

```
RGui
File Edit View Misc Packages Windows Help
R Console
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> setwd("C:/orCourse ")
> mkmsc <- read.table("marks.dat", header=TRUE)
> mk2nd <- read.table("marks2.dat", header=TRUE)
> attributes(mk2nd)
$names
[1] "candno" "exam1" "exam2" "exam3"

$class
[1] "data.frame"

$row.names
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 :
[37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55

R version 2.8.1 (2008-12-22)
```

## البيانات وإدارة الدورة وإحصائيات بسيطة:

### أفكار أساسية:

### أشياء Objects:

تعمل R على أشياء Objects. جميع الأشياء تمتلك الخواص التالية والتي نطلق عليها الصفات الجوهرية **intrinsic attributes** :

- نمط أو صيغة mode: ويبين نوع الشيء - والصيغ الممكنة هنا هي: عددي numeric أو مركب complex أو منطقي logical أو نص character أو قائمة list.

### مثال:

```
> x <- 1
> A <- "Gomphotherium"
```

```

> compar <- TRUE
> z <- 1i
> mode(x)
[1] "numeric"
> mode(A)
[1] "character"
> mode(compar)
[1] "logical"
> mode(z)
[1] "complex"
> B <- c("Ali", "Ahmad", "Omar", "Saad")
> mode(B)
[1] "character"

```

• طول length: عدد المكونات والتي تصنع الشيء.

```

> length(x)
[1] 1
> length(A)
[1] 1
> length(compar)
[1] 1
> length(z)
[1] 1
> length(B)
[1] 4
>

```

وببساطة الشيء هو طريقة مناسبة لتخزين المعلومات. ففي النمذجة الرياضية نحتاج لتخزين مشاهدات لمتغير ذو أهمية. وهذا يتم عن طريق استخدام متجه عددي. لاحظ انه لا يوجد قيم مفردة *Scalar* في R فالعدد الفردي هو متجه عددي له طول 1. المتجهات نشير إليها كتركيبات ذرية *Atomic Structures* حيث جميع مكوناتها لها نفس النمط أو الصيغة.

إذا كان الشيء يخزن معلومات فإننا نحتاج إلى تسميته بحيث نستطيع الإشارة إليه لاحقاً. يطلق على اسم الشيء معرف *Identifier* والمعرف يتم إختياره بواسطة المستخدم. ونستطيع إختيار أي إسم معرف على أن تتوفر فيه الشروط التالية:

- أي تشكيلة من الحروف والأرقام و النقطة ويستحسن أن يكون الإسم واصفا لطبيعة الشيء.
- لا يبدأ المعرف بعدد أو نقطة فمثلا المعرف moonbase3.sample مقبول ولكن 3moonbase.sample أو moonbase3.sample. غير مقبولة.

- المعرفات حساسة لنوع الحرف *Case Sensitive* فمثلا moon.sample تختلف عن moon.Sample
  - بعض النصوص تعتبر قيم محفوظة مثل c و q و C و D و F و ا و T. ولهذا يجب تجنب إستخدامها كمعرفات.
- في العادة نحن نهتم بمجموعات بيانات *Data Sets* تتكون من عدة متغيرات ففي R مجموعات البيانات تمثل بشيء يعرف بإطار البيانات *Data Frame*. وكجميع الأشياء إطار البيانات له الخواص الجوهرية نمط و طول. إطار البيانات له نمط قائمة وطول إطار البيانات هو عدد المتغيرات التي يحويها. وكبقية الأشياء الكبيرة يمتلك إطار البيانات صفات اخرى خلاف النمط والطول وهذه تسمى صفات غير جوهرية *Non-intrinsic Attributes* مثل:
- أسماء *names*: وهي أسماء المتغيرات التي تكون مجموعة البيانات.
  - أسماء سطور *row.names* وهي أسماء الأفراد التي تتبع لهم المشاهدات.
  - فئة *class*: يمكن إعتبار هذه الصفة كتوصيف مفصل عن نوع الشيء وفي هذه الحالة الفئة هي *data.frame*
- صفة الفئة تخبر بعض الدوال *generic functions* كيف تتعامل مع الشيء فمثلا أشياء من فئة *data.frame* تعرض على الشاشة بطريقة خاصة.

```
> names(x)
NULL
> names(A)
NULL
> class(x)
[1] "numeric"
> class(A)
[1] "character"
> class(compar)
[1] "logical"
> class(z)
[1] "complex"
> class(B)
[1] "character"
>
```

## دوال و حجج وقيم مرجعة

### Functions, arguments and return values

تعمل R بواسطة إستدعاء دوال *calling functions* وفكرة الدالة معروفة حيث تأخذ الدالة مجموعة من المدخلات *inputs* وتقوم بتطبيقهم *map* لمخرج *output* واحد. في

R المدخلات والمخرجات هي أشياء. نطلق على المدخلات حجج *arguments* والمخرجات قيم مرجعة *return values*. الكثير من الدوال المهمة والمفيدة موجودة في تركيب R بينما هناك أخرى موجودة في حزم يمكن تحميلها عند اللزوم وسوف نتحدث عنها لاحقاً.

وأحد أهم ميزات R هو سهولة تعريف المستخدم لدواله الخاصة والتي سنتحدث عنها لاحقاً. الدوال تعتبر أيضاً أشياء ونمط الدالة هو *function*. أحيانا الدوال لها تأثيرات جانبية *side-effects* فمثلاً استدعاء دالة قد يسبب رسم بياني وفي كثير من الأحيان يكون التأثير الجانبي أهم من القيمة المرجعة.

## فضاء العمل و دليل العمل

### Workspace and working directories

خلال جلسة في R يتم توليد عدد من الأشياء. فمثلاً يمكن أن نولد متجهات و إطارات بيانات ودوال خلال هذه الجلسة هذه الأشياء تخزن في منطقة من ذاكرة الحاسب يطلق عليها فضاء عمل *workspace*. إذا أردنا حفظ الأشياء لإستخدام لاحق فإننا نوعز لـ R لكتابتهم في ملف في دليل العمل *working directory* الحالي. لاحظ الفرق الأشياء التي في الذاكرة مؤقتة تفقد بعد الخروج من النظام بينما الملفات دائمة حيث تخزن على القرص الصلب. إدارة الأشياء والملفات هي جزء مهم في إستخدام R بشكل فعال. ويمكن معرفة دليل العمل كالتالي:

```
> getwd()
```

```
[1] "C:/Documents and Settings/amb/My Documents"
```

```
>
```

ويتم تخزين فضاء العمل كالتالي:

```
> save.image("C:\\Documents and  
Settings\\amb\\Desktop\\20100426.RData")
```

وإسترجاع كالتالي:

```
> load("C:\\Documents and Settings\\amb\\My  
Documents\\20100426.RData")
```

```
>
```

## المتجهات والتخصيص

### Vectors and assignment

يمكن إحداث المتجهات في محث التحكم *command prompt* بإستخدام دالة الربط *concatenation function* *c(...)* والتي لها البناء اللغوي *syntax* التالي:

*c(object1,object2,...)*

هذه الدالة تأخذ الحجج من نفس النمط أو الصنف وتعيد متجها يحوي هذه القيم

```
> c(1,2,3)
[1] 1 2 3
> c("Ali","Bet","Cat")
[1] "Ali" "Bet" "Cat"
```

لكي نستطيع استخدام المتجهات يجب إعطائها معرف ( طبعا نحن لانريد ان نكتب المتجه من جديد كلما نريد استخدامه) ولكي نعطي اسم معرف للمتجه نستخدم عامل

التخصيص *assignment operator* <-  
 name <- expression  
 name الآن يشير لشيء قيمته هي نتيجة تقييم العبارة *expression*

```
> numbers <- c(1,2,3)
> people <- c("Ali","Bet","Cat")
> numbers
[1] 1 2 3
> people
[1] "Ali" "Bet" "Cat"
```

الرمز # في أول السطر يخبر R أن مايتبع هو تعقيب أو تعليق *comment*.  
 نستطيع إجراء عمليات بسيطة على المتجهات كالتالي:

```
> c(1,2,3)+c(4,5,6)
[1] 5 7 9
> numbers + numbers
[1] 2 4 6
> numbers - c(8,7.5,-2)
[1] -7.0 -5.5 5.0
> c(1,2,4)*c(1,3,3)
[1] 1 6 12
> c(12,12,12)/numbers
[1] 12 6 4
```

لاحظ في المثال السابق أن عمليتي الضرب والقسمة أجريت عنصر بعنصر.

ملاحظة: نستطيع إستدعاء الأوامر التي سبق أن أدخلناها بإستخدام زر السهم أعلى ↑ .  
 نتيجة أي عملية حسابية يمكن أن يعطى لها معرف لكي يمكن إستخدامها لاحقاً

```
> calc1 <- numbers + c(8,7.5,-2)
> calc2 <- calc1 * calc1
```

```

> calc1
[1] 9.0 9.5 1.0
> calc2
[1] 81.00 90.25 1.00
> calc1 <- calc1 + calc2
> calc1
[1] 90.00 99.75 2.00
> calc2
[1] 81.00 90.25 1.00

```

إذا حاولنا جمع متجهين لها أطوال مختلفة فإن R تستخدم قاعدة إعادة استخدام أو قاعدة التدوير *recycling rule* حيث يتم تكرار المتجه الأصغر حتى تصبح الأطوال واحدة

```

> small <- c(1,2)
> large <- c(0,0,0,0,0,0)
> large + small
[1] 1 2 1 2 1 2

```

ملاحظة: لمسح الأشياء من فضاء العمل نستخدم

```

> rm(list=objects())

```

مثال بسيط:

أوزان 5 خراف بالكيلو جرام هي 84.5, 72.6, 75.7, 94.8, 71.3 سوف نضع هذه القيم في متجه ونجري عليها بعض العمليات

```

> weight <- c(84.5, 72.6, 75.7, 94.8, 71.3)
> weight
[1] 84.5 72.6 75.7 94.8 71.3
> total <- sum(weight)
> numobs <- length(weight)
> meanweight <- total/numobs
> meanweight
[1] 79.78
...
> mean(weight)
[1] 79.78

```

## أطر البيانات Data Frames:

إطار البيانات هو شئ *object* في R والذي يمكن إعتباره كتمثيل لمجموعة بيانات *data set*. يتكون إطار البيانات من متغيرات (متجهات عامودي) لها نفس الطول و كل سطر يعود أو يتبع وحدة تجربة *experimental unit*. البناء اللغوي *Syntax* لتكوين إطار بيانات هو

```
name <- data.frame(variable1, variable2,  
...)
```

وأي متغير في إطار بيانات يمكن الوصول إليه أو إستخدامه بإستعمال الرمز \$ كالتالي:

**name\$variable**

بعد تكوين إطار بيانات نستطيع مشاهدته وتحريره في شكل صفحة نشر *spreadsheet* مستخدمين الأمر *fix(...)* (أو بشكل مكافئ *data.entry(...)*) ويمكن إضافة متغيرات جديدة لإطار البيانات عن طريق التخصيص.

### عودة للمثال السابق:

لنفترض بالإضافة للأوزان لدينا إرتفاع الخراف مقاسة بالسنتيمتر كالتالي:  
86.5 71.8 77.2 84.9 75.4 سوف ندخل هذه البيانات في متغير وكذلك سوف نكون تركيبة تربط بين الوزن والإرتفاع لنفس الوحدة التجريبية (هنا خروف) ويتم هذا عن طريق إضافة كل متغير إلى إطار بيانات وسوف نسمي هذا الإطار *sheep* ثم ننظر إليه بإستخدام *fix(sheep)*

```
> height <- c(86.5, 71.8, 77.2, 84.9, 75.4)  
> sheep <- data.frame(weight, height)  
> mean(sheep$height)  
[1] 79.16  
> fix(sheep)
```

لنفترض ان لدينا مقاس آخر لمتغير الطول بالسنتيمتر **130.4, 100.2, 109.4, 140.6, 101.4** ونضيفه لإطار البيانات كالتالي:

```
> sheep$backlength <- c(130.4, 100.2, 109.4, 140.6,
                        101.4)
```

تمرين: أنظر لإطار البيانات لتشاهد ماذا حدث.

### تحليل وصفي:

نستطيع الحصول على مجموعة من الإحصائيات الوصفية باستخدام الدالة `summary(...)` وحجة الأمر ممكن أن تكون متغير واحد أو إطار بيانات. يتكون المخرج من جدول كالتالي:

```
> summary(sheep$weight)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
71.30	72.60	75.70	79.78	84.50	94.80

```
> summary(sheep)
```

weight	height	backlength
Min. : 71.30	Min. : 71.80	Min. : 100.2
1st Qu. : 72.60	1st Qu. : 75.40	1st Qu. : 101.4
Median : 75.70	Median : 77.20	Median : 109.4
Mean : 79.78	Mean : 79.16	Mean : 116.4
3rd Qu. : 84.50	3rd Qu. : 84.90	3rd Qu. : 130.4
Max. : 94.80	Max. : 86.50	Max. : 140.6

```
> IQR(sheep$height)
```

```
[1] 9.5
```

```
> sd(sheep$backlength)
```

```
[1] 18.15269
```

### نظام المساعدة في R:

توجد عدة طرق للحصول على المساعدة في R.

- إذا كان هناك أي تساؤل عن دالة معينة نستخدم ؟ يتبعها إسم الدالة فتظهر نافذة المساعدة لتلك الدالة مثل:

```
> ?mean
```

```
> ?setwd
```

```
> ?t.test
```

- للحصول على مساعدة عامة نستخدم `help.start()` والتي تظهر نافذة تسمح بتصفح المعلومات التي تحتاج إليها.



## إدارة الجلسة ووضوح الرؤية Session management and visibility

جميع الأشياء *objects* التي كونت خلال جلسة في R تخزن في فضاء عمل *workspace* في الذاكرة ونستطيع أن نشاهد الأشياء التي في فضاء العمل في الوقت الحالي باستخدام الأمر `object()` (لاحظ القوسين فهي مهمة جدا لكي يعمل الأمر) في المثال السابق نجد

```
> objects()
```

```
[1] "height" "meanweight" "numobs" "sheep" "total"  
[6] "weight"
```

المعلومات في المتغيرات `height` و `weight` هي الآن مغلقة *encapsulated* في إطار البيانات `sheep` ولهذا سوف نقوم بتنظيف فضاء العمل وذلك بإزالة المتغيرات `height` و `weight` وبقية الأشياء التي لم نعد في حاجتها باستخدام الدالة `rm(...)` وفي المثال السابق

```
> rm(height,weight,meanweight,numobs,total)
```

```
> objects()
```

```
[1] "sheep"
```

المتغيرات `height` و `weight` نستطيع الوصول إليها الآن فقط من خلال إطار البيانات `sheep` كالتالي

