



การใช้ซอฟต์แวร์ฟรี R ในการวิเคราะห์สถิติเบื้องต้น สำหรับอุตสาหกรรมอาหาร ส่วนที่ 1 พื้นฐาน

ผู้ช่วยศาสตราจารย์ ดร.ปราโมทย์ คูวิจิตรจารุ

ภาควิชาเทคโนโลยีอาหาร คณะวิศวกรรมศาสตร์และเทคโนโลยีอุตสาหกรรม

มหาวิทยาลัยศิลปากร

Khuwijitjaru_p@su.ac.th



- R (อาร์) เป็นซอฟต์แวร์สำหรับการคำนวณทางสถิติที่เป็นซอฟต์แวร์ฟรี (Free of charge) และเป็นซอฟต์แวร์แบบโอเพ่นซอร์ส (Open Source)
- พัฒนามาจากภาษา S (เอส) โดย Robert Gentleman และ Ross Ihaka, นักสถิติจาก University of Auckland ในนิวซีแลนด์
- S ถูกพัฒนาไปเป็นซอฟต์แวร์ทางการค้าชื่อ S-PLUS
- ปัจจุบัน R อยู่ภายใต้การดูแลของ R Development Core Team



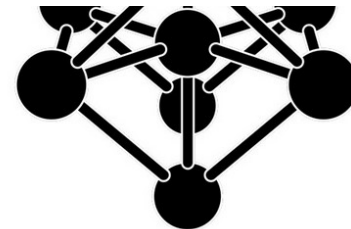
ข้อดีของ R

- ฟรี และมีการพัฒนาอยู่ตลอดเวลา มีการเปิดให้ดาวโหลดผ่านเซิร์ฟเวอร์ในหลากหลายประเทศ
- พัฒนาโดยนักสถิติและมีชุมชนออนไลน์ (ภาษาอังกฤษ) ที่เข้มแข็ง
- ได้รับการยอมรับทั่วโลก มีการอบรมและสอนในสถาบันต่างๆ อย่างกว้างขวาง
- มีแพ็คเกจเพื่อวิเคราะห์หลากหลายสำหรับการวิเคราะห์ข้อมูลขั้นสูงต่างๆ
- สร้างกราฟคุณภาพสูงสำหรับการตีพิมพ์
- การใช้งานแบบคำสั่ง สามารถเก็บคำสั่งไว้ในรูป script เพื่อนำมาวิเคราะห์ซ้ำได้ง่าย





useR!2017 BRUSSELS

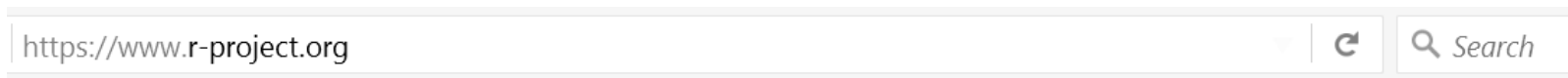
[NEWS](#)[ABOUT](#)[REGISTRATION](#)[PROGRAM](#)[VENUE](#)[FAQ](#)[EXTRA](#)

Annual useR! conference



การดาวน์โหลดและติดตั้ง R บนวินโดวส์

- ดาวโหลดโปรแกรมติดตั้งจาก www.r-project.org



[\[Home\]](#)

Download

[CRAN](#)

R Project

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Reporting Bugs](#)

[Development Site](#)

[Conferences](#)

[Search](#)

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It covers a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please check the [mirror](#).

If you have questions about R like how to download and install the software, or how to use it, please read our [answers to frequently asked questions](#) before you send us an email.

News

- **useR! 2017** (July 4 - 7 in Brussels) has opened registration and more at [http://www.useR2017.org](#)
- Tomas Kalibera has joined the R core team.
- The R Foundation welcomes five new ordinary members: Jennifer Bryan, Tomas Kalibera, and Balasubramanian Narasimhan



ขั้นตอนการดาวโหลด

- คลิกที่ CRAN และเลือกเซิร์ฟเวอร์ เช่น 0-Cloud
- คลิก Download R for Windows
- คลิก base
- คลิก Download R x.x.x for Windows

R-3.3.2 for Windows (32/64 bit)

[Download R 3.3.2 for Windows](#) (62 megabytes, 32/64 bit)

[Installation and other instructions](#)

[New features in this version](#)

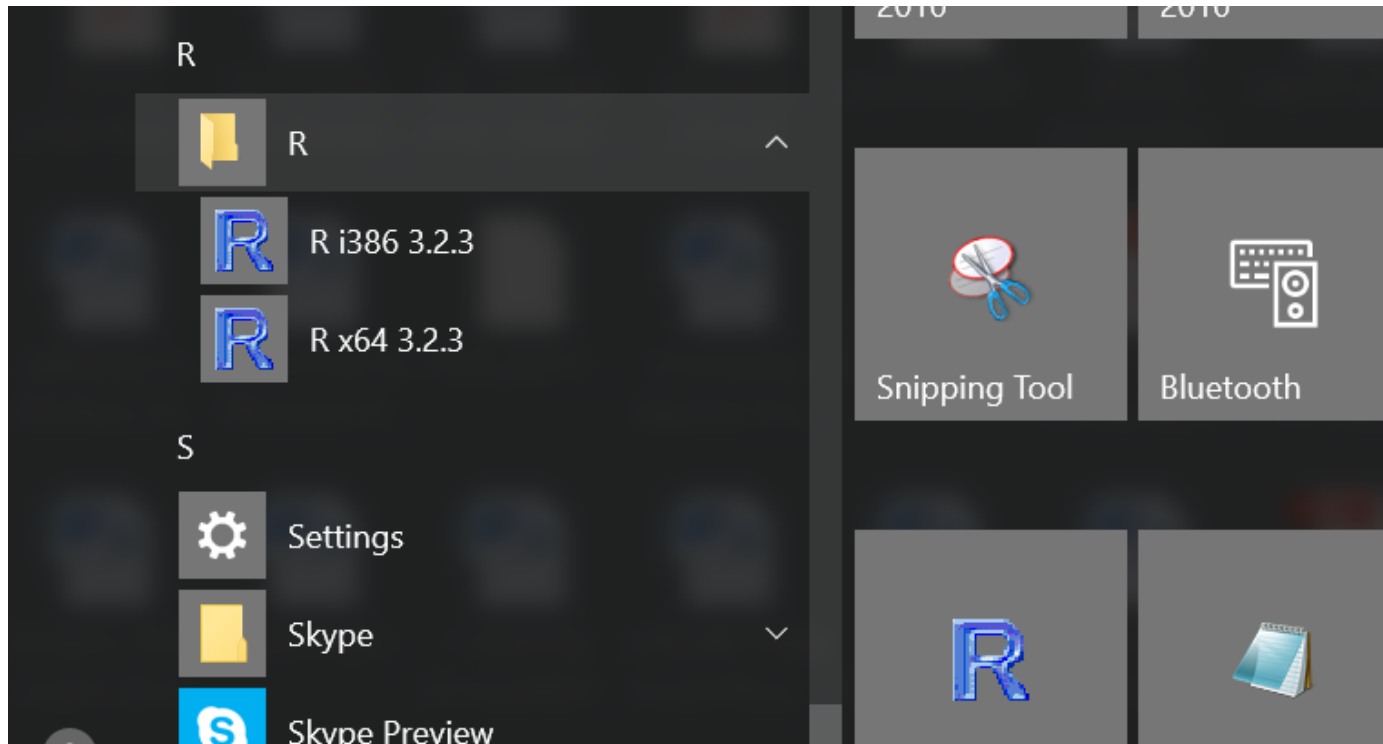
If you want to double-check that the package you have downloaded exactly matches the package distributed by R to the [true fingerprint](#). You will need a version of md5sum for windows: both [graphical](#) and [command line version](#)

