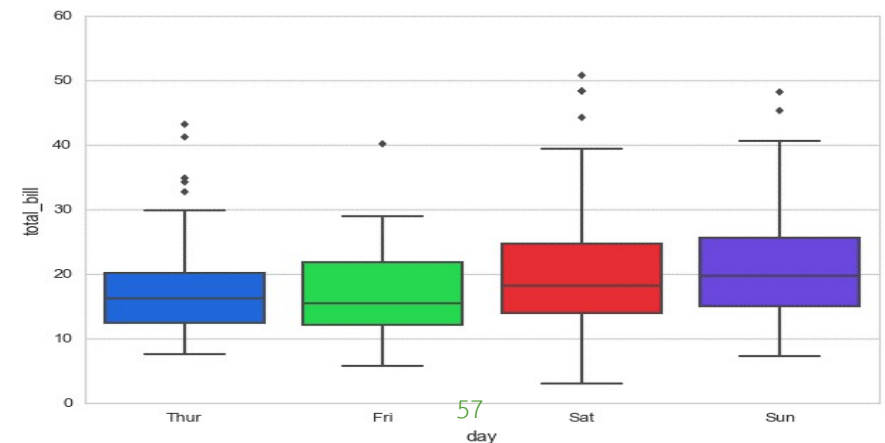
► `d <- density(mydata$write)`

► `Plot(d, xlab=".....",
main=".....")`

► `polygon(d, col="red", border="blue")`

Boxplot

- ▶ `boxplot(ชื่อข้อมูล$ชื่อตัวแปร1, ชื่อข้อมูล$ชื่อตัวแปร2, names=c("ชื่อกราฟแต่ละกล่อง",.....),ylab=".....", xlab=".....",main=".....")`
- ▶ `boxplot(mydata$read,mydata$write,mydata$math, mydata$science, mydata$socst, names=c("read", "write", "math", "science", "scost"),xlab="SIPA SCORE")`



Scatterplots

► Simple Scatterplot

```
attach(mtcars)
```

```
plot(wt, mpg, main="Scatterplot Example",  
     xlab="Car Weight ", ylab="Miles Per Gallon ", pch=19)
```

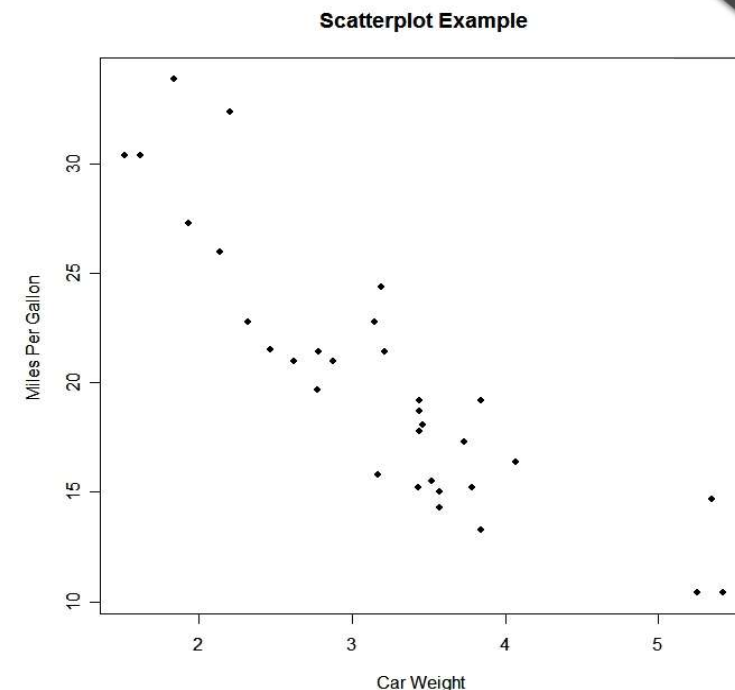
```
# Add fit lines
```

```
abline(lm(mpg~wt), col="red")
```

```
lines(lowess(wt,mpg), col="blue")
```

```
# regression line (y~x)
```

```
# lowess line (x,y)
```