```
> weight
```

```
Error: Object "weight" not found
```

```
> sheep$weight
```

```
[1] 84.5 72.6 75.7 94.8 71.3
```

وميزة تغليف *encapsulation* المعلومات هي في كوننا نستطيع تكوين إطار بيانات آخر مثلا لنقل بقر `cow` لهو متغيرات `height` و `weight` بدون خلط أو غموض. في حال إرادتنا استخدام المتغيرات في إطار البيانات `sheep` بشكل مكثف نجعلها مرئية من خط الأمر باستخدام الأمر `attach(...)` وبعد الإنتهاء من هذه المتغيرات نعميها بالأمر `detach()` (لاحظ الأقواس) بحيث تصبح المتغيرات المغلفة غير مرئية

```
> weight
```

```
Error: Object "weight" not found
```

```
> attach(sheep)
```

```
> weight
```

```
[1] 84.5 72.6 75.7 94.8 71.3
```

```
> detach()  
> weight  
Error: Object "weight" not found
```

نستطيع تخزين فضاء العمل الحالي في ملف متى شئنا ومن الممارسات الجيدة تكوين مجلد *folder* لأي مشروع جديد نقوم به. سوف نكون مجلد أو دليل عمل *Working directory* كالتالي:

- كون مجلد وسمه مثلا *orCourse* في C: (او اي قرص آخر)
- أجعله دليل عمل

```
> setwd("C:/orCourse")
```

وأمر تخزين فضاء العمل هو *save.image(...)* وفي حال عدم تحديد إسم للملف فإن البرنامج يعطيه الإسم *Rdata*. (لاحظ النقطة قبل الإسم). وطبعاً من الأفضل استخدام إسم يدل على طبيعة المشروع مثل

- تخزين فضاء العمل الحالي

```
> save.image("example1.Rdata")
```

- سرد الملف الموجود في دليل العمل

```
> dir()
```

الأمر *dir()* يعطي سرد لجميع الملفات في دليل العمل الحالي سوف تلاحظ أن مجلدك *orCourse* يحوي الآن ملف *example1.Rdata* هذا الملف يحوي أشياء *objects* من الجلسة الحالية أي إطار البيانات *sheep* من المهم جدا التفريق بين أشياء *objects* (والتي هي موجودة في فضاء العمل والتي نسردها بالأمر *object()*) وفضاء العمل (والذي يمكن أن يخزن في ملف محتوي مجلد والذي يسرد بالأمر *dir()*).

## إستيراد بيانات :Importing data

عملياً نحتاج إلى إستيراد المعلومات من مجاميع بيانات *data sets* كبيرة جداً والتي هي مخزنة بشكل إلكتروني. سوف نستخدم بيانات مخزنة على شكل نص *text*. المتغيرات موجودة في أعمدة وأول سطر في العمود يحوي إسم المتغير. الملف *sheep.dat* يحوي أوزان وأطوال 100 خروف تم إختيارها عشوائياً. سوف نقوم بنسخ هذا الملف إلى دليل العمل *orCourse*. المعلومات تقرأ في R بإستخدام الدالة *read.table(...)*. هذه الدالة ترجع إطار بيانات.

- استرجع البيانات من مجلد عام

```
C:\Documents and Settings\amb\Desktop
```

```
copy C:\Documents and Settings\...\sheep.dat
```

- قم بلصق الملف في المجلد

```
paste C:\orCourse\
```

- إقرأ البيانات في R

```
> sheep2 <- read.table("sheep.dat", header=TRUE)
```

إستخدام *header = TRUE* يعطينا إطار بيانات يكون فيه السطر الأول من ملف البيانات مستخدماً كـ معرف لأعمدة مجموع البيانات. إذا أغفلنا *header = TRUE* فسوف يعتبر السطر الأول كأحد المشاهدات المعطاة.

## إختبار فرضيات :A hypothesis test

سوف نختبر الفرضية أن متوسط مجتمع الخراف  $\mu = 80\text{kg}$  أي

$$H_0 : \mu = 80,$$

$$H_1 : \mu \neq 80.$$

سوف نستخدم مستوى معنوية 5% أي  $\alpha = 0.05$  وبفرض أن أوزان الخراف لها توزيع طبيعي بتباين غير معروف لهذا نستخدم إختبار *t* (بذيلين) بإستخدام الدالة *t.test(...)* كالتالي:

```
> attach(sheep2) # to make variables accessible
```

```
> t.test(weight, mu=80)
```

```
One Sample t-test
```

```
data: weight
```

```
t = 2.1486, df = 99, p-value = 0.03411
```

```
alternative hypothesis: true mean is not equal to 80
```

```
95 percent confidence interval:
```

```
80.21048 85.29312
sample estimates:
mean of x
82.7518
```

للإختبار أي من الفرضيات البديلة (ذيل واحد)

$$H_1 : \mu > 80,$$

$$H_1 : \mu < 80.$$

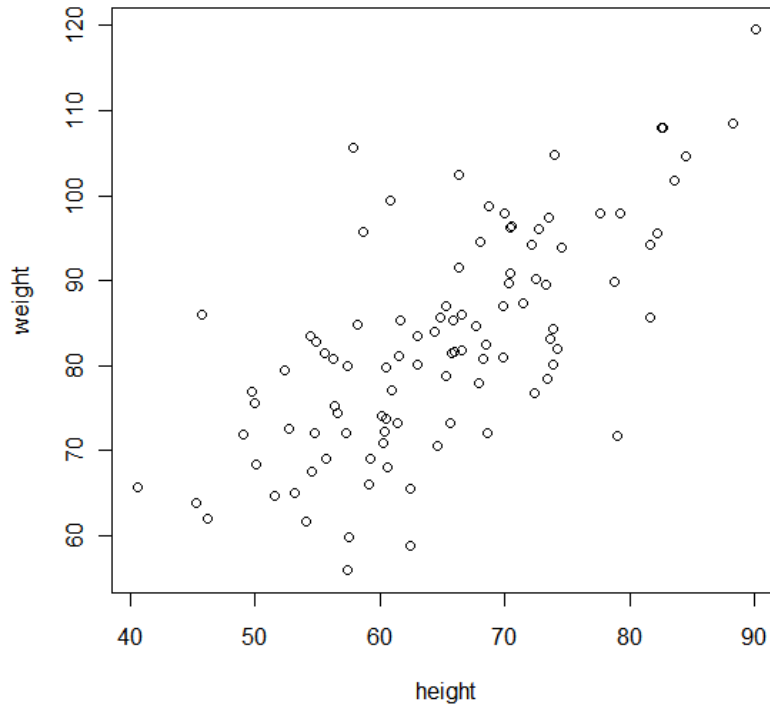
نستخدم

```
> t.test(weight, mu=80, alternative="greater")
> t.test(height, mu=66, alternative="less")
```

### نموذج خطي A linear model:

سوف نستخدم الإنحدار الخطي لإيجاد علاقة بين الوزن (صعب القياس) والطول (سهل القياس) ولكن اولاً نرسم المتغيرين في رسم أو مخطط إنتشار باستخدام الدالة `plot(...)` كالتالي:

```
> plot(height,weight)
```



لاحظ أن متغير محور X يعطى في الدالة أولاً.

لتطبيق نماذج خطية نستخدم الدالة `lm(...)` (لاحظ أن هذه الدالة مرنة جدا وتستخدم هنا في أبسط أشكالها ولها استخدامات أخرى نتطرق لها في حينها) لتطبيق إنحدار خطي بسيط للوزن على الطول

```
> lmres <- lm(weight ~ height)
```

نلاحظ شيئين:

- الحجة `weight ~ height` تسمى صيغة نموذج `model formula`. الرمز `~` يعني "موصوف بـ" `described by` فالأمر هنا يطلب نموذج خطي والذي فيه الوزن يوصف بالطول سوف نتكلم أكثر عن صيغ النموذج لاحقا.
  - بعد إدخال الدالة بحججها والضغط على إدخال لم يحدث شيء فقط ظهر محذ الأوامر `>` ولم تخرج نتائج؟ ذلك لأن `R` تعمل بوضع جميع المعلومات في شيء `object` يعاد بواسطة الدالة `lm(...)` وهذا يعرف بإسم شيء النموذج `model extractor` والذي يمكن مشاهدته وأستخدامه بإستخدام دوال مستخلصة `lmres functions`.
- أسهل طريقة لإستخلاص معلومة هي بطباعة معرف (إسم) شيء نموذج وفي هذا المثال `lmres` كما يمكننا إستخدام الدالة `summary(...)` لإعطاء تفاصيل أكثر أو الدالة `abline(...)` لتوليد رسم للخط المطبق.

```
> lmres
```

Call:

```
lm(formula = weight ~ height)
```

Coefficients:

```
(Intercept)      height
  26.0319         0.8724
```

```
> summary(lmres)
```

Call:

```
lm(formula = weight ~ height)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-23.1146  -6.1125  -0.1758   5.6578  29.1479
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  26.03185     6.05581   4.299 4.06e-05 ***
height       0.87239     0.09204   9.479 1.64e-15 ***
```

---

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

Residual standard error: 9.298 on 98 degrees of freedom  
Multiple R-squared: 0.4783, Adjusted R-squared: 0.473  
F-statistic: 89.84 on 1 and 98 DF, p-value: 1.64e-15

## الخروج والعودة لفضاء عمل مخزن `save.image("sheep2.Rdata")` :workspace

قبل إنهاء جلسة من الأفضل إزالة أي أشياء *objects* غير أساسية ثم تخزين فضاء العمل في دليل العمل الحالي

```
> save.image("sheep2.Rdata")
> quit()
```

لاحظ عند الخروج فإن R يسأل فيما إذا كان يراد تخزين فضاء العمل ولو وافقنا فإنه سوف يعيد تخزين فضاء العمل في ملف *Rdata*. الافتراضي ولهذا عند طلب التخزين نرفض تخزينه مرة أخرى (هذا إجراء إختياري في R). لكي نعود لفضاء العمل الذي تم تخزينه نعيد تشغيل R ونعيد تعريف دليل العمل ثم نستخدم الدالة `load(...)` لتحميل فضاء العمل

```
> setwd("C:/orCourse ")
> load("sheep2.Rdata")
> objects()
```

نلاحظ وجود جميع الأشياء من الجلسة السابقة.

## خواص الأشياء `Properties of objects`

الدالة `length(...)` تعطي طول المتجه فمثلا

```
> length(sheep2)
[1] 2
> length(plot)
[1] 1
```

طول إطار بيانات هو عدد المتغيرات التي يحويها الإطار. يمكن إيجاد طول دالة ولكن ليس له معنى. الدالة `mode(...)` تعطي طور الشيء بينما الدالة `attributes(...)` تعطي الصفات غير الجوهرية

```
> mode(sheep2)
[1] "list"
> attributes(sheep2)
$names
[1] "weight" "height"
$class
[1] "data.frame"
$row.names
```

```

[1] "1" "2" "3" # ... etc to "100"
> mode(plot)
[1] "function"
> attributes(plot)
NULL
> mode(lmres)
[1] "list"
> attributes(lmres)
$names
[1] "coefficients" "residuals" "effects" "rank"
[5] "fitted.values" "assign" "qr" "df.residual"
[9] "xlevels" "call" "terms" "model"
$class
[1] "lm"

```

لاحظ برغم أن كلا منهما من طور *list* إلا أن إطارات البيانات من فئة *data.frame* بينما شئى النموذج الخطي فهو من فئة *lm*. وهذا يؤثر على الطريقة التي يظهر بها على الشاشة عند طباعة المعرف

```

> sheep2
  weight height
1   75.19  56.34
2  101.80  83.57
3   74.51  56.58
.
.

```

```
> lmres
```

```
Call:
lm(formula = weight ~ height)
```

```
Coefficients:
(Intercept)      height
  26.0319         0.8724
```

لقد تمت طباعة إطار البيانات كليا بينما اعطي مخرجات قليلة لشئى نموذج خطي. نستطيع أن نجبر R لكي يعامل كليهما كقائمة *list* باستخدام الدالة `unclass(...)` ولنحاول التالي:

```
> unclass(sheep2)
```



\$weight

```
[1] 75.19 101.80 74.51 81.42 83.20 80.84 81.67 90.82 102.51 104.87 96.03
[12] 86.97 87.08 83.92 69.00 67.57 81.89 98.79 80.74 61.61 84.64 72.29
[23] 82.41 80.20 73.82 64.95 70.64 94.60 72.08 65.60 97.95 61.97 84.81
[34] 80.21 73.18 108.50 79.46 76.84 74.10 78.72 59.80 76.92 85.28 96.27
[45] 82.77 108.00 97.90 77.99 91.60 97.37 78.40 107.93 89.74 83.57 65.76
[56] 90.13 73.25 77.18 95.66 105.70 85.66 94.22 104.60 72.03 65.99 81.16
[67] 93.84 86.07 95.56 89.84 55.90 85.40 89.59 72.07 64.76 68.09 86.05
[78] 81.95 75.62 99.48 84.30 81.49 79.84 72.53 119.50 83.53 70.92 58.87
[89] 96.47 63.89 87.33 81.01 97.90 79.97 69.10 68.37 71.81 85.68 94.27
[100] 71.90
```

\$height

```
[1] 56.34 83.57 56.58 55.60 73.59 56.24 65.93 70.42 66.28 73.99 72.76 69.87 65.33
[14] 64.40 59.20 54.59 66.53 68.67 68.27 54.06 67.66 60.38 68.54 62.96 60.48 53.20
[27] 64.56 68.01 54.76 62.47 77.62 46.20 58.23 73.84 65.58 88.21 52.42 72.40 60.17
[40] 65.30 57.55 49.76 65.82 70.41 54.89 82.67 70.00 67.91 66.26 73.51 73.42 82.56
[53] 70.27 63.01 40.57 72.50 61.38 60.98 58.64 57.91 81.64 72.12 84.47 57.33 59.14
[66] 61.51 74.59 45.81 82.23 78.81 57.40 61.60 73.24 68.65 51.60 60.55 66.49 74.25
[79] 50.03 60.82 73.89 65.74 60.50 52.72 90.05 54.40 60.25 62.46 70.50 45.35 71.46
[92] 69.82 79.17 57.46 55.67 50.15 78.97 64.84 81.64 49.11
```

attr(,"row.names")

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
[21] 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40
[41] 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
[61] 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
[81] 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
>
```

> unclass(lmres)