Frequently asked questions



ขั้นตอนการติดตั้ง

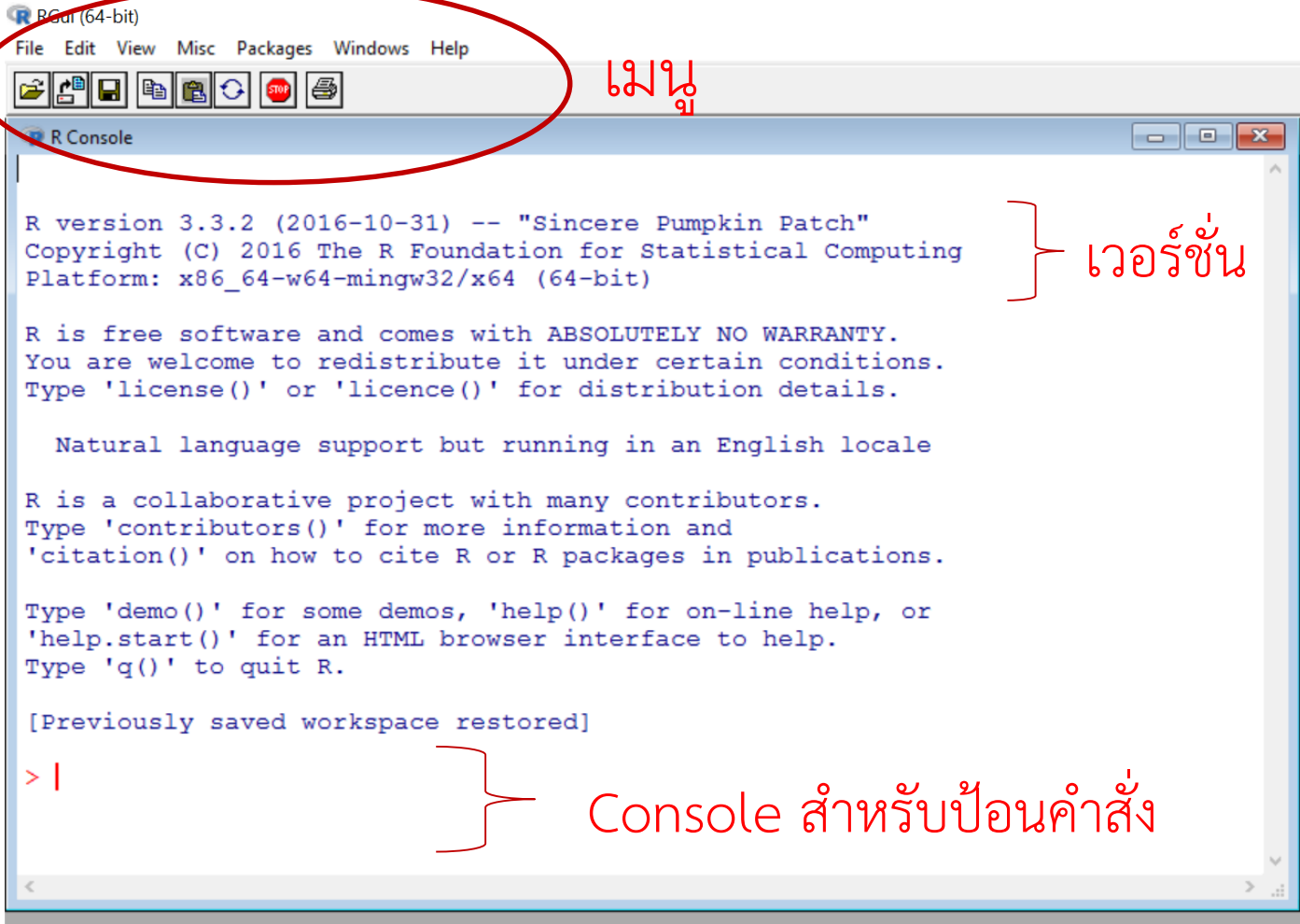
- ดับเบิลคลิกที่ ไฟล์ที่ดาวโหลดมา
- คลิก Next ทุกขั้นตอน
- เมื่อติดตั้งเสร็จจะเห็นไอคอนโปรแกรมบนเดสทอป หรือที่ Start menu





เริ่มการใช้งาน R

- คลิกไอคอนโปรแกรม R จะปรากฏ RGui ซึ่งเป็น Graphical User Interface อย่างง่ายของ R ซึ่งมีเมนูบางอย่างและ R Console





การใช้ R Console

- การใช้งาน R จะต้องพิมพ์คำสั่งหรือข้อมูลลงไปบนหน้า R Console นี้ เช่น พิมพ์ $5 + 7$ หลังเครื่องหมาย “>” แล้วเคาะ Enter

> 5+7

[1] 12

>

- R จะแสดงผลลัพธ์ของการบวกนั้นคือ 12
- [1] เป็นเลขแสดงตำแหน่งของข้อมูลตัวแรกของแถว



การใช้ R เป็นเครื่องคิดเลข

ทดลองคำนวณต่อไปนี้

- $167 - 82$
- $50 * 7$ (* คือการคูณ)
- $1000/20$ (/ คือการหาร)
- $2^{**}20$ (** คือการยกกำลัง)
- $\log(100)$ (log คือฟังก์ชันหาค่า natural logarithm)
- $\log_{10}(100)$ (log10 คือฟังก์ชันการหาค่า logarithm ฐาน 10)
- $\text{sqrt}(2)$ (sqrt คือฟังก์ชันการหาค่ารากที่ 2)



ฟังก์ชัน (function)

- ฟังก์ชันคือคำสั่งที่ใช้สำหรับการคำนวณหรือแสดงผลบางอย่าง
- R มีฟังก์ชันสำหรับการคำนวณที่ใช้บ่อยๆ มากมาย เช่น `log10()` คือฟังก์ชันในการคำนวณค่า \log ฐาน 10 หรือ `sqrt()` คือการหาค่ารากที่ 2
- ฟังก์ชันเหล่านี้เป็นฟังก์ชันในแพ็คเกจ base
- เราใช้ คำสั่ง ? (เครื่องหมายคำถาม) ตามด้วยชื่อฟังก์ชันเพื่อดูรายละเอียดของฟังก์ชันนั้นๆ เช่น `?log`



ตัวอย่างหน้า help ของฟังก์ชัน log()

`log {base}`

R Documentation

Logarithms and Exponentials

Description

`log` computes logarithms, by default natural logarithms, `log10` computes common (i.e., base 10) logarithms, and `log2` computes binary (i.e., base 2) logarithms. The general form `log(x, base)` computes logarithms with base `base`.

`log1p(x)` computes $\log(1+x)$ accurately also for $|x| \ll 1$.

`exp` computes the exponential function.

`expm1(x)` computes $\exp(x) - 1$ accurately also for $|x| \ll 1$.