Scatterplot Matrices

Basic Scatterplot Matrix

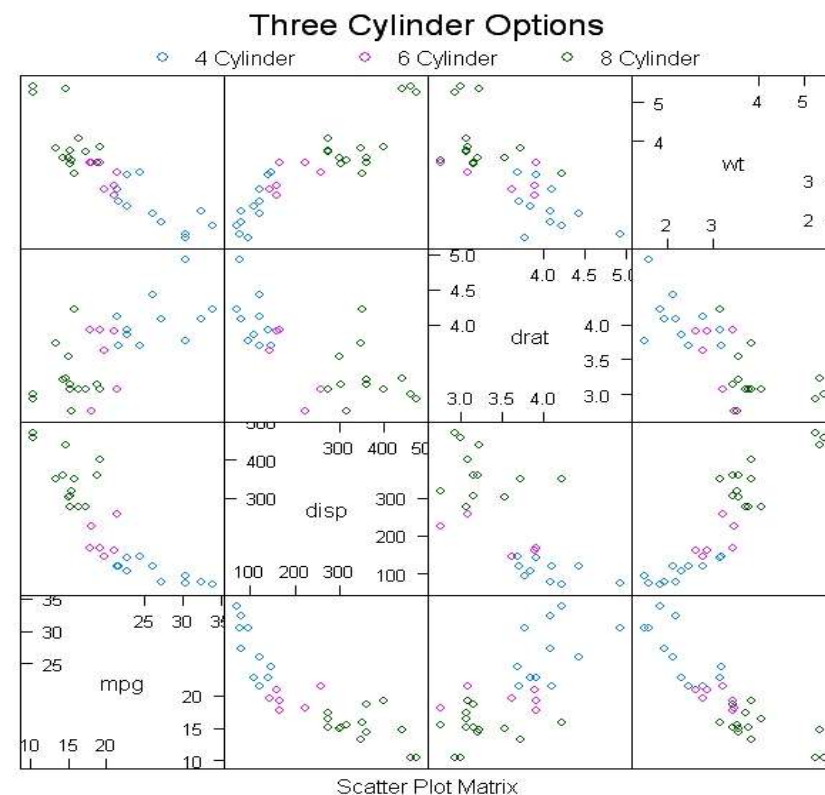
```
pairs(~mpg+disp+drat+wt,data=mtcars,  
      main="Simple Scatterplot Matrix")
```

Scatterplot Matrices from the car Package

```
library(car)
```

```
scatterplot.matrix(~mpg+disp+drat+wt|cyl,
```

```
data= mtcars,main="Three Cylinder Options")
```