\$coefficients

```
(Intercept)      height
26.0318530      0.8723918
```

\$residuals

```
1 2 3 4 5 6
0.007594467 2.862366459 -0.881779559 6.883164379 -7.031163636 5.744833644
7 8 9 10 11 12
-1.878642646 3.354318288 18.656020233 14.289879655 6.522921537 -0.015866236
13 14 15 16 17 18
4.054792418 1.706116768 -8.677446007 -6.085719929 -2.182077711 12.851003893
19 20 21 22 23 24
-4.850039398 -11.583352289 -0.417880416 -6.416868300 -3.415585177 -0.757639078
25 26 27 28 29 30
-4.974107478 -7.493095363 -11.713465916 9.236782464 -1.724026530 -14.930167108
31 32 33 34 35 36
4.203097515 -4.366352945 7.978774014 -10.239261579 -10.063305525 5.514468628
37 38 39 40 41 42
7.697370221 -12.353017424 -4.423666028 -4.279035829 -16.437999580 7.477932340
43 44 45 46 47 48
1.827320449 8.813042206 8.852562539 9.847519056 10.800722833 -7.285978359
49 50 51 52 53 54
7.763468068 7.208627706 -11.682857034 9.873482151 2.405177054 2.568741334
55 56 57 58 59 60
4.335212743 0.849743398 -6.329260075 -2.050303365 18.471093386 29.147939381
61 62 63 64 65 66
-11.593917417 5.271252272 4.877213863 -4.016073390 -11.635102501 1.467328995
67 68 69 70 71 72
2.736444590 20.073879847 -2.208628563 -4.945048696 -20.207140814 5.628813735
73 74 75 76 77 78
-0.335826515 -13.851548272 -6.287268524 -10.765174902 2.012817960 -8.856942206
79 80 81 82 83 84
5.942386561 20.389279319 -6.192881168 -1.892888209 1.028444687 0.505652689
```

	85	86	87	88	89	90
14.909267764	10.040034508	-7.673457370	-21.651443190	8.934526946	-1.704819937	
	91	92	93	94	95	96
-1.042969157	-5.932246647	2.800890265	3.810515680	-5.497903045	-1.412300452	
	97	98	99	100		
-23.114631380	3.082264387	-2.983917417	3.024986993			

\$effects

	(Intercept)		height			
-827.5180000	-88.1319145	-0.7006440	7.0875994	-7.2544388	5.9340527	
-1.9198022	3.2064095	18.6065394	14.0570945	6.3193795	-0.1506989	
4.0278978	1.7013328	-8.5586006	-5.8572723	-2.2375022	12.7447011	
-4.9468322	-11.3423040	-0.5001706	-6.3260773	-3.5187972	-0.7281873	
-4.8856939	-7.2316007	-11.7220539	9.1461711	-1.4996206	-14.8890656	
3.8840095	-3.9384340	8.1206810	-10.4684805	-10.0961439	4.9436046	
7.9774093	-12.5480005	-4.3278823	-4.3052172	-16.2799256	7.8212127	
1.7887761	8.6653711	9.0738777	9.4083679	10.6627995	-7.3742122	
7.7144628	6.9872545	-11.9020905	9.4369462	2.2608345	2.5970044	
4.8969843	0.6523828	-6.2622439	-1.9737773	18.6032527	29.2974544	
-12.0085805	5.0829261	4.3952679	-3.8527690	-11.5148306	1.5312544	
2.4893945	20.5110710	-2.6373188	-5.2924288	-20.0455006	5.6905994	
-0.5507805	-13.9573755	-5.9877340	-10.6784256	1.9583444	-9.0959088	
6.2792477	20.4696094	-6.4232888	-1.9295306	1.1163827	0.7785593	
14.2946579	10.2729994	-7.5795756	-21.6101039	8.7847161	-1.2566924	
-1.2156039	-6.0658905	2.4449512	3.9707294	-5.2951323	-1.0782923	
-23.4658155	3.0670194	-3.3985805	3.3837210			

\$rank

[1] 2

\$fitted.values

	1	2	3	4	5	6	7	8
75.18241	98.93763	75.39178	74.53684	90.23116	75.09517	83.54864	87.46568	
	9	10	11	12	13	14	15	16
83.85398	90.58012	89.50708	86.98587	83.02521	82.21388	77.67745	73.65572	
	17	18	19	20	21	22	23	24
84.07208	85.93900	85.59004	73.19335	85.05788	78.70687	85.82559	80.95764	
	25	26	27	28	29	30	31	32
78.79411	72.44310	82.35347	85.36322	73.80403	80.53017	93.74690	66.33635	
	33	34	35	36	37	38	39	40
76.83123	90.44926	83.24331	102.98553	71.76263	89.19302	78.52367	82.99904	
	41	42	43	44	45	46	47	48
76.23800	69.44207	83.45268	87.45696	73.91744	98.15248	87.09928	85.27598	
	49	50	51	52	53	54	55	56
83.83653	90.16137	90.08286	98.05652	87.33482	81.00126	61.42479	89.28026	
	57	58	59	60	61	62	63	64
79.57926	79.23030	77.18891	76.55206	97.25392	88.94875	99.72279	76.04607	
	65	66	67	68	69	70	71	72
77.62510	79.69267	91.10356	65.99612	97.76863	94.78505	76.10714	79.77119	
	73	74	75	76	77	78	79	80
89.92583	85.92155	71.04727	78.85517	84.03718	90.80694	69.67761	79.09072	

	81	82	83	84	85	86	87	88
90.49288	83.38289	78.81156	72.02435	104.59073	73.48997	78.59346	80.52144	
	89	90	91	92	93	94	95	96
87.53547	65.59482	88.37297	86.94225	95.09911	76.15948	74.59790	69.78230	
	97	98	99	100				
94.92463	82.59774	97.25392	68.87501					

\$assign

[1] 0 1

\$qr

\$qr

	(Intercept)	height
1	-10.0	-6.501660e+02
2	0.1	-1.010233e+02
3	0.1	-7.570349e-02
4	0.1	-8.540422e-02
5	0.1	9.267347e-02
6	0.1	-7.906905e-02
7	0.1	1.684939e-02
8	0.1	6.129458e-02
9	0.1	2.031394e-02
10	0.1	9.663295e-02
11	0.1	8.445755e-02
12	0.1	5.585029e-02
13	0.1	1.091017e-02
14	0.1	1.704377e-03
15	0.1	-4.976889e-02
16	0.1	-9.540191e-02
17	0.1	2.278862e-02
18	0.1	4.397184e-02
19	0.1	4.001236e-02
20	0.1	-1.006482e-01
21	0.1	3.397415e-02
22	0.1	-3.808841e-02
23	0.1	4.268501e-02
24	0.1	-1.254976e-02
25	0.1	-3.709854e-02
26	0.1	-1.091611e-01
27	0.1	3.288170e-03
28	0.1	3.743870e-02
29	0.1	-9.371913e-02
30	0.1	-1.740012e-02
31	0.1	1.325652e-01
32	0.1	-1.784520e-01
33	0.1	-5.937063e-02
34	0.1	9.514815e-02
35	0.1	1.338485e-02
36	0.1	2.373925e-01
37	0.1	-1.168821e-01
38	0.1	8.089401e-02
39	0.1	-4.016714e-02
40	0.1	1.061321e-02
41	0.1	-6.610175e-02
42	0.1	-1.432127e-01
43	0.1	1.576054e-02
44	0.1	6.119559e-02
45	0.1	-9.243230e-02
46	0.1	1.825537e-01
47	0.1	5.713712e-02
48	0.1	3.644883e-02
49	0.1	2.011597e-02
50	0.1	9.188157e-02
51	0.1	9.099069e-02
52	0.1	1.814648e-01
53	0.1	5.980977e-02
54	0.1	-1.205482e-02

```
55      0.1 -2.341818e-01
56      0.1  8.188388e-02
57      0.1 -2.818971e-02
58      0.1 -3.214919e-02
59      0.1 -5.531216e-02
60      0.1 -6.253821e-02
61      0.1  1.723580e-01
62      0.1  7.812238e-02
63      0.1  2.003714e-01
64      0.1 -6.827946e-02
65      0.1 -5.036281e-02
66      0.1 -2.690288e-02
67      0.1  1.025722e-01
68      0.1 -1.823125e-01
69      0.1  1.781983e-01
70      0.1  1.443447e-01
71      0.1 -6.758655e-02
72      0.1 -2.601200e-02
73      0.1  8.920892e-02
74      0.1  4.377387e-02
75      0.1 -1.249990e-01
76      0.1 -3.640564e-02
77      0.1  2.239267e-02
78      0.1  9.920662e-02
79      0.1 -1.405400e-01
80      0.1 -3.373299e-02
81      0.1  9.564308e-02
82      0.1  1.496864e-02
83      0.1 -3.690057e-02
84      0.1 -1.139125e-01
85      0.1  2.556061e-01
86      0.1 -9.728267e-02
87      0.1 -3.937525e-02
88      0.1 -1.749911e-02
89      0.1  6.208647e-02
90      0.1 -1.868659e-01
91      0.1  7.158923e-02
92      0.1  5.535336e-02
93      0.1  1.479082e-01
94      0.1 -6.699263e-02
95      0.1 -8.471131e-02
96      0.1 -1.393522e-01
97      0.1  1.459285e-01
98      0.1  6.059807e-03
99      0.1  1.723580e-01
100     0.1 -1.496468e-01
```

```
attr(,"assign")
```

```
[1] 0 1
```

```
$qraux
```

```
[1] 1.100000 1.191463
```

```
$pivot
```

```
[1] 1 2
```

```
$tol
```

```
[1] 1e-07
```

```
$rank
```

```
[1] 2
```

```

attr(,"class")
[1] "qr"

$df.residual
[1] 98

$xlevels
list()

$call
lm(formula = weight ~ height)

$terms
weight ~ height
attr(,"variables")
list(weight, height)
attr(,"factors")
      height
weight      0
height      1
attr(,"term.labels")
[1] "height"
attr(,"order")
[1] 1
attr(,"intercept")
[1] 1
attr(,"response")
[1] 1
attr(,".Environment")
<environment: R_GlobalEnv>
attr(,"predvars")
list(weight, height)
attr(,"dataClasses")
      weight      height
"numeric" "numeric"

$model
      weight height
1    75.19  56.34
2   101.80  83.57
3    74.51  56.58
4    81.42  55.60

```

5	83.20	73.59
6	80.84	56.24
7	81.67	65.93
8	90.82	70.42
9	102.51	66.28
10	104.87	73.99
11	96.03	72.76
12	86.97	69.87
13	87.08	65.33
14	83.92	64.40
15	69.00	59.20
16	67.57	54.59
17	81.89	66.53
18	98.79	68.67
19	80.74	68.27
20	61.61	54.06
21	84.64	67.66
22	72.29	60.38
23	82.41	68.54
24	80.20	62.96
25	73.82	60.48
26	64.95	53.20
27	70.64	64.56
28	94.60	68.01
29	72.08	54.76
30	65.60	62.47
31	97.95	77.62
32	61.97	46.20
33	84.81	58.23
34	80.21	73.84
35	73.18	65.58
36	108.50	88.21
37	79.46	52.42
38	76.84	72.40
39	74.10	60.17
40	78.72	65.30
41	59.80	57.55
42	76.92	49.76
43	85.28	65.82
44	96.27	70.41
45	82.77	54.89
46	108.00	82.67
47	97.90	70.00
48	77.99	67.91
49	91.60	66.26
50	97.37	73.51
51	78.40	73.42
52	107.93	82.56
53	89.74	70.27
54	83.57	63.01
55	65.76	40.57
56	90.13	72.50
57	73.25	61.38
58	77.18	60.98
59	95.66	58.64
60	105.70	57.91
61	85.66	81.64
62	94.22	72.12
63	104.60	84.47
64	72.03	57.33
65	65.99	59.14
66	81.16	61.51
67	93.84	74.59
68	86.07	45.81
69	95.56	82.23
70	89.84	78.81
71	55.90	57.40
72	85.40	61.60
73	89.59	73.24
74	72.07	68.65
75	64.76	51.60

```

76 68.09 60.55
77 86.05 66.49
78 81.95 74.25
79 75.62 50.03
80 99.48 60.82
81 84.30 73.89
82 81.49 65.74
83 79.84 60.50
84 72.53 52.72
85 119.50 90.05
86 83.53 54.40
87 70.92 60.25
88 58.87 62.46
89 96.47 70.50
90 63.89 45.35
91 87.33 71.46
92 81.01 69.82
93 97.90 79.17
94 79.97 57.46
95 69.10 55.67
96 68.37 50.15
97 71.81 78.97
98 85.68 64.84
99 94.27 81.64
100 71.90 49.11

```

>

لاحظ أن الجزء الأخير من `unclass(lmres)` يشبه مخرجات `sheep2` وذلك لأن `lmres` يحوي صفة (`$model`) والذي هو إطار بيانات مكون من متغيرات نحن نضعها في النموذج. نستطيع إيجاد فئة أي شيء مستخدمين الدالة `class(...)`

```
> class(lmres$model)
```

```
[1] "data.frame"
```

وهذه ليست النهاية لأن الطور `mode` هو طريقة مريحة لإخفاء حقيقة معقدة. في الحقيقة R تعمل على نوع `type` الشيء وفي بعض الأحيان يكون نوع أكثر تحديدا ودقة من طور `mode` ولكن في حالات أخرى فإن الخاصيتين متطابقتان. نستطيع إيجاد نوع شيء باستخدام الدالة `typeof(...)` والطريقة التي تخزن بها تعطى بالدالة `storage.mode(...)` كالتالي:

```
> typeof(sheep2)
```

```
[1] "list"
```

```
> typeof(sheep2$weight)
```

```
[1] "double"
```

```
> z <- 1 + 1i
```

```
> typeof(z)
```

```
[1] "complex"
```

```
> typeof(plot)
```

```
[1] "closure"
```

```
> storage.mode(plot)
```

```
[1] "function"
```

بعض الدوال:

```
> class(lmres$model)
[1] "data.frame"
> names(sheep2)
[1] "weight" "height"
> attributes(lmres)
$names
 [1] "coefficients" "residuals"      "effects"
"rank"          "fitted.values"
 [6] "assign"        "qr"              "df.residual"
"xlevels"       "call"
[11] "terms"         "model"

$class
[1] "lm"

> lmres$residuals
      1          2          3          4
5      0.007594467  2.862366459 -0.881779559  6.883164379 -
7.031163636  5.744833644
      7          8          9         10
11     -1.878642646  3.354318288  18.656020233  14.289879655
6.522921537 -0.015866236 ....

> lmres$effects
(Intercept)      height
-827.5180000 -88.1319145 -0.7006440  7.0875994 -7.2544388
5.9340527

-1.9198022  3.2064095  18.6065394  14.0570945  6.3193795
-0.1506989

4.0278978  1.7013328 -8.5586006 -5.8572723 -2.2375022
12.7447011 ...

> lmres$coefficients
(Intercept)      height
26.0318530  0.8723918
```

## :Named arguments and default values الحجاج المسماه والقيم الافتراضية

في التالي نحتاج إلى تحميل بعض الملفات من دليل العمل

```
> setwd("C:/orCourse")
> mkmsc <- read.table("marks.dat", header=TRUE)
```



```
> mk2nd <- read.table("marks2.dat", header=TRUE)
> attach(mk2nd)
```

في لغة R الدوال التالية

```
> plot(exam1, exam2)
> plot(exam2, exam1)
```

الاولى ترسم exam1 على محور x و exam 2 على محور y بينما الثانية تعكس الرسم وهذا هو الإختيار الافتراضي default في R. لتغيير الإختيار الافتراضي نستخدم حجج مسماة *named arguments* كالتالي

```
> plot(y=exam2, x=exam1)
> plot(x=exam1, y=exam2)
```