Usage

```
log(x, base = exp(1))  
logb(x, base = exp(1))  
log10(x)  
log2(x)
```

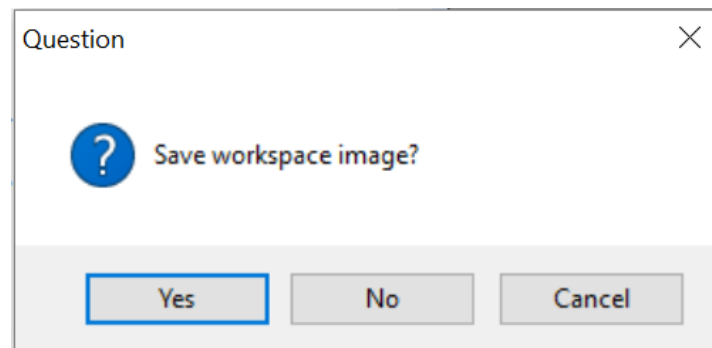
```
log1p(x)
```

```
exp(x)  
expm1(x)
```




การเซฟข้อมูลและการปิดโปรแกรม

- ปิดโปรแกรม RGui ด้วยการคลิกเครื่องหมายปิดหน้าต่างหรือพิมพ์ q() ใน console
- R จะถามว่าต้องการเซฟข้อมูล (workspace image) หรือไม่



- หากตอบ Yes โปรแกรมจะเซฟคำสั่งและข้อมูลต่างๆ ที่เคยพิมพ์ไว้
- สามารถใช้ปุ่มลูกศรขึ้นและลงเลื่อนหาคำสั่งที่เคยพิมพ์ไปแล้วได้
- Ctrl + L เพื่อเคลียร์หน้า console



ตัวแปร (variable)

- เราสามารถเก็บค่าใน “ตัวแปร”

```
> a <- 2
```

```
> b <- 3
```

```
> a*b
```

```
[1] 6
```

```
> c <- a*b
```

```
> c
```

```
[1] 6
```



การใช้คำสั่ง c

```
> x <- c(1, 2, 3) # สร้างตัวแปร x ที่เก็บค่าตัวเลข 1, 2, และ 3
```

```
> x
```

```
[1] 1 2 3
```

เรียกชุดของข้อมูลที่ประกอบด้วยหลายๆ จำนวนแบบนี้ว่า “เว็คเตอร์” (vector)

```
> x[1]
```

```
[1] 1
```

```
> x[2]
```

```
[1] 2
```

```
> x[3]=4
```

```
> x
```

```
[1] 1 2 4
```



การใช้คำสั่ง c

```
> name <- c("a", "b", "c")
```

```
> name
```

```
[1] "a" "b" "c"
```

```
> name[1]
```

```
[1] "a"
```

```
> treatment <- c("sucrose", "fructose", "glucose")
```

```
> treatment
```

```
[1] "sucrose" "fructose" "glucose"
```

```
> treatment <- rep(c("a", "b", "c"), each=3)
```

```
> treatment
```

```
[1] "a" "a" "a" "b" "b" "b" "c" "c" "c"
```



```
> number<-1:100
```

```
> number
```

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
```

```
[19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
```

```
[37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
```

```
[55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
```

```
[73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
```

```
[91] 91 92 93 94 95 96 97 98 99 100
```




seq()

```
> number<-seq(1,100,0.1)
```

```
> number
```

```
[1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1
```

```
[13] 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 3.0 3.1 3.2 3.3
```

```
...
```

```
[985] 99.4 99.5 99.6 99.7 99.8 99.9 100.0
```



ข้อมูลแบบตาราง

- ใน R เรียกข้อมูลแบบตารางว่า Data frame

```
> treatment <- rep(c("a", "b", "c"), each=3)
> treatment
[1] "a" "a" "a" "b" "b" "b" "c" "c" "c"
> value <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
> test.data <- data.frame(treatment, value)
> test.data
```

	treatment	value
1	a	1
2	a	2
3	a	3
4	b	4
5	b	5
6	b	6
7	c	7
8	c	8
9	c	9



การ save

```
> saveRDS (test.data, file="test.data.Rda")
```

```
> data<-readRDS ("test.data.Rda")
```

```
> data
```

	treatment	value
--	-----------	-------

1	a	1
---	---	---

2	a	2
---	---	---

3	a	3
---	---	---

4	b	4
---	---	---

5	b	5
---	---	---

6	b	6
---	---	---

7	c	7
---	---	---

8	c	8
---	---	---

9	c	9
---	---	---

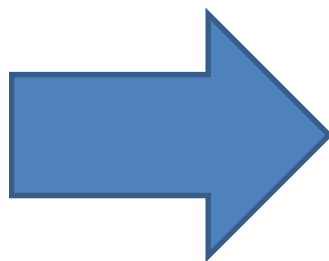


การดึงข้อมูลจาก Excel

- ข้อมูลในรูปของตาราง

ผู้ชิม	เบียร์ A	เบียร์ B
1	10	7
2	6	3
3	5	6
4	7	7
5	10	7
6	6	4
7	7	5
8	8	6
9	6	3
10	5	2

กรอกลง Excel



	A	B	C	D
1	Judge	BeerA	BeerB	
2	1	10	7	
3	2	6	3	
4	3	5	6	
5	4	7	7	
6	5	10	7	
7	6	6	4	
8	7	7	5	
9	8	8	6	
10	9	6	3	
11	10	5	2	
12				



Save file

Documents
Downloads
Music
Pictures
Videos
C:
Work

File name:
Save as type:
Authors:

Excel Template (*.xltx)
Excel Macro-Enabled Template (*.xltn)
Excel 97-2003 Template (*.xlt)
Text (Tab delimited) (*.txt)
Unicode Text (*.txt)
XML Spreadsheet 2003 (*.xml)
Microsoft Excel 5.0/95 Workbook (*.xls)
CSV (Comma delimited) (*.csv)
Formatted Text (Space delimited) (*.prn)
Text (Macintosh) (*.txt)
Text (MS-DOS) (*.txt)
CSV (Macintosh) (*.csv)
CSV (MS-DOS) (*.csv)
DIF (Data Interchange Format) (*.dif)
SYLK (Symbolic Link) (*.slk)
Excel Add-In (*.xlam)
Excel 97-2003 Add-In (*.xla)
PDF (*.pdf)
XPS Document (*.xps)
OpenDocument Spreadsheet (*.ods)

Tab delimited
CSV

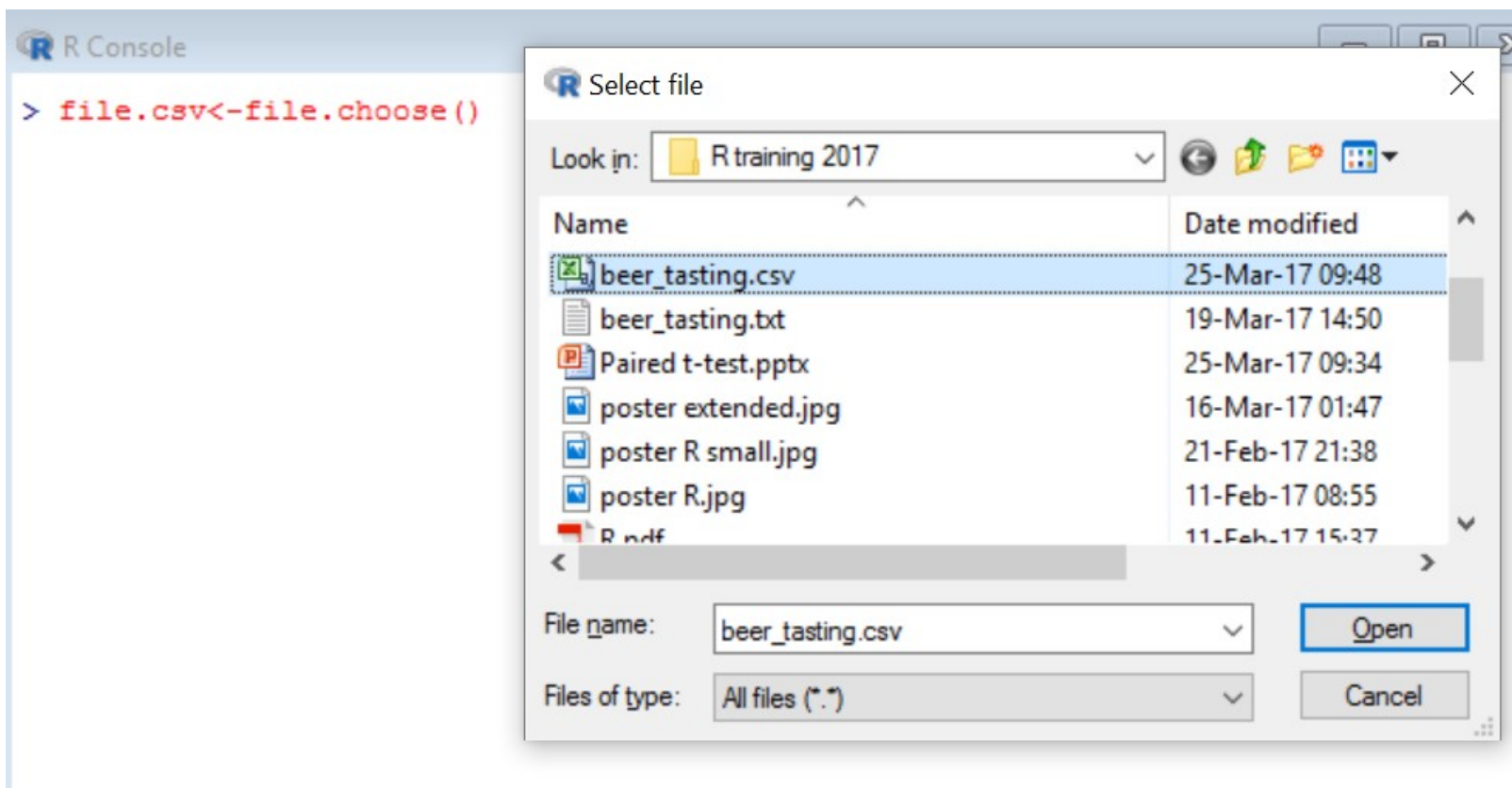
Tools Save



การอ่านไฟล์ CSV และ txt

- กรณีไฟล์ csv (Comma Separated Value)

> file.csv<- file.choose() #แสดงไดอะล็อกเพื่อเลือกไฟล์ที่ต้องการ





ตัวแปลชื่อไฟล์

ระบุแถวแรกเป็นหัวตาราง (True)