High Density Scatterplots

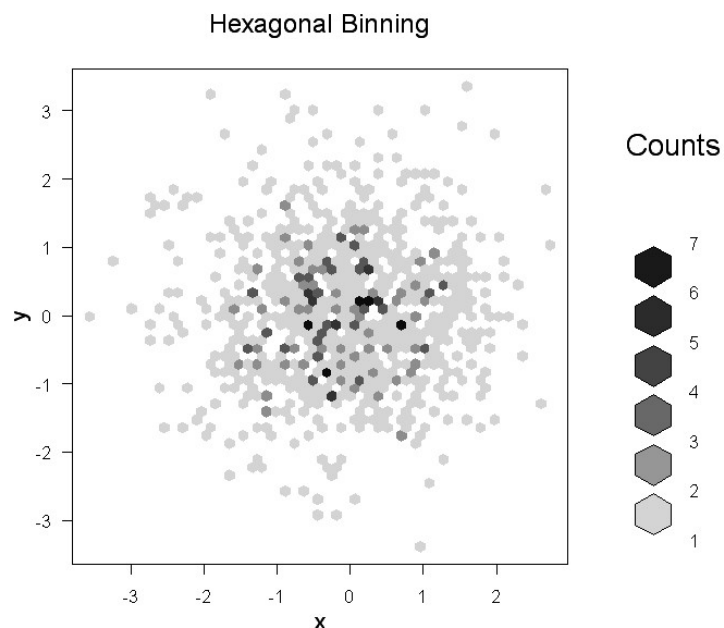
► `library(hexbin)`

`x <- rnorm(1000)`

`y <- rnorm(1000)`

`bin <- hexbin(x, y, xbins=50)`

`plot(bin, main="Hexagonal Binning")`



3D Scatterplots

► `library(scatterplot3d)`

`attach(mtcars)`

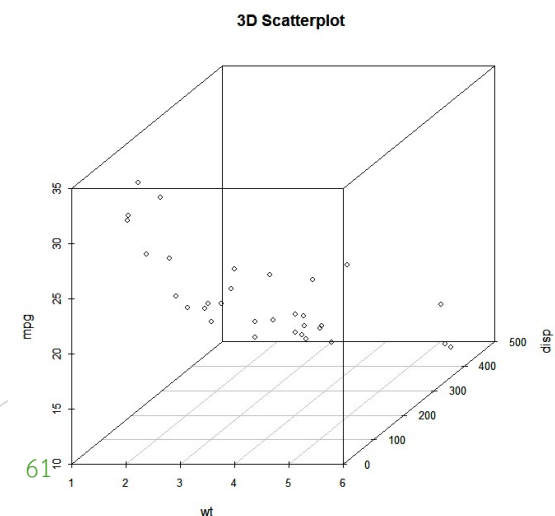
`scatterplot3d(wt, disp, mpg, main="3D Scatterplot")`

or

`scatterplot3d(wt, disp, mpg, pch=16,`

`highlight.3d=TRUE,`

`type="h", main="3D Scatterplot")`



Spinning 3D Scatterplots

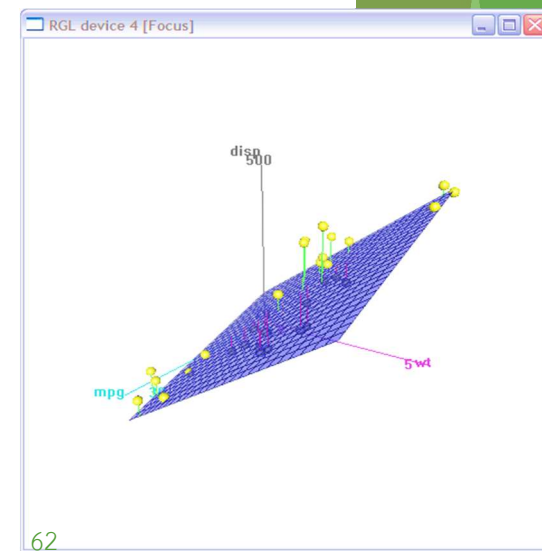
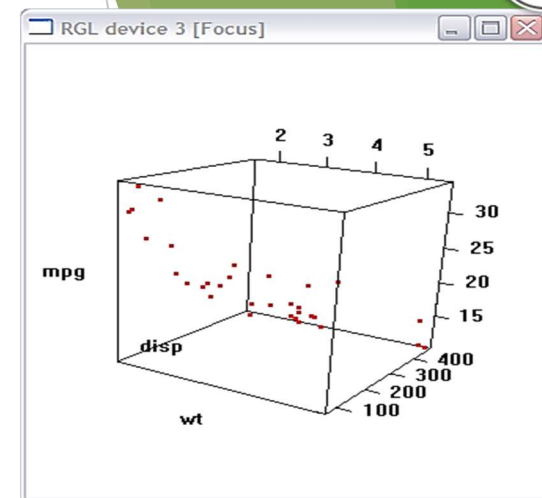
► `library(rgl)`