وكلاهما يعطي الرسم نفسه. الحجج المسماه تفيد أكثر في عدم الإلتزام بالترتيب أو القيم الافتراضية فمثلا

```
> t.test(exam1)
> t.test(exam1, mu=30)
> t.test(exam1, alternative="greater")
> t.test(exam1, mu=31.5, alternative="less")
> t.test(mu=30, exam1)
> t.test(x=exam1, mu=30)
```

لاحظ أن القيمة الافتراضية لـ mu في t.test هي الصفر. و alternative = "two.sided"

### الصفة names:

كما ذكرنا سابقا إطارات البيانات لها صفة *names* والتي تشير إلى أسماء المتغيرات في ذلك الإطار. ويمكن سرد الأسماء بإستخدام الدالة `names(...)`

```
> names(mkmsc)
[1] "courseA" "courseB" "courseC"
```

الغرض من العامل \$ هو تمكيننا من الوصول مباشرة إلى جزء- شئ *sub-object* مسرد بواسطة `names(...)` ويمكننا تغيير الأسماء والمحتويات للجزء - شئ عن طريق الإسناد فلنفترض اننا نريد تغيير `courseA` و `courseB` و `courseC` بالأسماء التالية `or221` و `or241` و `or342`

```
> names(mkmsc) <- c("or221", "or241", "or342")
```

```
> names(mkmsc)
```

```
[1] "or221" "or241" "or342"
```

```
> mkmsc$or221
```

```
[1] 52 71 44 90 23 66
```

ونستطيع إضافة 2 لكل قيمة

```
> mkmsc$or221 <- mkmsc$or221 + 2
```

```
> mkmsc$or221
```

```
[1] 54 73 46 92 25 68
```

## صفات غير جوهرية أخرى Other non-intrinsic attributes:

الصفات غير الجوهرية الأخرى يمكن تغييرها بالإسناد فمثلا في إطار بيانات نريد تغيير أسماء الأسطر الافتراضية والتي هي غالبا 1 و 2 و 3 و الخ إلى أسماء ذات معنى

```
> row.names(mkmsc)
[1] "1" "2" "3" "4" "5" "6"
> row.names(mkmsc) <- c("Ali", "Badr", "Ahmad",
+"Bakur", "Saeed", +"Faris")
> mkmsc
      or221 or241 or342
Ali      54    67    71
Badr     73    80    84
Ahmad    46    55    55
Bakur    92    76    68
Saeed    25    49    52
Faris    68    61    61
>
```

ملاحظة: في R يكمل السطر الطويل إلى السطر التالي بوضع علامة + في بداية السطر التالي.

## المتتابعات المنسقة Regular sequences:

المتتابعات المنسقة هي متتابعات من الأعداد أو الحروف والتي تتبع نمط أو نسق ثابت وهي مفيدة في إختيار أجزاء من متجة وفي توليد قيم للمتغيرات الوصفية. هناك عدة طرق لتوليد متتابعات منها مولد متتابعة *sequence generator*:

```
> 1:10
[1] 1 2 3 4 5 6 7 8 9 10
> 10:1
[1] 10 9 8 7 6 5 4 3 2 1
> 2*1:10
[1] 2 4 6 8 10 12 14 16 18 20
> 1:10 + 1:20
[1] 2 4 6 8 10 12 14 16 18 20 12 14 16 18 20 22 24
26 28 30
> 1:10-1
[1] 0 1 2 3 4 5 6 7 8 9
> 1:(10-1)
```

```
[1] 1 2 3 4 5 6 7 8 9
```

لاحظ أن ( : ) له افضلية على العملية الحسابية.  
الدالة seq تعطي مقدار اكبر من المرونة في توليد متتابعات ولها البناء التالي

**seq(from, to, by, length, along)**

الحجج from و to مفسرة ذاتيا الحجة by تدل على مقدار الزيادة و length عدد الوحدات في المتابعة. لاحظ انه لا تستخدم جميع هذه الحجج معا وإلا سيسبب هذا في زيادة في التعيين وينتج عنه خطأ. الحجة along تسمح لمتجه يذكر والذي طوله هو الطول المطلوب للمتابعة

```
> seq(1,10)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(to=10, from=1)
[1] 1 2 3 4 5 6 7 8 9 10
> seq(1,10,by=0.5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5
7.0 7.5 8.0
[16] 8.5 9.0 9.5 10.0
> seq(1,10,length=19)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5
7.0 7.5 8.0
[16] 8.5 9.0 9.5 10.0
> seq(1,10,length=19,by=0.25)
Error in seq.default(1, 10, length = 19, by = 0.25)
:
Too many arguments
> seq(1,by=2,length=6)
[1] 1 3 5 7 9 11
> seq(to=30,length=13)
[1] 18 19 20 21 22 23 24 25 26 27 28 29 30
> seq(to=30)
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
20 21 22 23 24 25
[26] 26 27 28 29 30
```

الدالة rep تولد متتابعات بتكرارات معينة

```
> rep(1, times = 3)
[1] 1 1 1
> rep((1:3), each =5)
[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
```

## المتجهات المنطقية Logical vectors

القيم المنطقية في R هي TRUE و FALSE  
المقارنة تعمل بإستخدام == للمساواة و != لعدم المساواة و > لأكبر من و >= لأكبر من  
أو يساوي و < لأقل من و <= لأقل من أو يساوي  
العمال المنطقية & لـ AND و | لـ OR و ! لـ NOT

```
> tf1 <- c(TRUE,TRUE,FALSE,FALSE)
> tf2 <- c(TRUE,FALSE,TRUE,FALSE)
> tf1 & tf2
[1] TRUE FALSE FALSE FALSE
> tf1 | tf2
[1] TRUE TRUE TRUE FALSE
> (tf1 & !tf2) | (!tf1 & tf2)
[1] FALSE TRUE TRUE FALSE
```

## تأشير متجه وإختيار مجموعة جزئية Indexing vectors and subset :selection

أحيانا نرغب في إستخدام جزء من البيانات أو مجموعة جزئية والتي تحقق شرط معين  
ولذلك تستخدم [ ] لكي توضح لجزء من المتجه ويعرف بعامل التذييل *subscripting*  
*operator* والمتجه ما بين القوسين يعرف بإسم متجه التأشير *indexing vector* فإذا  
كان متجه التأشير له قيم صحيحة (غالبا متتابعة متسقة) فإن العناصر المؤشرة التابعة  
لقيم المتجه المؤشر يتم إختيارها وإذا وجدت علامه - في المقدمة فعندئذ كل العناصر  
ماعدى الذين تم تأشيرهم سيتم إختيارهم

```
> attach(mkmsc)
> or221
[1] 54 73 46 92 25 68
> or221[c(1,2,6)]
[1] 54 73 68
> or221[1:4]
[1] 54 73 46 92
> or221[-(1:4)]
[1] 25 68
> or221[seq(6,by=-2)]
[1] 68 92 73
```

```
> or221[rep((1:3),each=2)]
[1] 54 54 73 73 46 46
> row.names(mkmsc)[1:3]
[1] "Ali" "Badr" "Ahmad"
```

نلاحظ أننا نستطيع عمل التالي:

- إيجاد علامات الطلاب في or221 التي أعلى من 70
- علامات طلاب or221 والذين حصلوا على أقل من 65 في or241
- علامات الطلاب الذين مجموع علاماتهم الكلي أقل من 200

```
> or221[or221>70]
[1] 73 92
> or241[or221<65]
[1] 67 55 49
> or342[(or221+or241+or342)<200]
[1] 71 55 52 61
> or342[or221>50 & or241>50 & or342>50]
[1] 71 84 68 61
```

التالي يستخرج الأسماء بدل العلامات

```
> row.names(mkmsc)[or221>70]
[1] "Badr" "Bakur"
> row.names(mkmsc)[or241<50 | or221<50 | or342<50]
[1] "Ahmad" "Saeed"
> names(mkmsc)[c(sum(or221), sum(or241),
sum(or342)) > 350]
[1] "or221" "or241" "or342"
```

### :Character sequences أو النصوص متتابعات الحروف

لنفترض أنه لدينا أزواج من النقاط  $\{(x_1, y_1), (x_2, y_2), \dots, (x_5, y_5)\}$  ونريد تمثيلها في R لذلك نستخدم الدالة `paste(...)` والتي تجمع الحجج وتعيد حرف `character`.  
الحجة `sep` لتحديد ما نريد إدخاله بين الحجج

```
> paste(c("x","y"), rep(1:5,each=2), sep="")
[1] "x1" "y1" "x2" "y2" "x3" "y3" "x4" "y4" "x5"
"y5"
> str1 <- paste("(x", 1:5, sep="")
```

```

> str2 <- paste("y", 1:5, ")", sep="")
> label1 <- paste(str1,str2,sep="," )
> label1
[1] "(x1,y1)" "(x2,y2)" "(x3,y3)" "(x4,y4)"
"(x5,y5)"
> mode(label1)
[1] "character"

```

## :القوائم Lists

القائمة هي مجموعة مرتبة من الأشياء. الأشياء الموجودة في قائمة يشار إليها كمكونات *components* وهذه المكونات قد تكون من أطوار مختلفة. تولد القوائم بإستخدام الدالة `list(...)` والقيمة المرجعة هي قائمة بمكونات من الحجج التي اعطيت للقائمة

```

> mklst <- list("BSc exam marks", c(10,6,2005),
mkmsc)
> mode(mklst)
[1] "list"
> length(mklst)
[1] 3
> mklst
[[1]]
[1] "BSc exam marks"

[[2]]
[1] 10 6 2005

[[3]]
      or221 or241 or342
Ali      54    67    71
Badr     73    80    84
Ahmad    46    55    55
Bakur    92    76    68
Saeed    25    49    52
Faris    68    61    61

```

نستطيع الآن إستخدام عامل التأشير [ ] للوصول إلى أجزاء من القائمة

```

> mklst[2]

```

```
[[1]]  
[1] 10 6 2005
```

```
> mode(mk1st[2])  
[1] "list"  
> length(mk1st[2])  
[1] 1
```

لاحظ أن نتيجة استخدام التأشير على قائمة هو قائمة أخرى. إذا أردنا أن نضيف 4 إلى يوم فإننا لا نستطيع أن نستخدم [ ] لأن mk1st[2] ليست قيمة عددية

```
> mk1st[2]+c(4,0,0)  
Error in mk1st[2] + c(4, 0, 0) : non-numeric  
argument to binary operator
```

ولكي نتجنب هذا الخطأ نستخدم أقواس مضاغفة [ ] مثل

```
> mk1st[[2]]  
[1] 10 6 2005  
> mode(mk1st[[2]])  
[1] "numeric"  
> mk1st[[2]] <- mk1st[[2]] + c(4,0,0)  
> mk1st
```

في الحقيقة بما أن mk1st[[2]] متجه عددي فإننا نستطيع استخدام أي عنصر فيه مباشرة ونضيف 4 إليه

```
> mk1st[[2]][1] <- mk1st[[2]][1] + 4
```

لاحظ أن العامل [ ] إذا طبق على قائمة فإنه يعيد قائمة والعامل [ ] يعيد عنصر من القائمة وطور القيمة المعادة هو طور العنصر. العلامات label الافتراضية لعناصر قائمة هي 1 و 2 و 3 و ... ولا توضح ماتحويه القائمة ويكون علينا تذكر اسم كل عنصر في القائمة وهذا سهلا لقوائم تحوي عنصر أو عنصرين ولكن للقوائم التي تحوي عدد كبير من العناصر يكون من الصعب تذكر أسماء كل العناصر وترتيبها في القائمة لهذا من الأفضل إعطاء أسماء لعناصر القائمة مثلا كالتالي

```
> names(mk1st) <- c("title", "date", "marks")
```

ويمكننا الآن الإشارة إلى عناصر القائمة إما باستخدام العامل [ ] أو العامل \$



```

> mklst[[2]]
[1] 18 6 2005
> mklst[["date"]]
[1] 18 6 2005
> mklst$date
[1] 18 6 2005

```

لاحظ باستخدامنا للعامل \$ نستطيع الإشارة إلى إسم مباشرة ولانحتاج لإستخدام "" . كما يمكن إعطاء الأسماء عند تعريف القائمة مثل

```

> mklst <- list("title"="MSc exam marks",
+"date"=c(10,6,2005),
+ "marks"=mkmsc)

```

### الرسومات البسيطة في R:

تمتلك R مقدرات رسم قوية ومرنة. سوف نستعرض هنا بعضا من هذه المقدرات.

#### بعض الرسومات الوصفية:

سوف نقوم برسم المدرج التكراري لأحد المتغيرات وذلك باستخدام الدالة hist(...)  
كالتالي:

أولا نحدد دليل العمل ونقرأ ملف البيانات

```

> setwd("C:/orCourse")
> mk2nd <- read.table("marks2.dat", header=TRUE)
> attributes(mk2nd)
$names
[1] "candno" "exam1" "exam2" "exam3"

$class
[1] "data.frame"

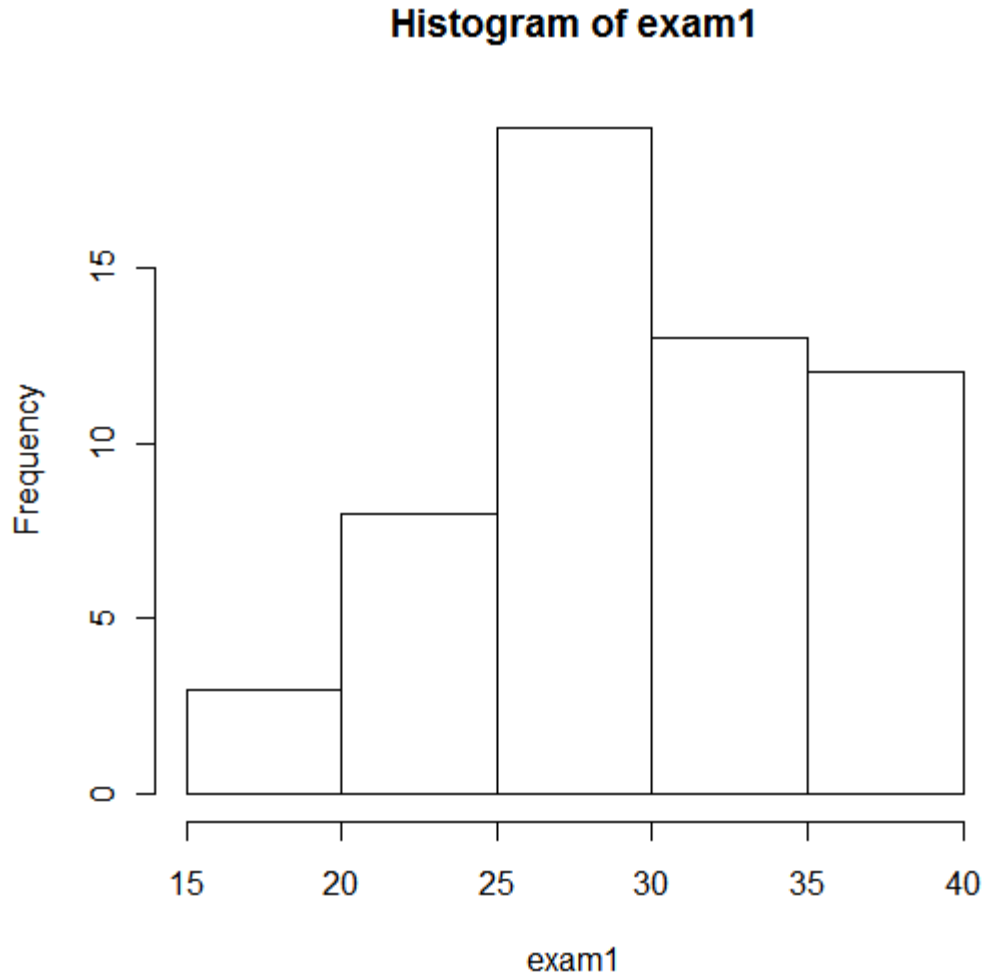
$row.names
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
16 17 18 19 20 21 22
[23] 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44
[45] 45 46 47 48 49 50 51 52 53 54 55