```
> data1<-read.csv(file.csv,header=T)
```

```
> data1
```

	Judge	BeerA	BeerB
1	1	10	7
2	2	6	3
3	3	5	6
4	4	7	7
5	5	10	7
6	6	6	4
7	7	7	5
8	8	8	6
9	9	6	3
10	10	5	2



- กรณีนีไฟล์ txt (Tab delimited)

```
> file.tab<-file.choose()  
> data2<-read.table(file.tab,header=T)  
> data2
```

	Judge	BeerA	BeerB
1	1	10	7
2	2	6	3
3	3	5	6
4	4	7	7
5	5	10	7
6	6	6	4
7	7	7	5
8	8	8	6
9	9	6	3
10	10	5	2

#รวมทั้งสองคำสั่งเข้าด้วยกัน

```
> data <-read.csv(file.choose(),header=T)
```

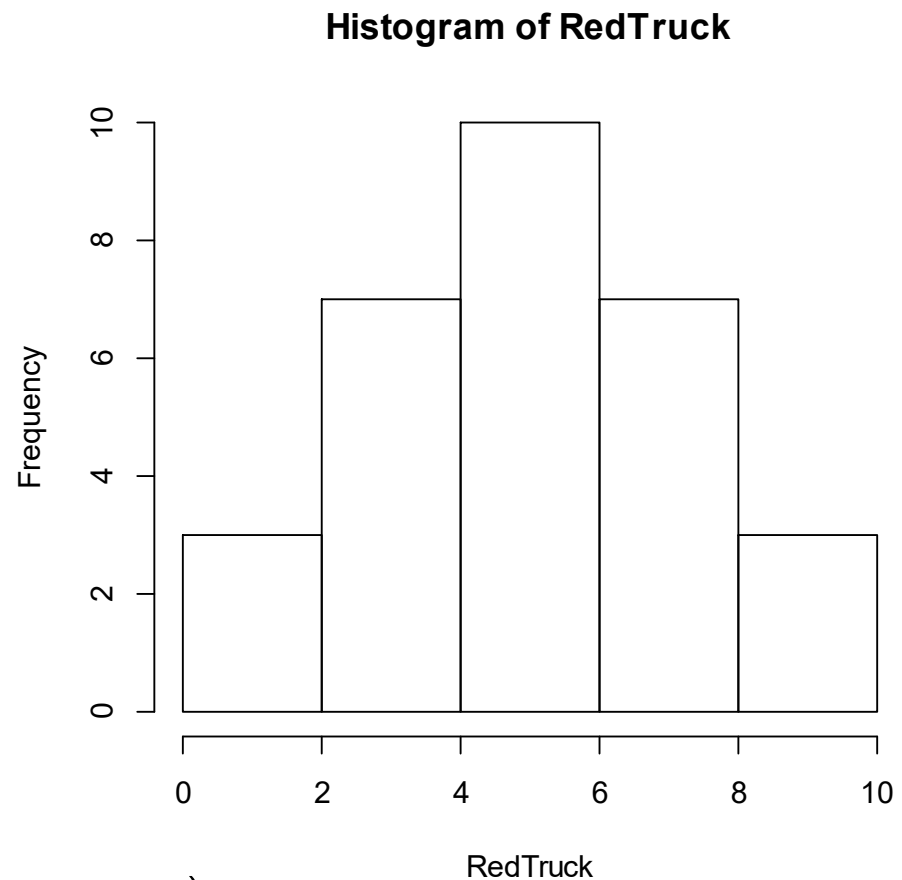


Descriptive statistics

```
> wine<-read.table(file.choose(),header=T)
```

```
> attach(wine)
```

```
> hist(RedTruck)
```





การจัด layout สำหรับการพล็อต

multiframe *rowise*

```
> par(mfrow=c(2,2))
```

```
> hist(RedTruck)
```

```
> hist(WoopWoop)
```

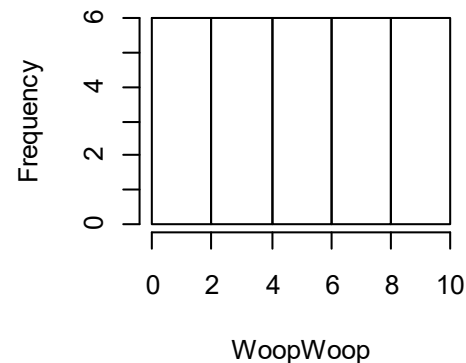
```
> hist(HobNob)
```