`plot3d(wt, disp, mpg, col="red", size=3)`

► `library(Rcmdr)`

`attach(mtcars)`

`scatter3d(wt, disp, mpg)`



การคำนวณ Z-score

- ▶ $Z = \frac{x - \bar{x}}{\sigma}$ เป็นการวิธีปรับค่าตัวเลข โดยที่ค่า Z จะบอกถึง probability ที่ตำแหน่งนั้นได้ เช่น
- ▶ เก็บข้อมูลความสูงมีค่าเฉลี่ยเป็น 170 และค่า sd เป็น 8 หากต้องการรู้ว่าจำนวนคนที่มีความสูงน้อยกว่า 160 คำนวณได้จาก $z = (160 - 170) / 8 = -1.25$ ซึ่ง $\text{pnorm}(-1.25)$ ได้ค่า 0.1055498 หมายถึงมี 10%

Z-score

- ▶ คำสั่งแปลงข้อมูลเป็นค่า Z-score
 - ▶ **Scale**(ชื่อชุดข้อมูล\$ชื่อตัวแปร)
- ▶ คำวนวนค่า ความน่าจะเป็นของค่า Z
 - ▶ **pnorm**(ชื่อชุดข้อมูล\$ชื่อตัวแปร)

Test Normality

H_0 : specific distribution

H_1 : not specific distribution

- ▶ Anderson darling
- ▶ histogram
- ▶ Shapiro wilk normality test
- ▶ qqplot
- ▶ Kolmogorov smirnov test

library(nortest)

- ▶ Anderson darling

ad.test(ชื่อตัวแปร)

- ▶ Shapiro wilk normality test

shapilo.test(ชื่อตัวแปร)

- ▶ Kolmogorov smirnov test

lillie.test(ชื่อตัวแปร)

- ▶ Histogram

hist(ชื่อตัวแปร)

- ▶ qqplot

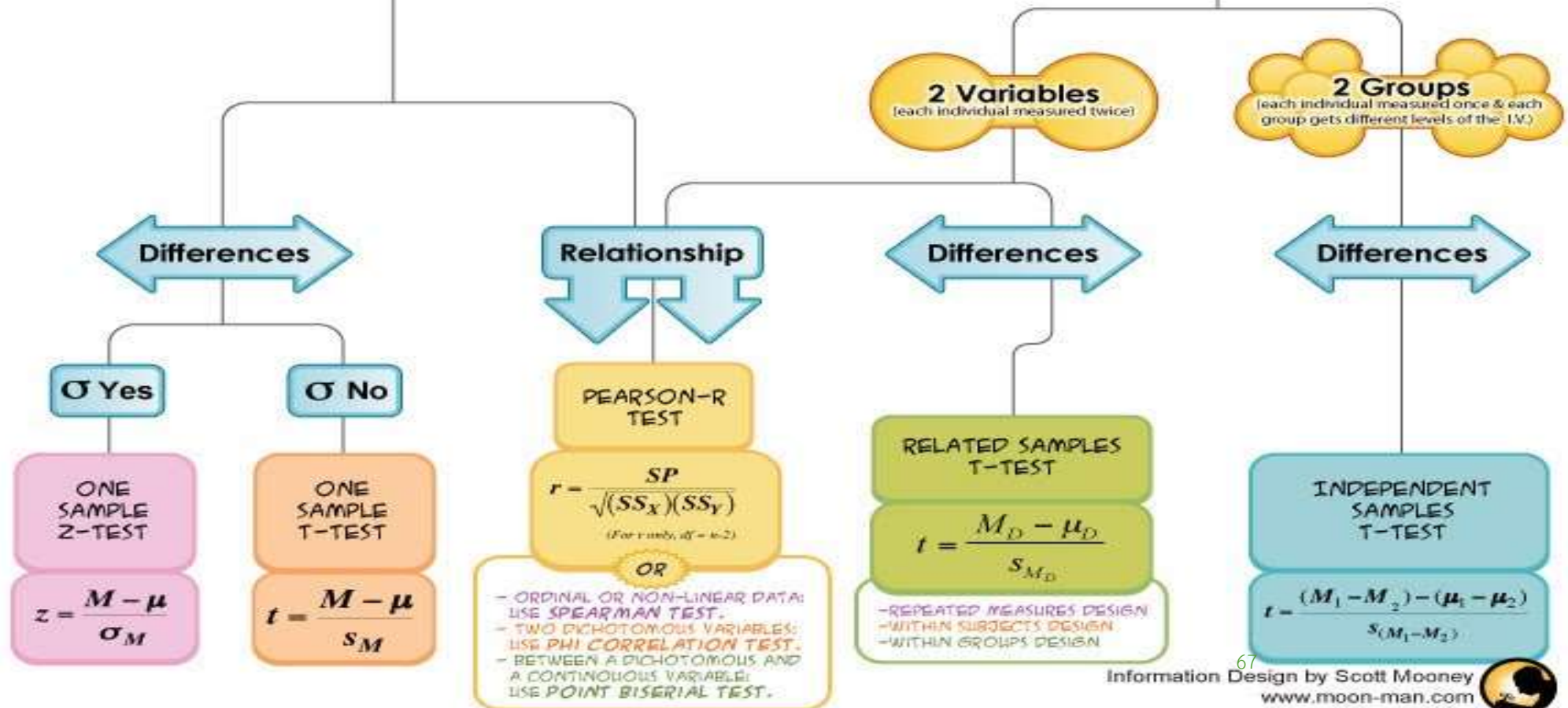
- ▶ qqnorm(ชื่อตัวแปร)