```

```
> attach(mk2nd)
```

ونرسم المدرج التكراري للمتغير exam1

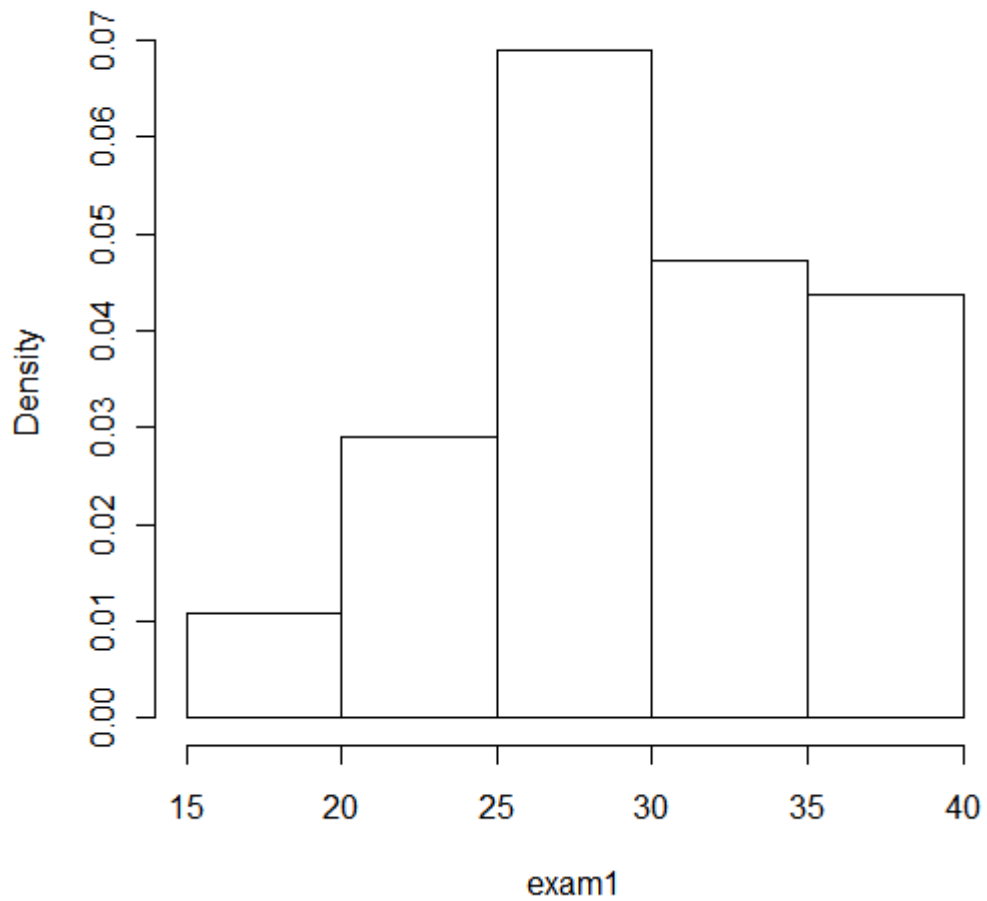
```
> hist(exam1)
```



لاحظ هنا تم رسم التكرارات. لرسم الإحتمالات ندخل

```
> hist(exam1, probability = TRUE)
```

## Histogram of exam1



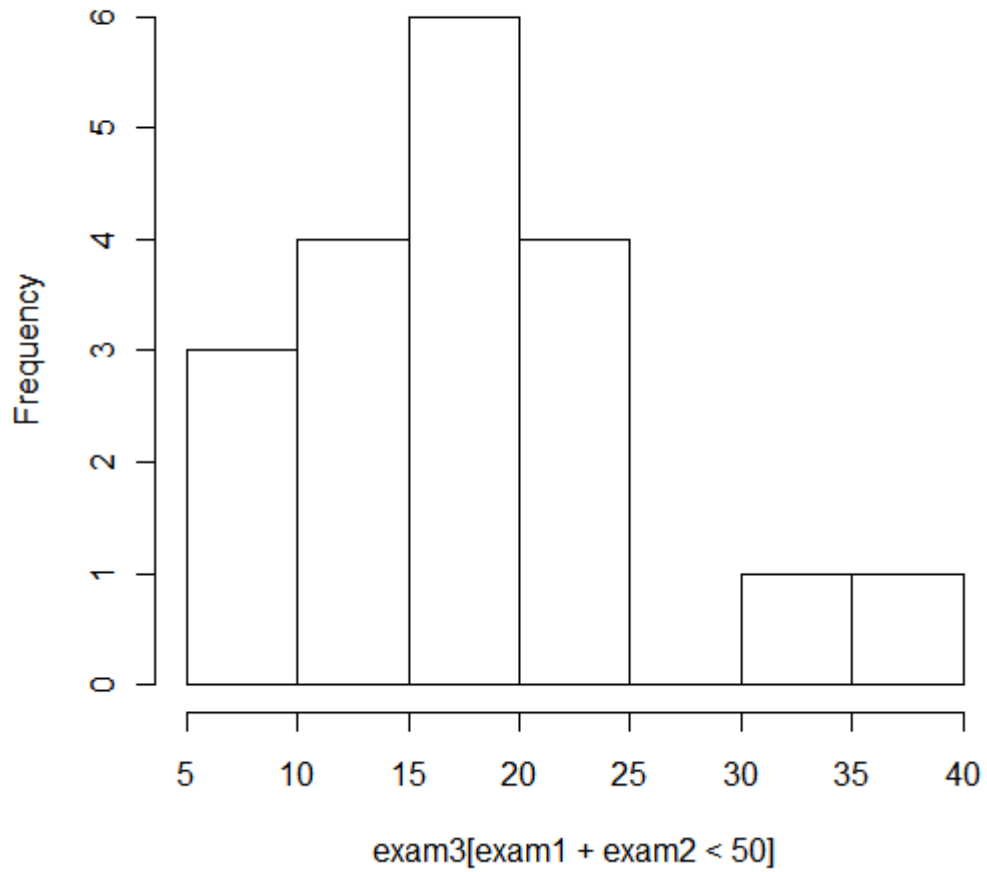
نريد فئات أكثر مثلا 10 فئات

```
> hist(exam1, nclass=10)
```

كما أننا نستطيع إختيار أجزاء كالتالي

```
> hist(exam3[exam1+exam2 < 50])
```

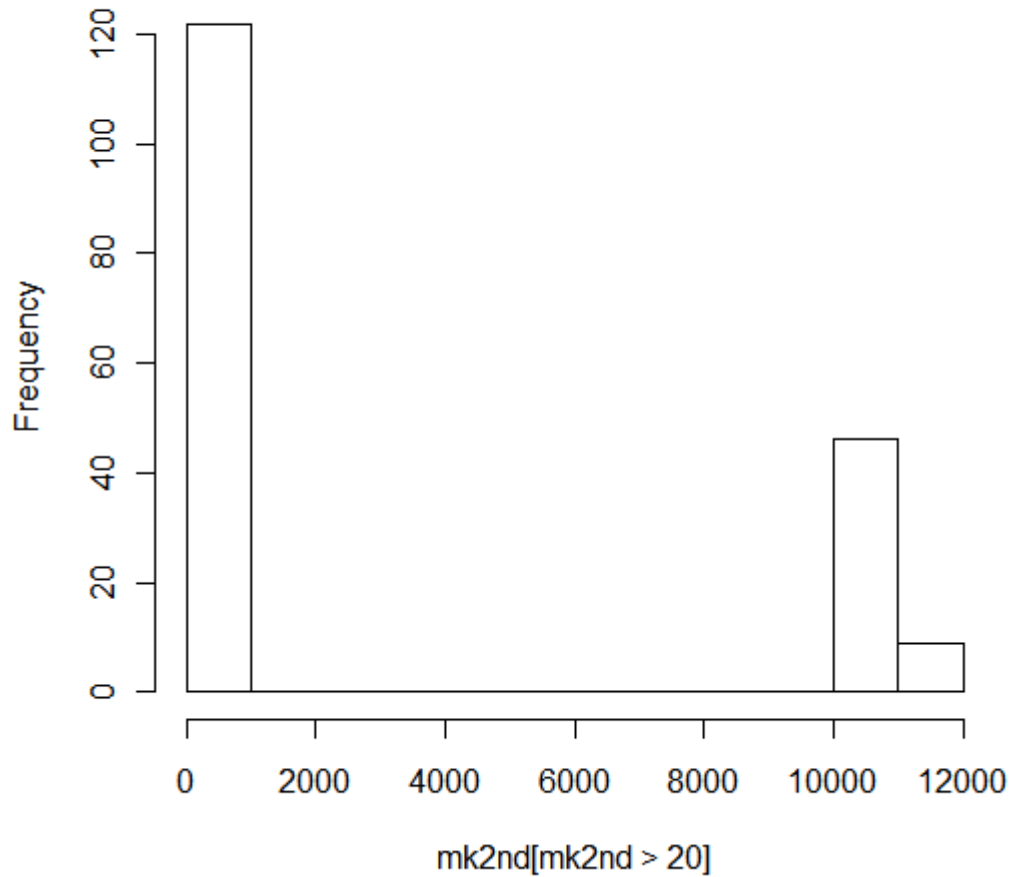
**Histogram of exam3[exam1 + exam2 < 50]**



```
> hist(mk2nd[mk2nd>20])
```

أو

Histogram of mk2nd[mk2nd > 20]



لاحظ أننا لانستطيع رسم إطار بيانات مباشرة

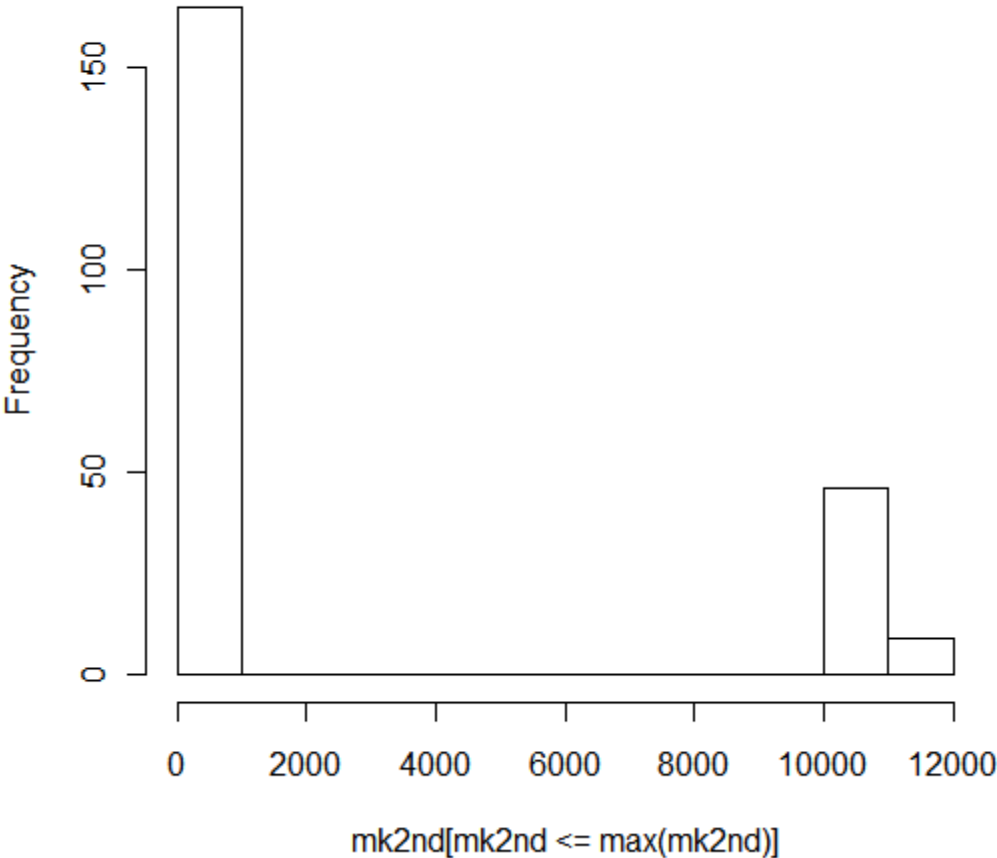
```
> hist(mk2nd)
```

```
Error in hist.default(mk2nd) : `x' must be numeric
```

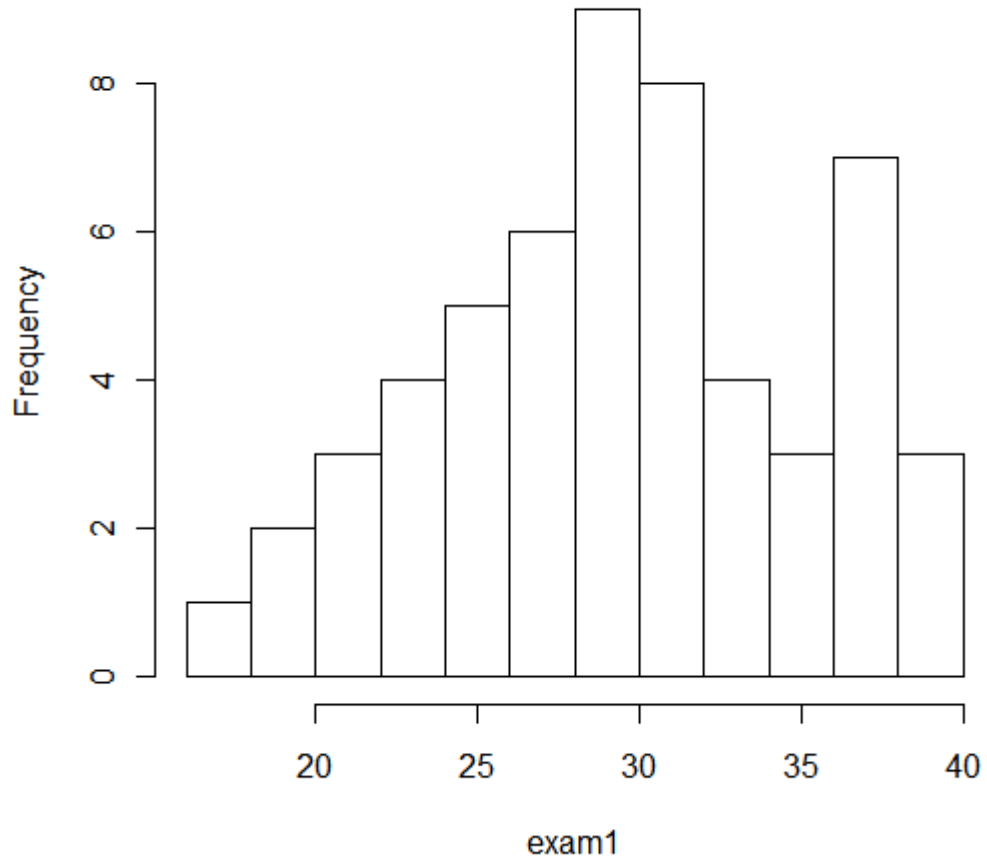
ولكن نستطيع التالي

```
> hist(mk2nd[mk2nd<=max(mk2nd)])
```