```
> hist(FourPlay)
```

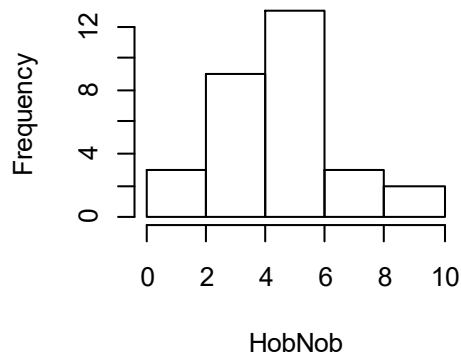
Histogram of RedTruck



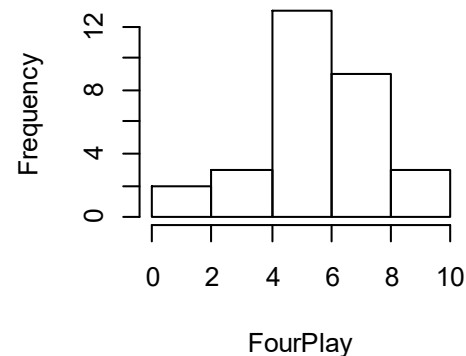
Histogram of WoopWoop



Histogram of HobNob



Histogram of FourPlay





Descriptive statistics

```
> wine<-read.table(file.choose(),header=T)
```

```
> mean(RedTruck)
```

```
[1] 5.5
```

```
> sd(RedTruck)
```

```
[1] 2.255262
```

```
> var(RedTruck)
```

```
[1] 5.086207
```

```
> median(RedTruck)
```

```
[1] 5.5
```

```
> summary(wine)
```

RedTruck	WoopWoop	HobNob	FourPlay
Min. : 1.0	Min. : 1.0	Min. : 1.000	Min. : 1.000
1st Qu.: 4.0	1st Qu.: 3.0	1st Qu.: 4.000	1st Qu.: 5.000
Median : 5.5	Median : 5.5	Median : 5.000	Median : 6.000
Mean : 5.5	Mean : 5.5	Mean : 5.033	Mean : 5.967
3rd Qu.: 7.0	3rd Qu.: 8.0	3rd Qu.: 6.000	3rd Qu.: 7.000
Max. :10.0	Max. :10.0	Max. :10.000	Max. :10.000



Writing a script

- การสร้างสคริปต์ทำให้เราสามารถทำงานซ้ำๆ ได้รวดเร็ว ทำซ้ำได้ดี
- File > New Script
 - อ่านไฟล์
 - ดูข้อมูลใน Dataframe
 - พล็อต Histogram
 - แสดง descriptive statistics
- File > Save as...



#Wine data