How to choose your Hypothesis Test

1 Sample

2 Samples



t-test

- ▶ t-test เป็นสถิติพื้นฐานตัวหนึ่งที่ใช้กันอย่างแพร่หลาย
- ▶ เปรียบเทียบค่าเฉลี่ยของกลุ่มตัวอย่างกับค่าเฉลี่ยประชากรว่าค่าเฉลี่ยที่ได้มาจากประชากรนี้หรือไม่
- ▶ เปรียบเทียบค่าเฉลี่ยของกลุ่มตัวอย่างว่ามาจากประชากรเดียวกันหรือไม่
- ▶ t-test ใช้กับตัวอย่างน้อยกว่า 30

t-test

▶ `t.test`(ชื่อข้อมูล\$ชื่อตัวแปร,
 `mu` = ค่าเฉลี่ยประชากร,
 `alternative` = "two.sided", "less", "greater",
 `paired` = FALSE, `var.equal` = FALSE,
 `conf.level` = 0.95)

t-test

t.test (mydata\$read, mu=50)

One Sample t-test data: mydata\$read

$t = 3.0759$, $df = 199$, $p\text{-value} = 0.002394$

alternative hypothesis: true mean is not equal to 50

95 percent confidence interval: 50.80035 53.65965

sample estimates: mean of x

52.23

t-test

- ▶ `t.test` (mydata\$read, mu=50)
- ▶ `t.test` (mydata\$read, mu=50, alternative="greater")

One Sample t-test data:

mydata\$read t = 3.0759, df = 199, p-value = 0.001197

alternative hypothesis: true mean is greater than 50

95 percent confidence interval: 51.03192 Inf

sample estimates: mean of x 52.23

t-test

- ▶ t-test เหมาะกับข้อมูลที่มีการกระจายตัวแบบ normal distribution
- ▶ ถ้า มีการกระจายตัวแบบ skewness distribution ใช้ Wilcoxon test

▶ **wilcox.test** (ชื่อข้อมูล\$ชื่อตัวแปร,
mu= ค่าเฉลี่ยประชากร,
alternative = "two.sided", "less", "greater",
paired = FALSE, conf.level = 0.95)

Two sample t-test

independent two group

where y is numeric and x is a binary factor

```
t.test(y~x)
```

where y1 and y2 are numeric

```
t.test(y,x)
```

Mann-Whitney U Test

#where y is numeric and A is A binary factor

```
wilcox.test(y~A)
```

where y and x are numeric

```
wilcox.test(y,x)
```

Paired t-test

► `t.test`(ตัวแปรที่1, ตัวแปรที่ 2, paired=TRUE)