**Histogram of mk2nd[mk2nd <= max(mk2nd)]**



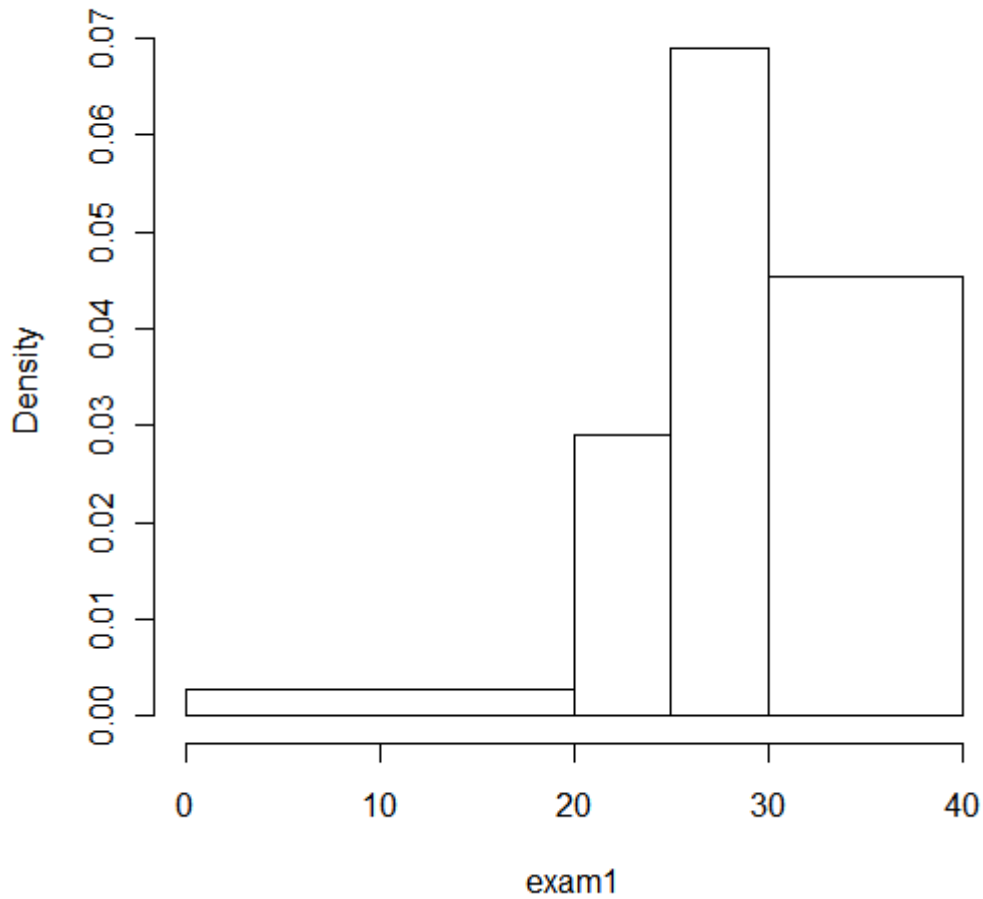
**Histogram of exam1**



تغير حدود الفئات

```
> hist(exam1, breaks=c(0,20,25,30,40))
```

## Histogram of exam1

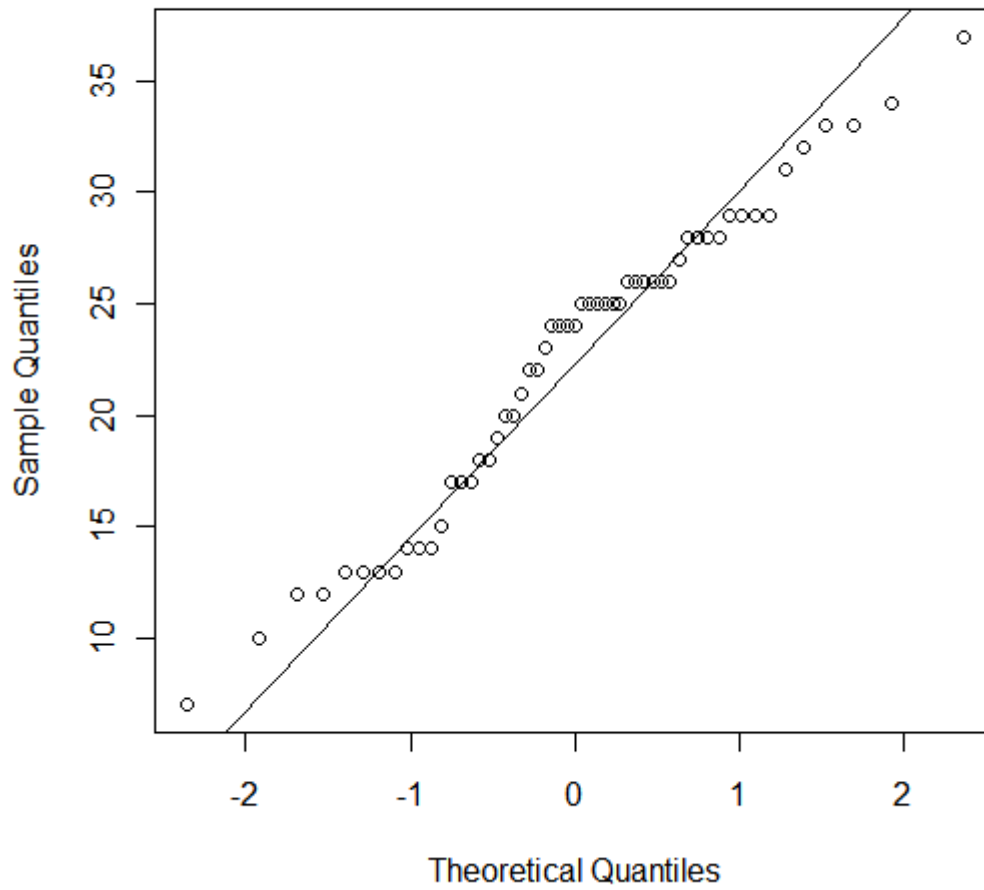


لمقارنة مجموعة بيانات بالتوزيع الطبيعي نستخدم الدالة `qqnorm(...)` والتي ترسم ربيعات البيانات ضد ربيعات التوزيع الطبيعي والدالة `qqline(...)` سوف يقوم برسم خط مستقيم على الرسم السابق يمر بالربيعين الأول والثالث وبالتالي إذا كانت البيانات طبيعية فإن نقاط الرسم الأول ستكون قريبة من الخط المستقيم

```
> qqnorm(exam2)
> qqline(exam2)
```

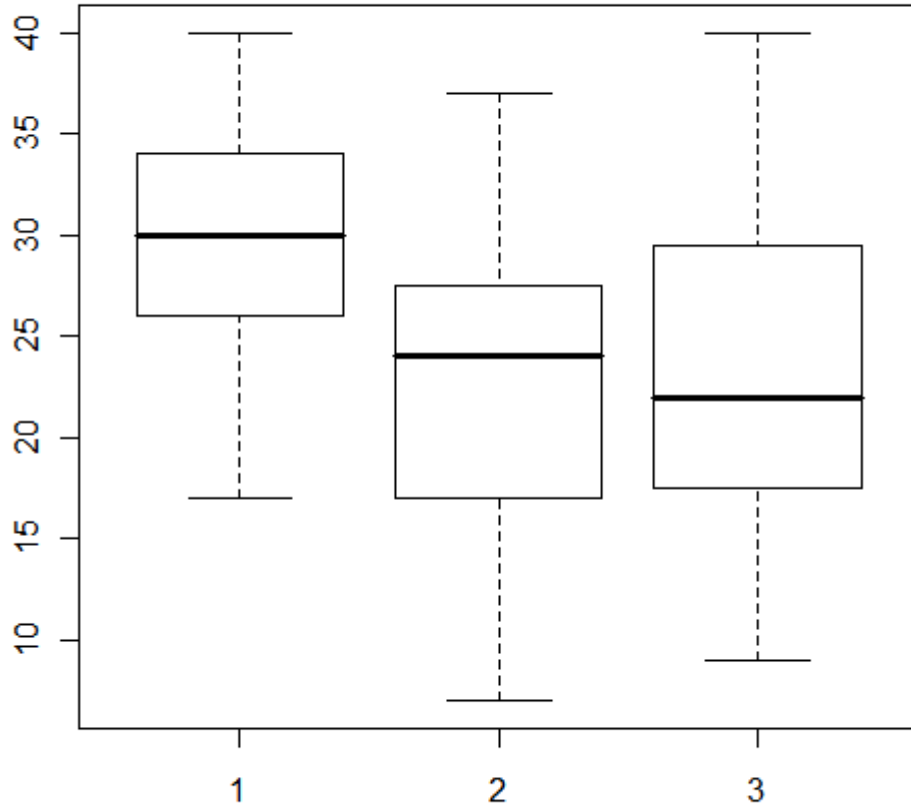


## Normal Q-Q Plot



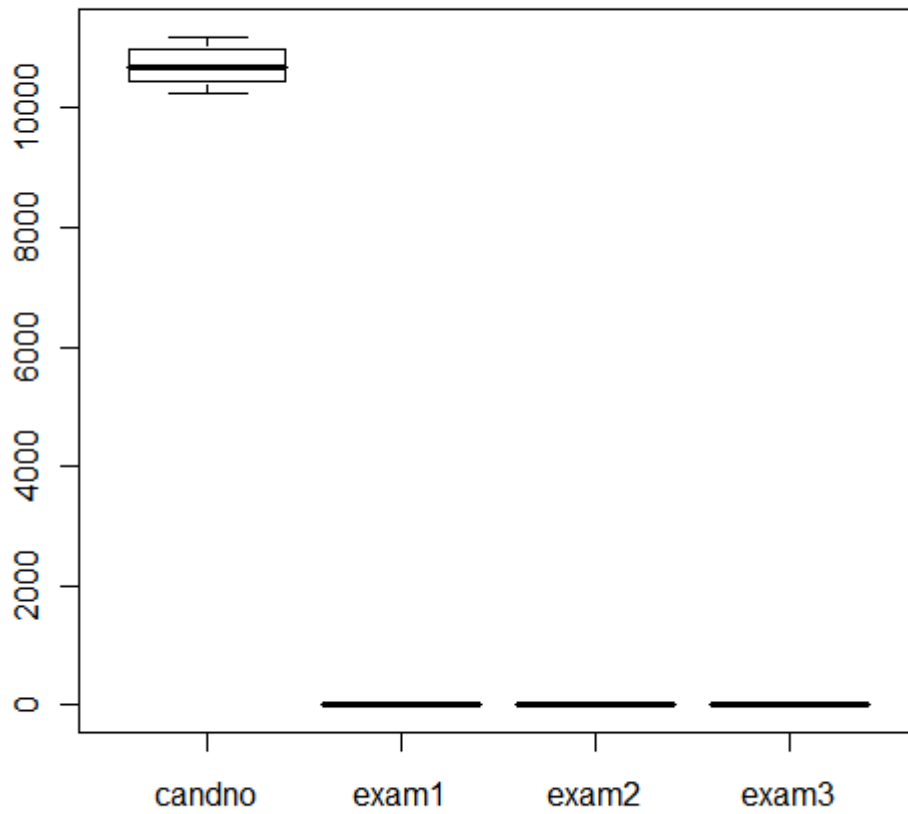
## رسومات الصندوق Boxplots

```
> boxplot(exam1, exam2, exam3)
```



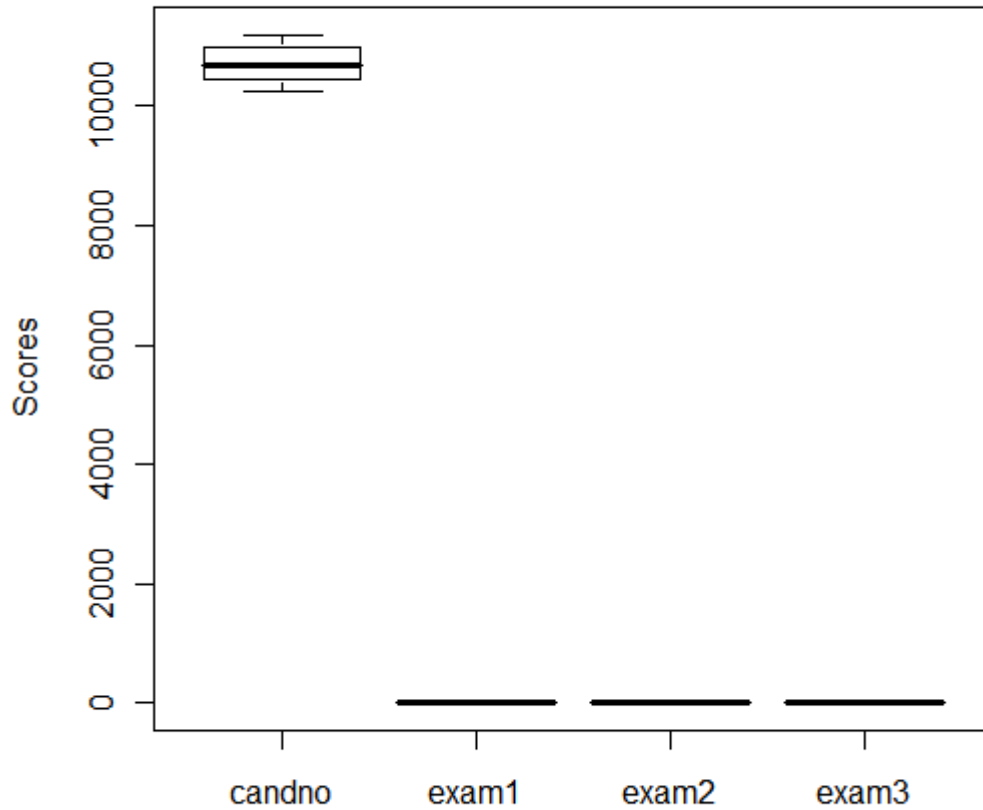
لاحظ التالي:

```
> boxplot(mk2nd)
```



```
> boxplot(mk2nd, main="Boxplot of exam scores",  
+ylab="Scores")
```