```
wine<-read.table(file.choose(),header=T)
```

```
attach(wine)
```

```
class(wine)
```

```
names(wine)
```

```
wine
```

```
summary(wine)
```

```
#plot histogram
```

```
par(mfrow=c(2,2))
```

```
hist(RedTruck,xlab="Rating")
```

```
hist(WoopWoop,xlab="Rating")
```

```
hist(HobNob,xlab="Rating")
```

```
hist(FourPlay,xlab="Rating")
```

```
detach(wine)
```

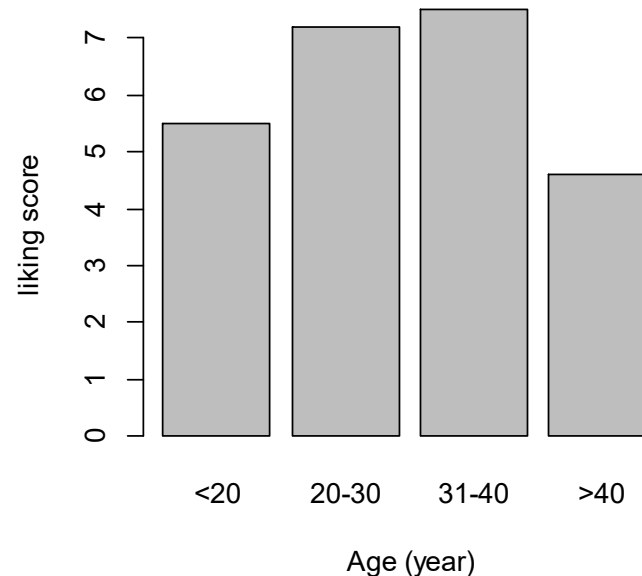
- Open Script

- ที่เมนู Edit → Run All



การพล็อตกราฟ

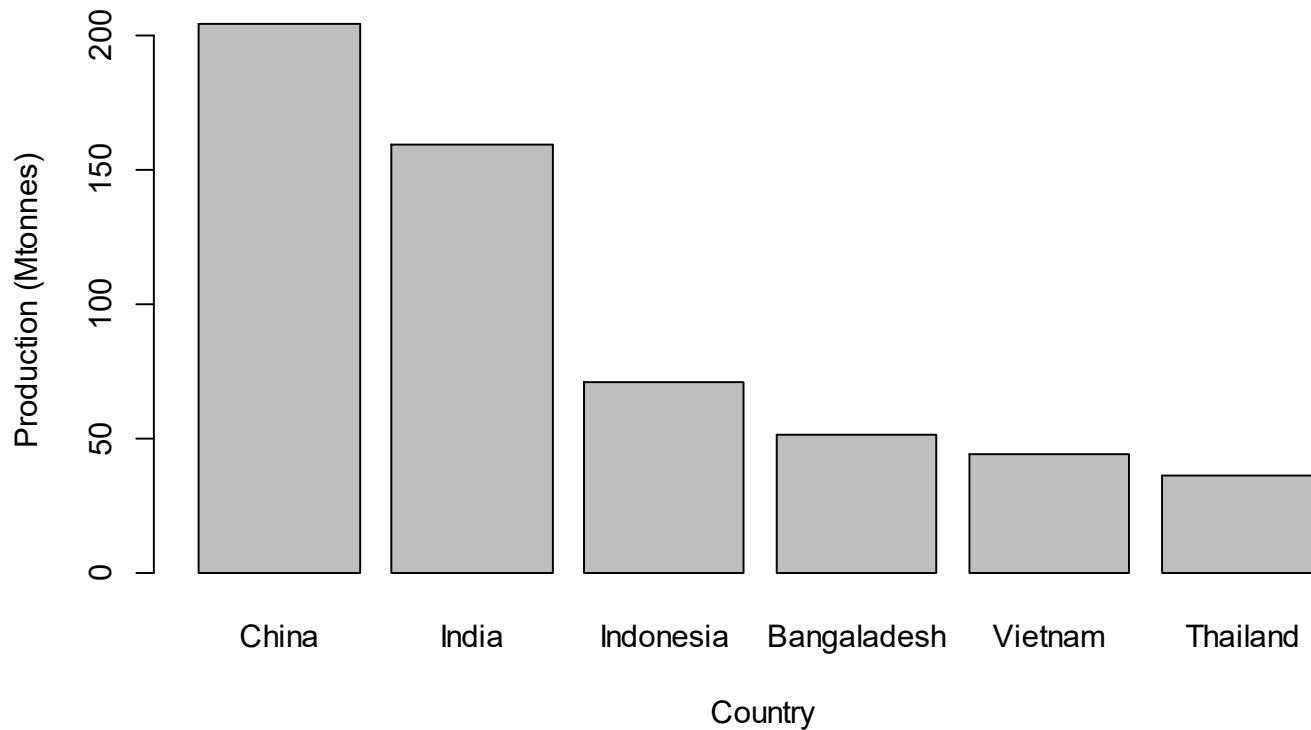
```
> liking<-c(5.5,7.2,7.5,4.6)
> barplot(liking)
> age<-c("<20","20-30","31-40",>40")
> barplot(liking,names.arg=age,xlab="Age
(year)",ylab="liking score")
```





การพล็อตกราฟ

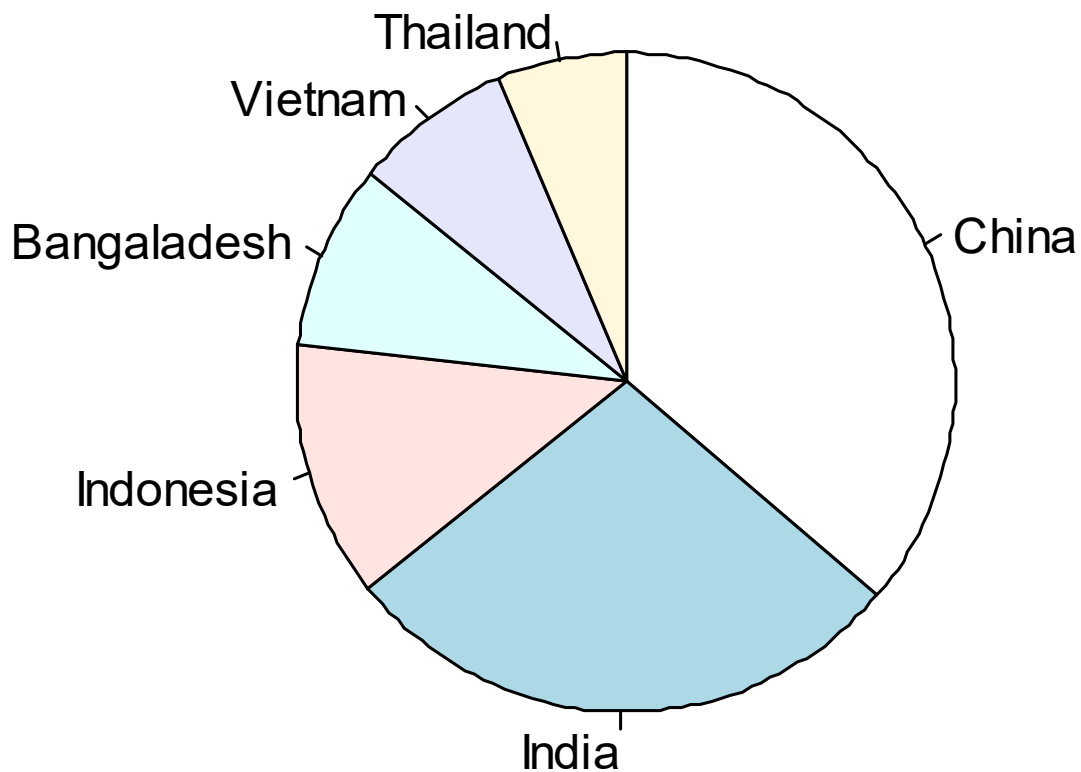
```
> rice<-read.table(file.choose(),header=T)  
> barplot(rice$Production,  
names.arg=rice$Country,xlab="Country",ylab="Production (Mtonnes)")
```





การพล็อตกราฟ

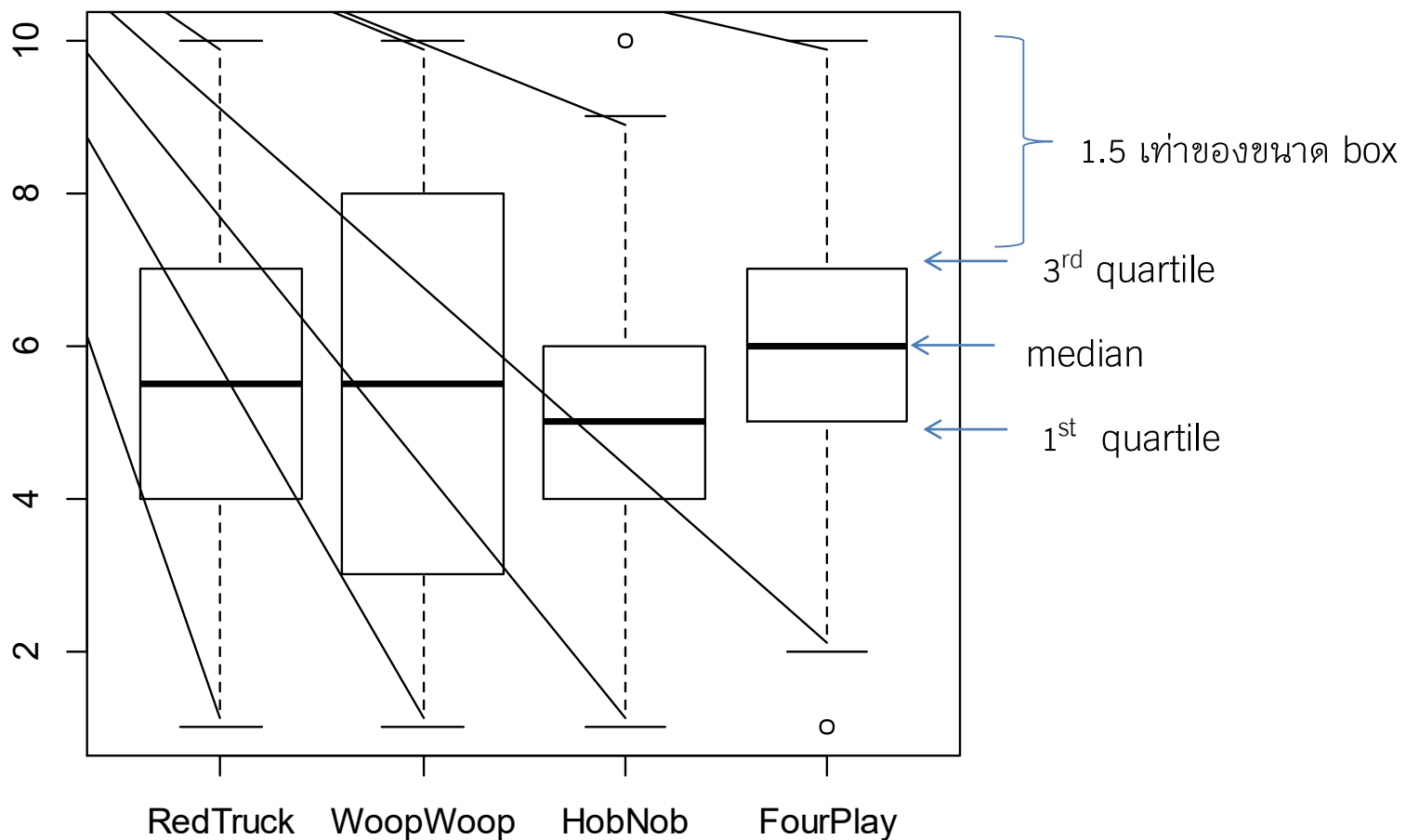
```
> pie(rice$Production, label=rice$Country, clockwise=T)
```





Box and whiskers plot

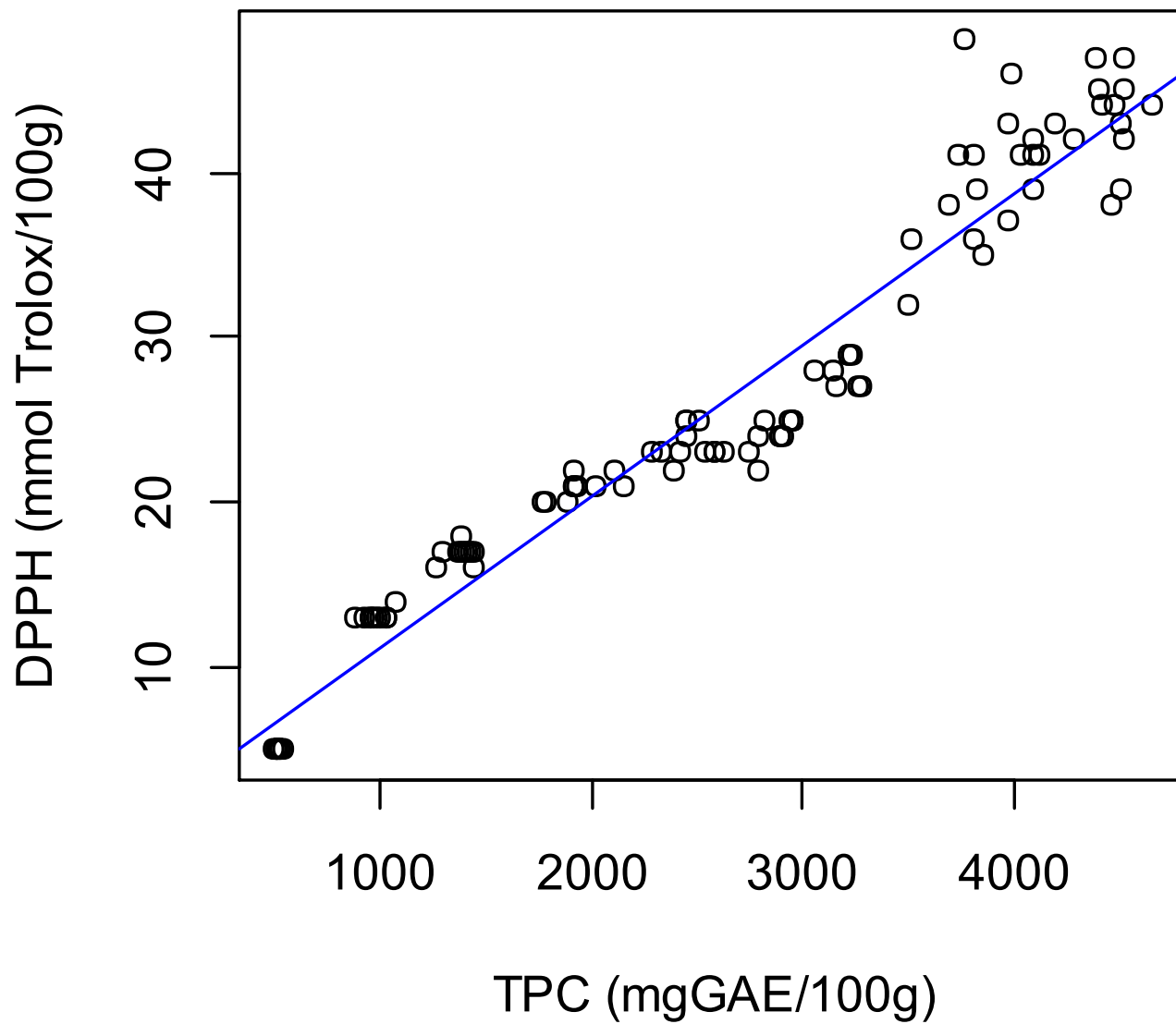
```
> boxplot(wine)
```





Correlation