Paired t-test data: mydata\$read and mydata\$write

t = -0.86731, df = 199, p-value = 0.8066

alternative hypothesis: true difference in means is greater than 0

95 percent confidence interval: -1.583431 Inf

sample estimates: mean of the differences

-0.545

Wilcoxon Signed Rank Test

where y1 and y2 are numeric

▶ **wilcox.test** (y1,y2,paired=TRUE)

การคำนวณค่า Chisquare

- ▶ เป็นสถิติแบบนอนพาราเมตริกซ์ ใช้กับข้อมูลที่เป็น categories และเป็นจำนวนนับความถี่
- ▶ คำสั่ง chisquare test คือการเปรียบเทียบว่าค่าที่พบ(observe)นี้แตกต่างจากค่าที่ควรจะเป็น (Expected) อย่างผิดปกติหรือไม่
- ▶ Null hypothesis ค่าที่พบนี้ไม่แตกต่างไปจากค่าที่ควรจะเป็น หรือตัวแปรทั้งสองมีความสัมพันธ์กัน

Chisquare test

- ▶ `chisq.test`(ชื่อข้อมูลที่อยู่ในรูปตารางแจกแจงความถี่)
- ▶ คำสั่งจัดรูปแบบข้อมูลในรูปตาราง
 - ▶ `Xtabs(~ตัวแปรที่1+ตัวแปรที่2,data=.....)`

การคำนวณ ANOVA

- ▶ เปรียบเทียบค่าเฉลี่ยประชากรตั้งแต่ 2 กลุ่มขึ้นไป
- ▶ ควบคุมความคลาดเคลื่อนในการทดสอบ
 - ▶ one-way anova : ตัวแปรอิสระ 1 ตัว
 - ▶ two-way anova : ตัวแปรอิสระ 2 ตัว
 - ▶ three-way anova : ตัวแปรอิสระ 3 ตัว

ANOVA Analysis

▶ สมมติฐาน

▶ null hypothesis : ค่าเฉลี่ยของแต่ละกลุ่มมีค่าเท่ากัน

▶ alternative hypothesis: มีค่าเฉลี่ยอย่างน้อยหนึ่งคู่ไม่เท่ากัน

oneway.test (ตัวแปรเชิงปริมาณ ~ ตัวแปรเชิงกลุ่ม, data=mydata, var.equal = TRUE)

***ถ้า reject H0 ทำการเปรียบเทียบรายคู่

การเปรียบเทียบรายคู่

LSD (Least Significant Difference)

Duncan's multiple-range test

Newman-keuls test

Tukey HSD (Honestly Significant Difference)

Sheffe's test

More
powerful

More
conservative

ที่มา: O'Mahony, 1986

LSD

`pairwise.t.test(ตัวแปรปริมาณ, ตัวแปรคุณภาพ, p.adjust="none")`

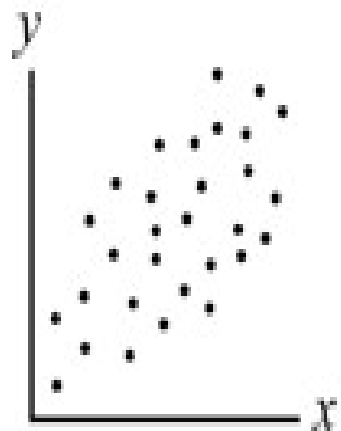
Bonferroni

`pairwise.t.test(ตัวแปรปริมาณ, ตัวแปรคุณภาพ, p.adjust="bonf")`

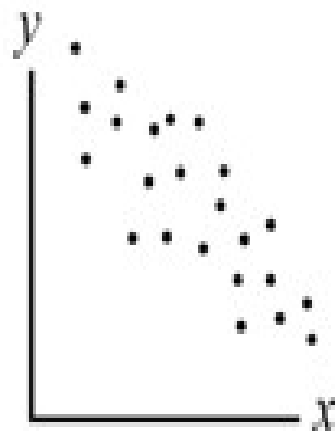
Tukey HSD

▶ TukeyHSD(aov(ตัวแปรปริมาณ,ตัวแปรคุณภาพ,data=.....))