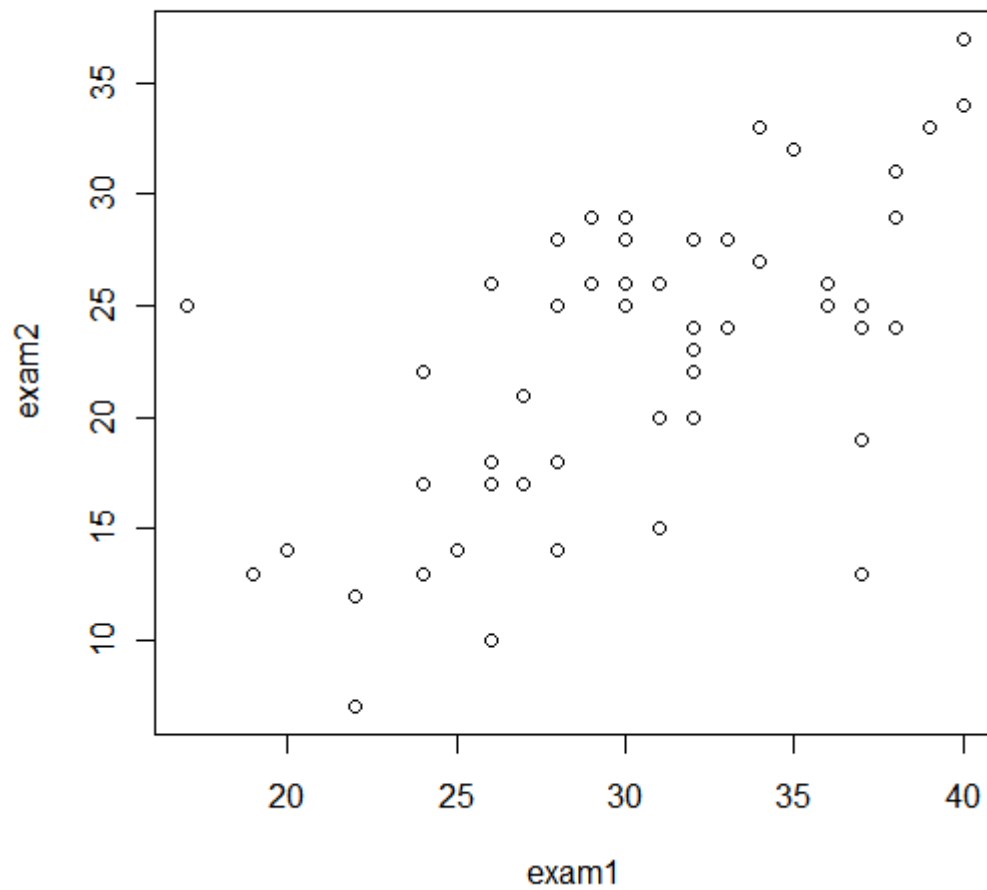
## Boxplot of exam scores



### Plot types, graphics أنواع الرسومات ومعالم الرسومات والتفاعلية : parameters and interactivity

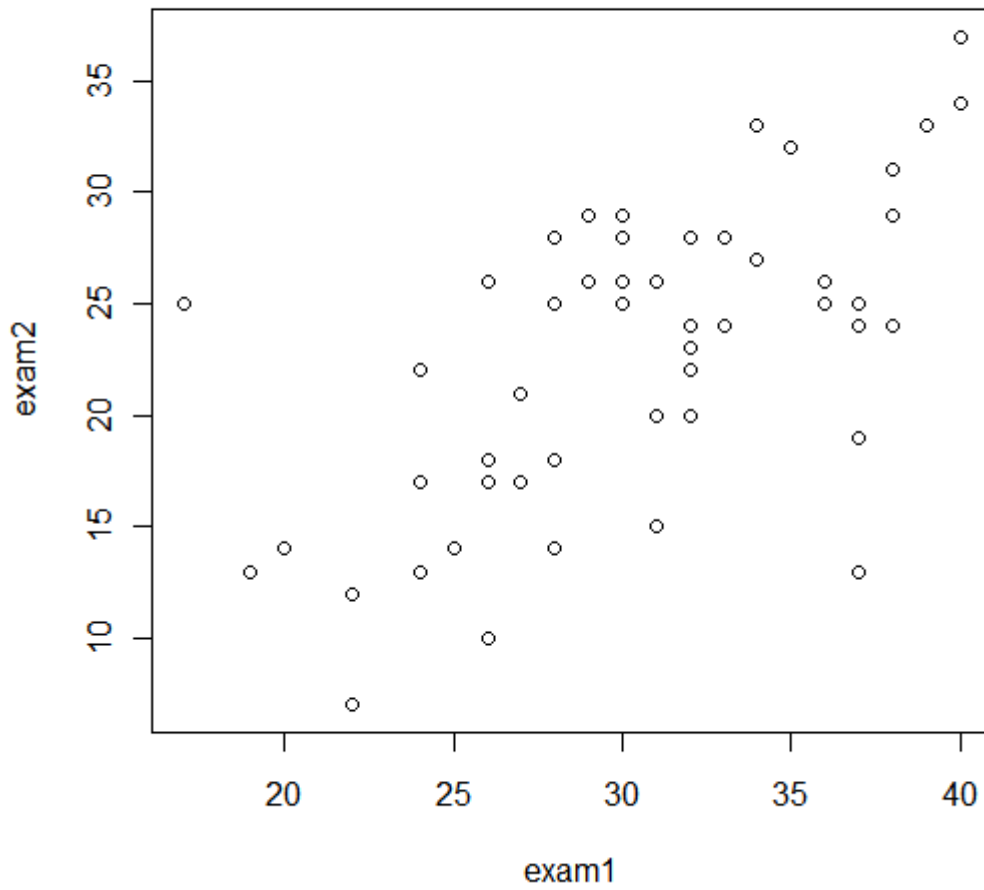
نوع الرسم الإفتراضي في R بسيط جدا ومناسب ولكن قد نحتاج أحيانا أن نغيره ولهذا  
تستخدم حجة `type` في دالة الرسم لاحظ التالي:

```
> plot(exam1, exam2)
```



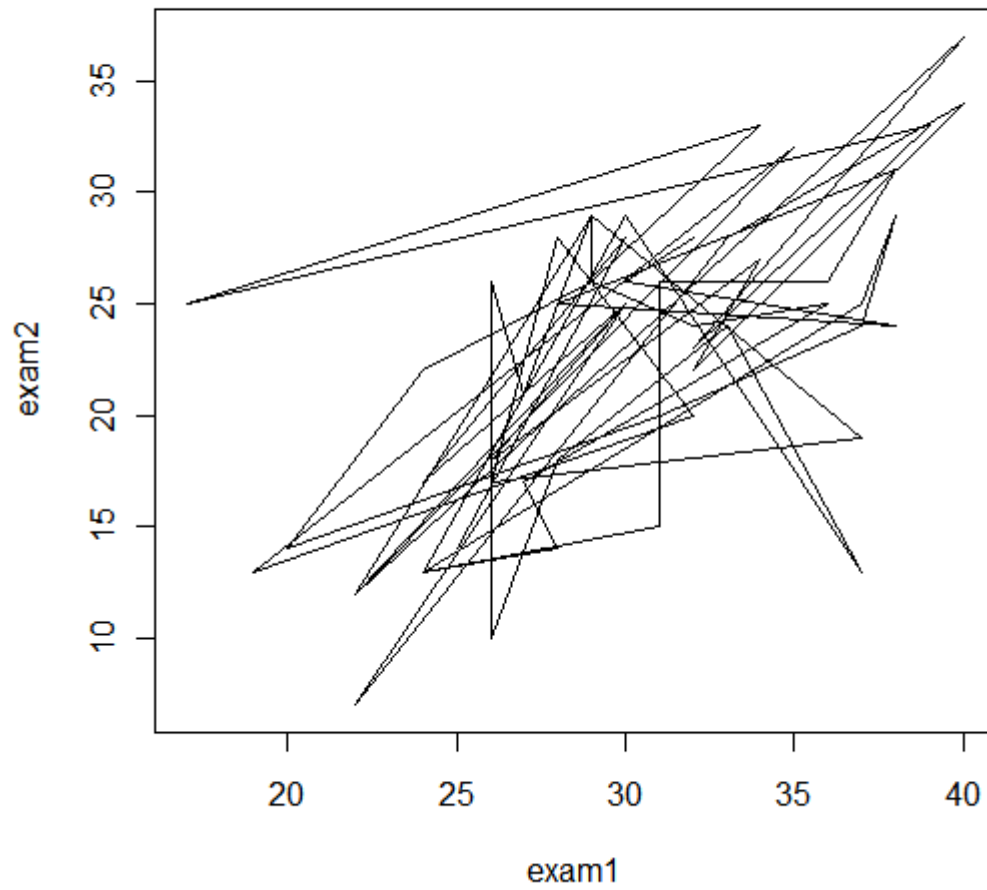
```
> plot(exam1, exam2, type="p") # points (the default)
```

p تعني نقاط وهي القيم الإفتراضية



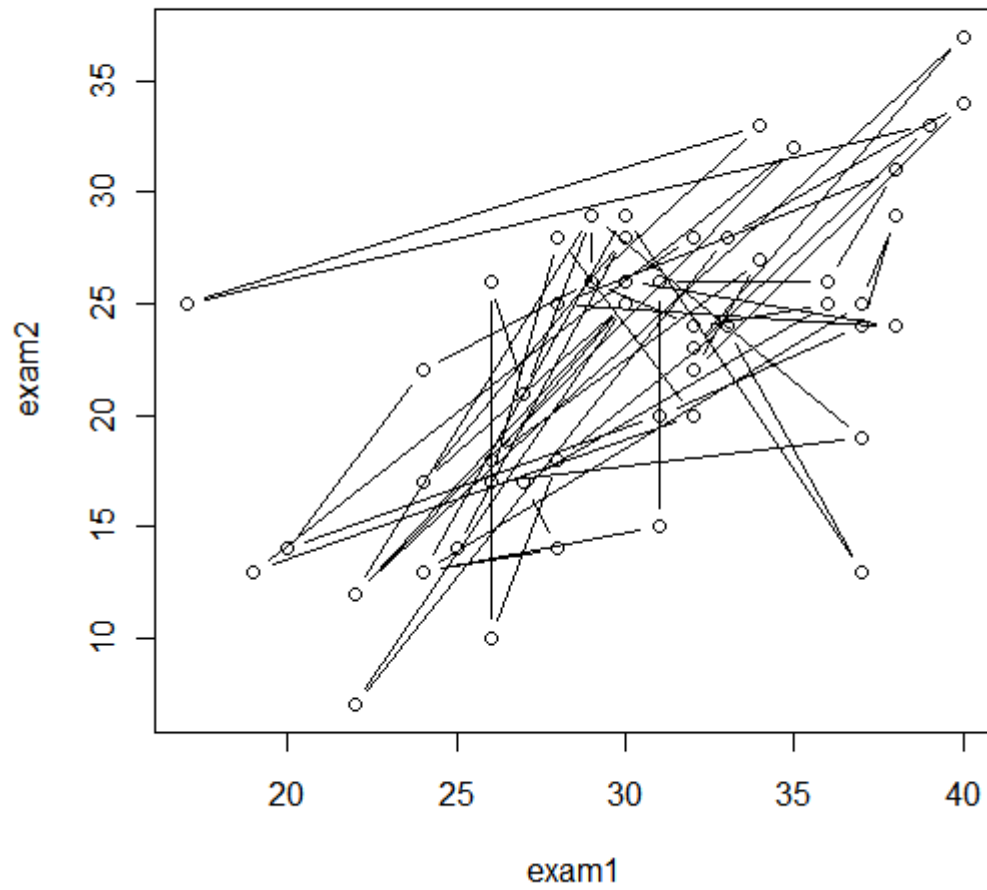
```
> plot(exam1, exam2, type="l") # lines
```

ا من أي خط line



```
> plot(exam1, exam2, type="b") # both lines and points
```