```
#phenolic compound
phenolic<-read.table(file.choose(),header=T)
print(class(phenolic))
print(names(phenolic))
attach(phenolic)
plot(DPPH~TPC,ylab="DPPH (mmol
Trolox/100g)",xlab="TPC (mgGAE/100g)")
abline(lm(DPPH~TPC),col="blue")
print(cor.test(DPPH,TPC))
```



```
> coconut<-read.table(file,header=T)
```

```
> cor(coconut)
```

	Solid	pH	Carbo	Reducing
Solid	1.0000000	-0.9703021	0.5255827	0.8086385
pH	-0.9703021	1.0000000	-0.6150656	-0.8612920
Carbo	0.5255827	-0.6150656	1.0000000	0.9098585
Reducing	0.8086385	-0.8612920	0.9098585	1.0000000



Regression analysis

- หาความสัมพันธ์ระหว่างตัวแปร y กับ x
- Simple regression: $y = mx + b$

```
> data<-read.table(file.choose(),header=T)
```

```
> data
```

	Conc	Area
1	16.733	515688
2	8.367	256757
3	5.578	170152
4	3.347	100142
5	1.673	52578
6	0.558	14322

```
> model1<-lm(Area~Conc) # lm(Y~ model)
```



```
> summary(model1)
```

```
Call:
```

```
lm(formula = Area ~ Conc)
```

```
Residuals:
```

```
          1          2          3          4          5          6
215.14   -48.59  -420.87 -1450.87  2743.32 -1038.14
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1892.6      1010.7   -1.872    0.134
Conc          30918.9       124.5  248.335 1.58e-09 ***
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1653 on 4 degrees of freedom
```

```
Multiple R-squared:  0.9999,    Adjusted R-squared:  0.9999
```

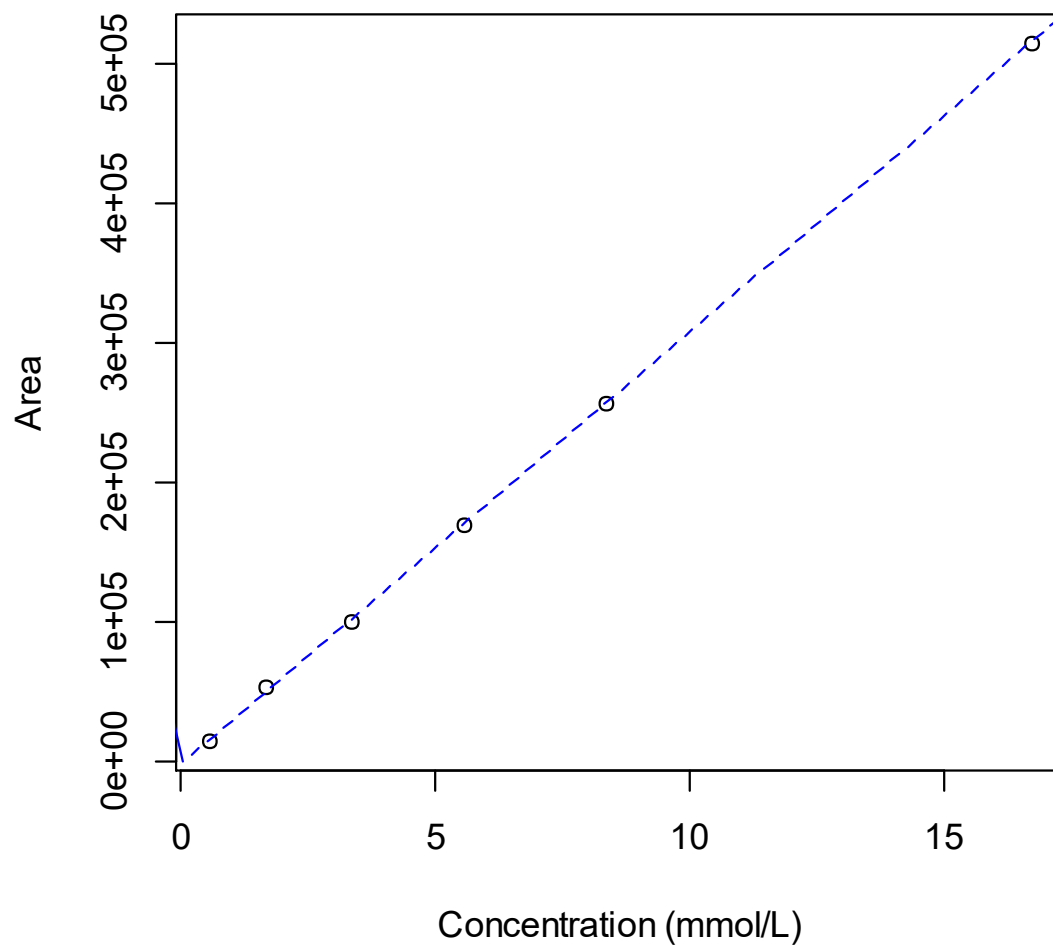
```
F-statistic: 6.167e+04 on 1 and 4 DF,  p-value: 1.577e-09
```

นั่นคือได้สมการ $\text{Area} = 30918 \cdot \text{Conc} - 1892.6$