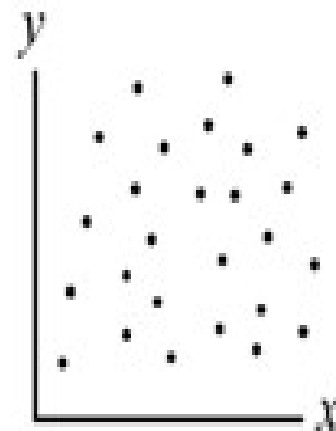
Linear correlation



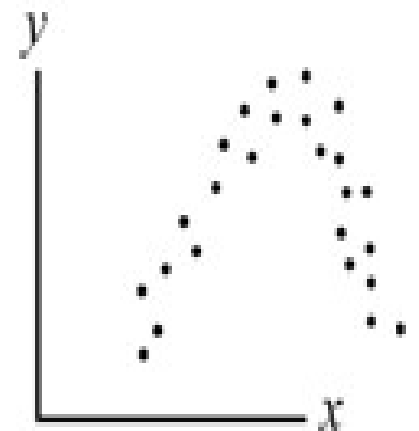
positive



negative



no correlation



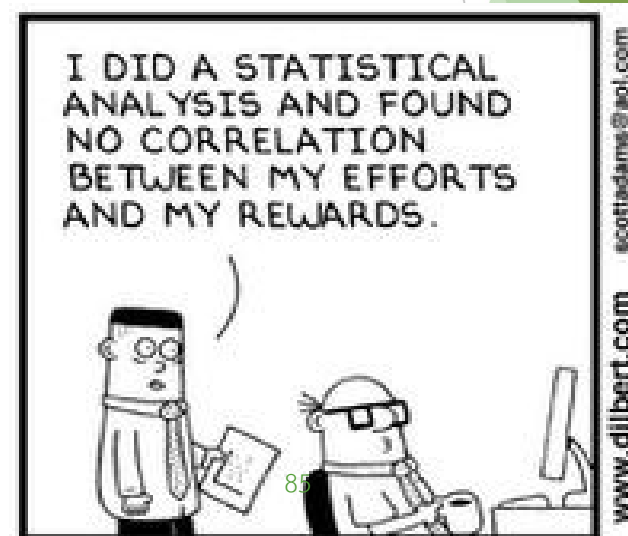
non linear

Correlation method

Correlation Coefficient
Shows Strength & Direction of Correlation



- ▶ Pearson r correlation
- ▶ Kendall Rank correlation
- ▶ Spearman rank correlation



Correlation

- ▶ Correlation เป็นการดูความสัมพันธ์ของตัวแปรเชิงปริมาณเชิงเส้นตรงของตัวแปรสองตัว
- ▶ `cor(mtcars, use="complete.obs", method="pearson", "kendall", "spearman")`



A BUNCH OF DATA



Regression



Assumptions of multiple regression

- ▶ Linear relationship.
- ▶ Multivariate normality.
- ▶ No or little multicollinearity.
- ▶ No auto-correlation.
- ▶ Homoscedasticity.
- ▶ correlation
- ▶ Shapiro test, mardia MVN test
- ▶ VIF
- ▶ Durbin Watson test
- ▶ Bartlett test

การวิเคราะห์การถดถอย Regression

- ▶ `fit=lm(ตัวแปรตาม(ปริมาณ)~ตัวแปรอิสระ1+....., data=.....)`
- ▶ `summary (fit)`
- ▶ `reduced.model=step(fit,direction="both","backward", "forward")`

outlier

▶ **outlierTest** (fit)

▶ **qqPlot**(fit, main="QQ Plot")

#qq plot for studentized residual

▶ **leveragePlots**(fit)

Multicollinearity (VIF)

► # Evaluate Collinearity

vif(fit) # variance inflation factors

sqrt(vif(fit)) > 2 # problem?

Non-independence of Errors

▶ # Test for Autocorrelated Errors

durbinWatsonTest (fit)

THANK
YOU