```
> plot (Area~Conc)
```

```
> abline (lm (Area~Conc) , col="blue", lty=2)
```





- Multiple regression

```
> file<-file.choose()  
> rsm<-read.table(file,header=T)  
> rsm
```

	x1	x2	y
1	-1.000	-1.000	76.5
2	-1.000	1.000	77.0
3	1.000	-1.000	78.0
4	1.000	1.000	79.5
5	0.000	0.000	79.9
6	0.000	0.000	80.3
7	0.000	0.000	80.0
8	0.000	0.000	79.7
9	0.000	0.000	79.8
10	1.414	0.000	78.4
11	-1.414	0.000	75.6
12	0.000	1.414	78.5
13	0.000	-1.414	77.0



```
> model3<-lm(y~x1*x2+I(x1^2)+I(x2^2),data=rsm)
> summary(model3)
```

Call:

```
lm(formula = y ~ x1 * x2 + I(x1^2) + I(x2^2), data = rsm)
```

Coefficients:

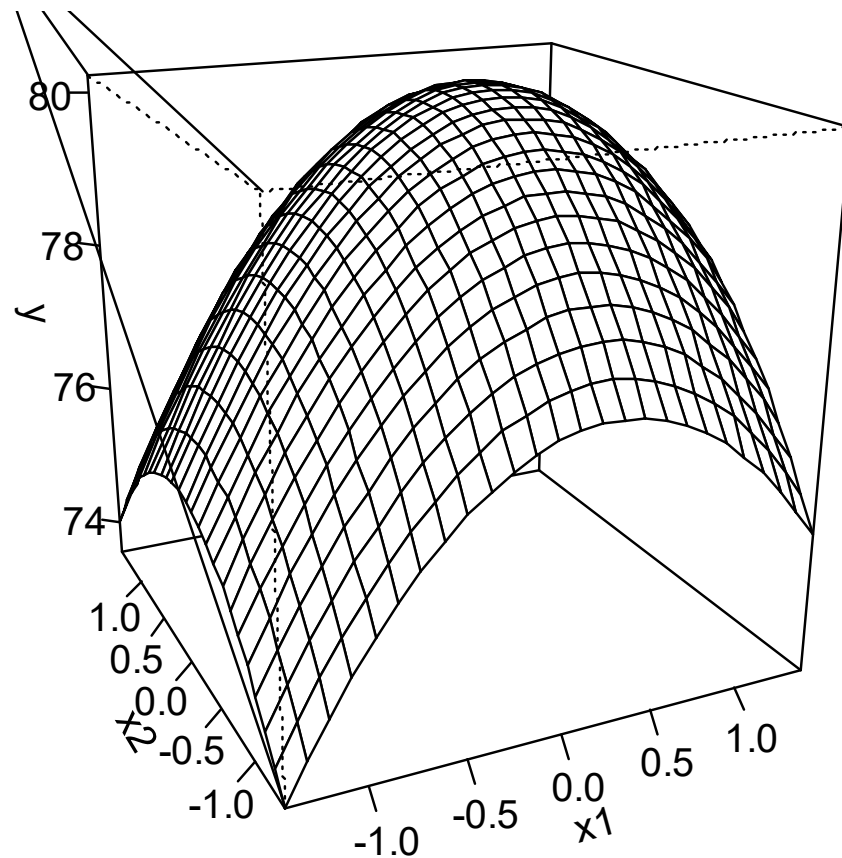
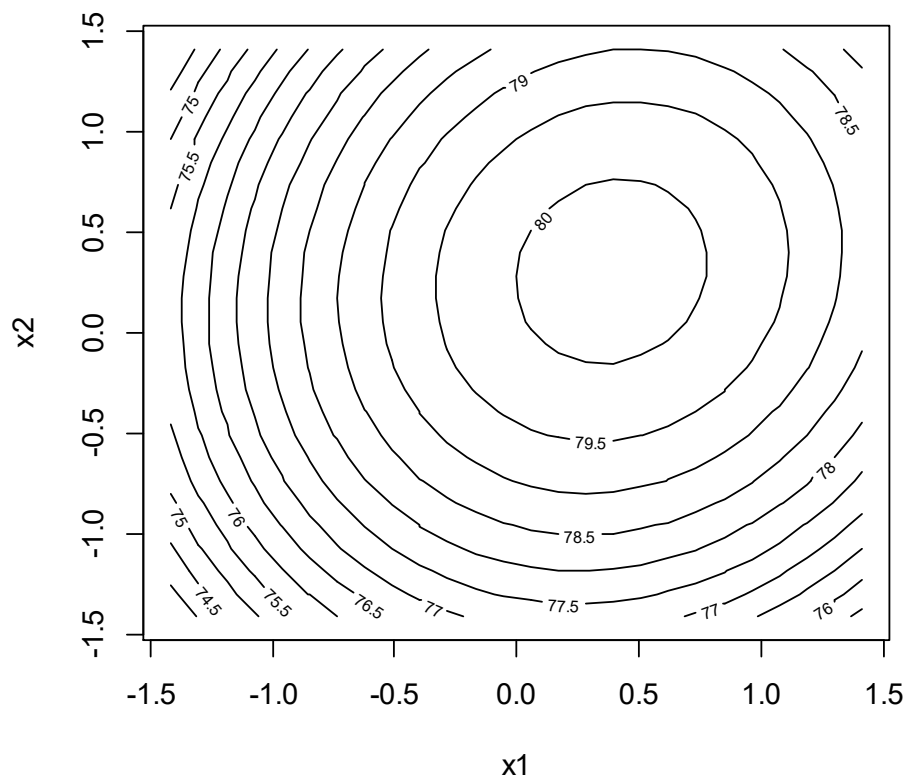
	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	79.93995	0.11909	671.264	< 2e-16	***
x1	0.99505	0.09415	10.568	1.48e-05	***
x2	0.51520	0.09415	5.472	0.000934	***
I(x1^2)	-1.37645	0.10098	-13.630	2.69e-06	***
I(x2^2)	-1.00134	0.10098	-9.916	2.26e-05	***
x1:x2	0.25000	0.13315	1.878	0.102519	

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2663 on 7 degrees of freedom

Multiple R-squared: 0.9827, Adjusted R-squared: 0.9704

F-statistic: 79.67 on 5 and 7 DF, p-value: 5.147e-06



Contour และ 3D plot จากแฟกเกจ rsm



การติดตั้งแพ็คเกจเพิ่มเติม

- แพ็คเกจ (package) เป็นชุดของโปรแกรมสำหรับการวิเคราะห์เฉพาะทาง
- แนะนำให้ติดตั้งแบบ online

```
> install.packages("psych")
```

```
> library(psych)
```

```
> describe(wine)
```

```
> install.packages("agricolae")
```

```
> library(agricolae)
```