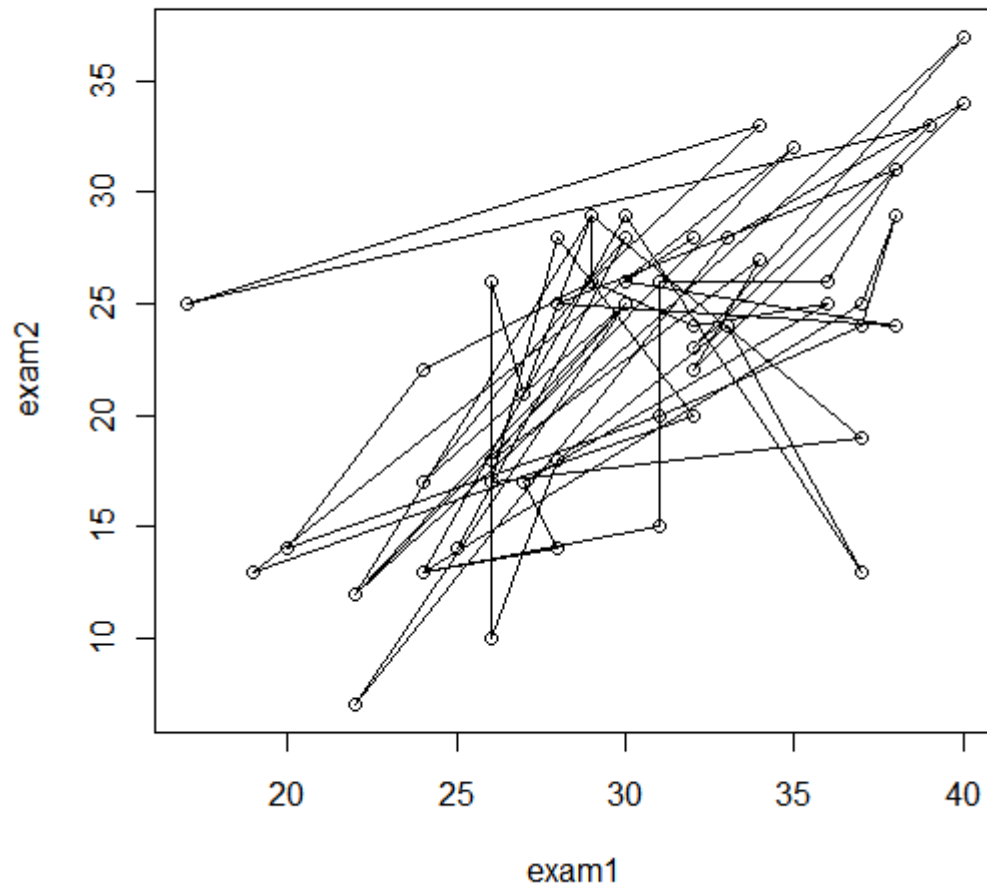
b من both أي كلا من نقاط و خطوط



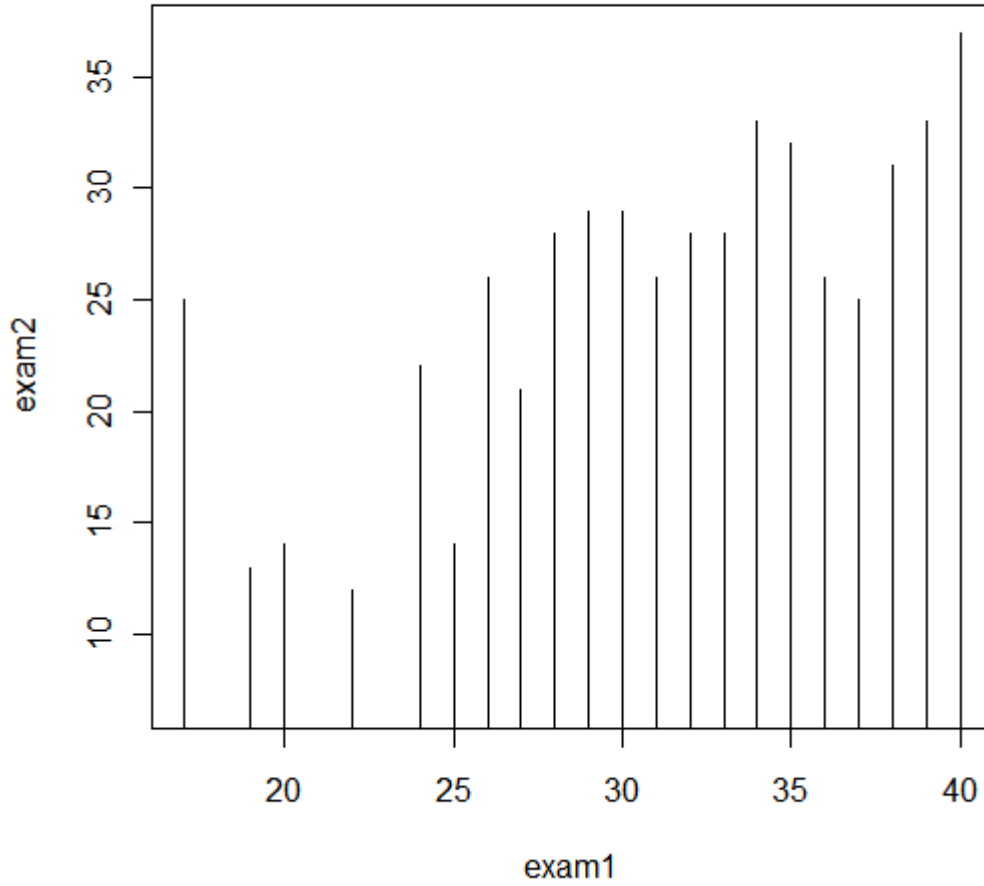
```
> plot(exam1, exam2, type="o") # overlaid lines on points
```

o من overlaid وهو وضع النقاط على الخطوط



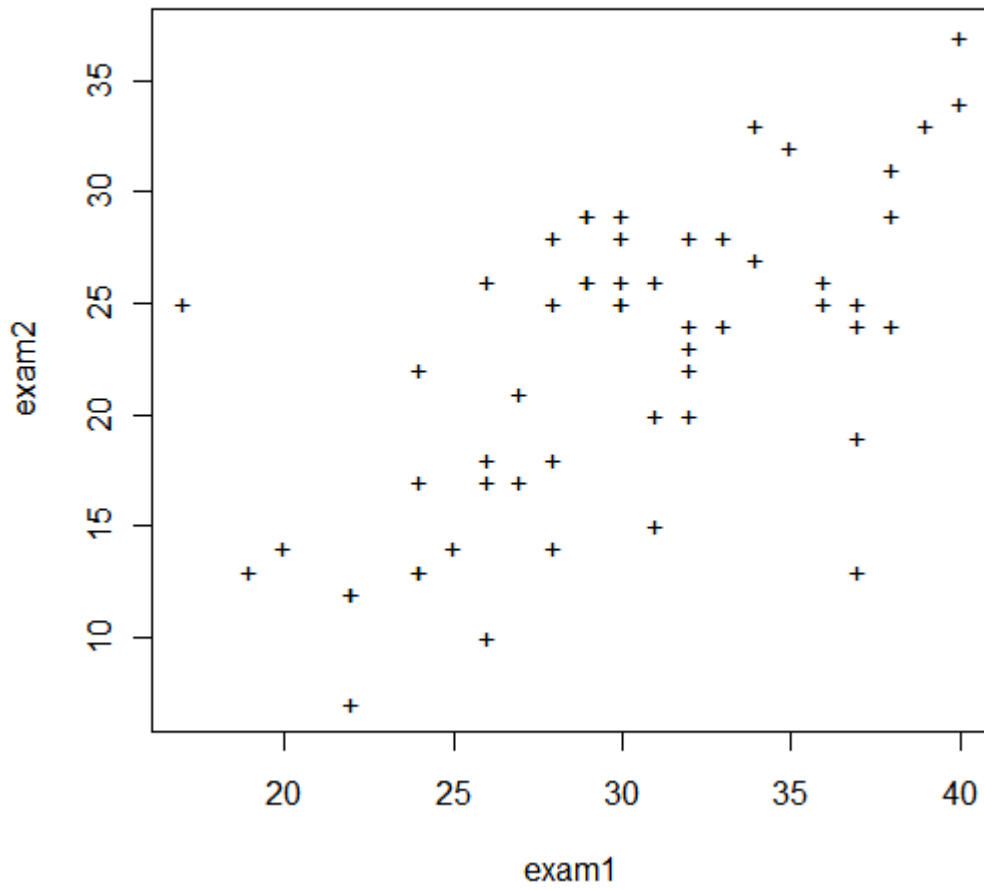


```
> plot(exam1, exam2, type="h") # high density  
(vertical lines)
```



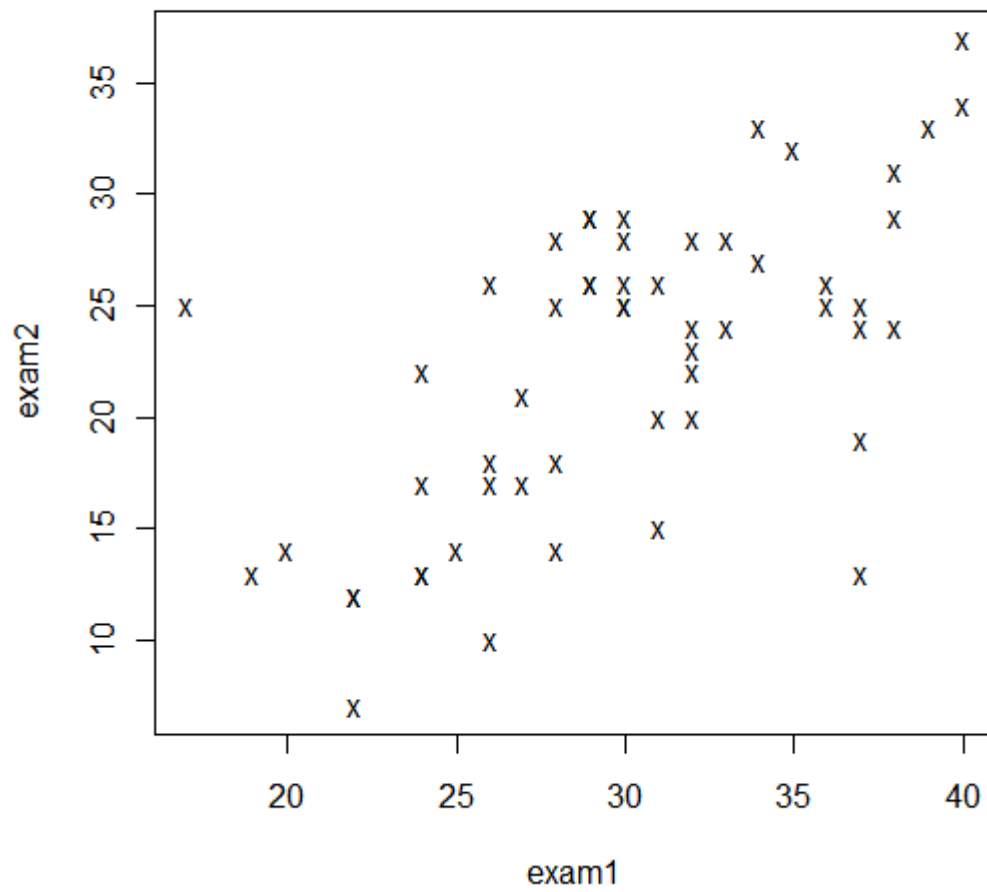
نستطيع تغيير حرف الرسم الافتراضي *plotting character* والذي هو 0 باستخدام الحجة *pch* إلى أي رمز منها

```
> plot(exam1, exam2, pch="+")
```



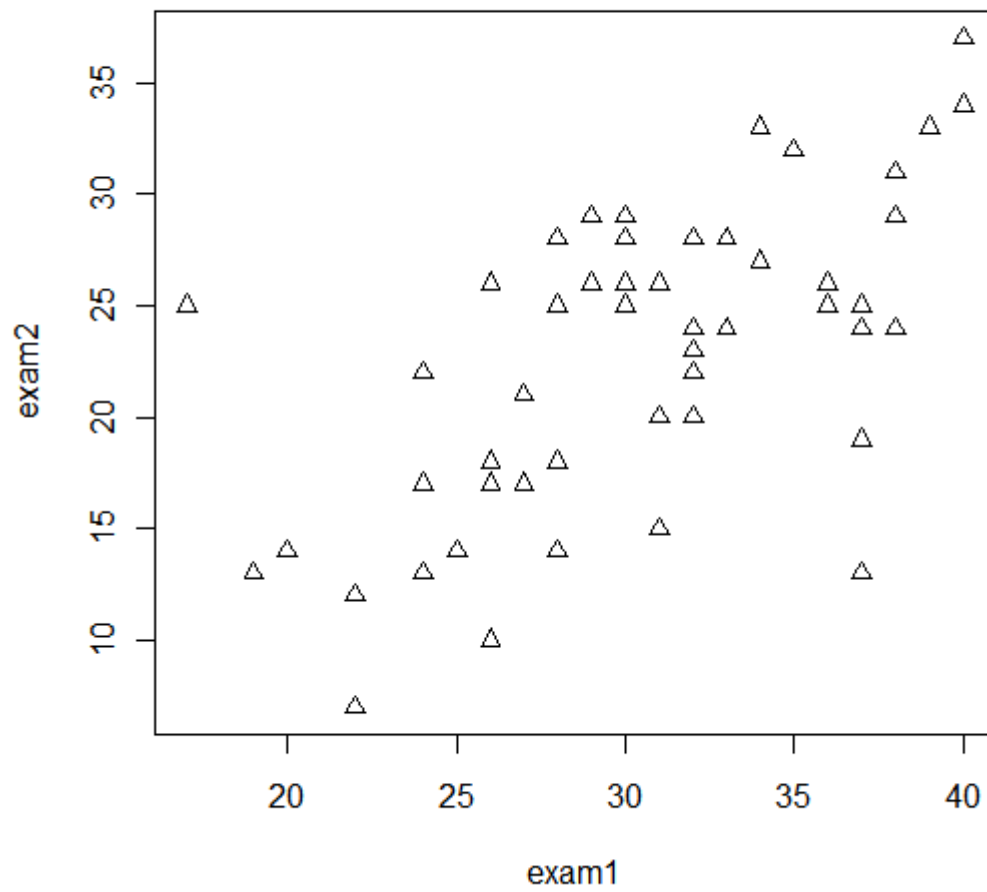
```
> plot(exam1, exam2, pch="x")
```

9



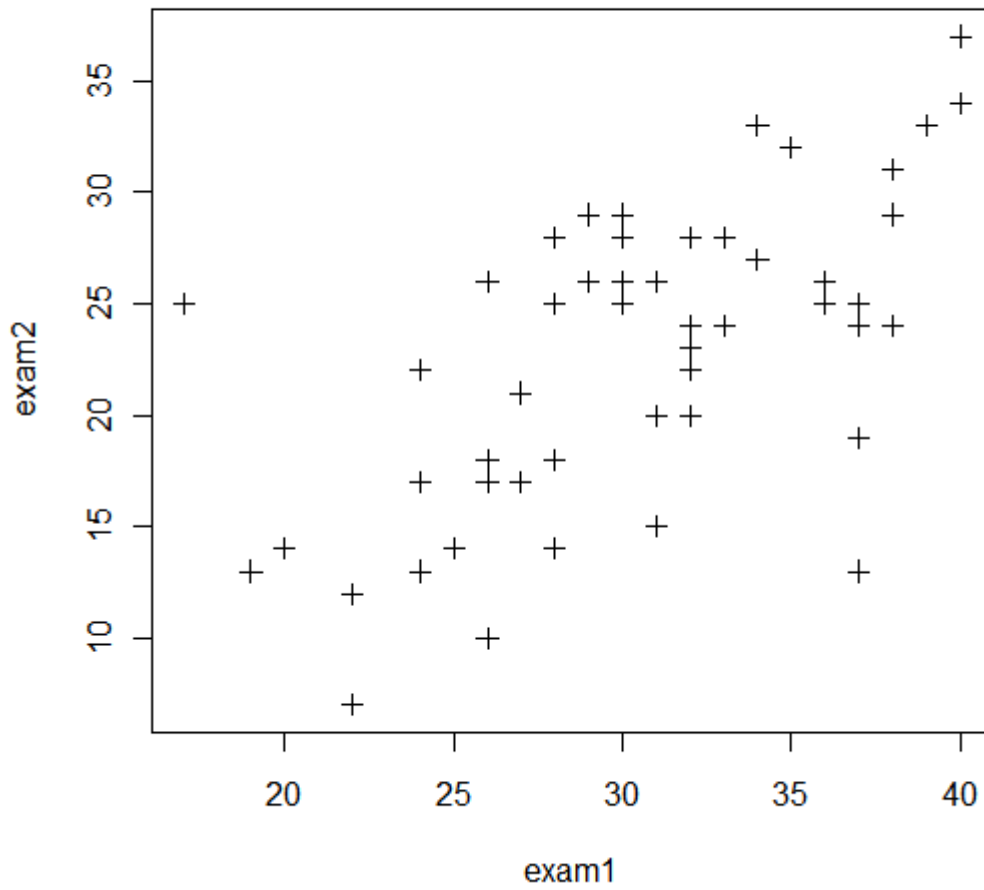
```
> plot(exam1, exam2, pch=2)
```

9



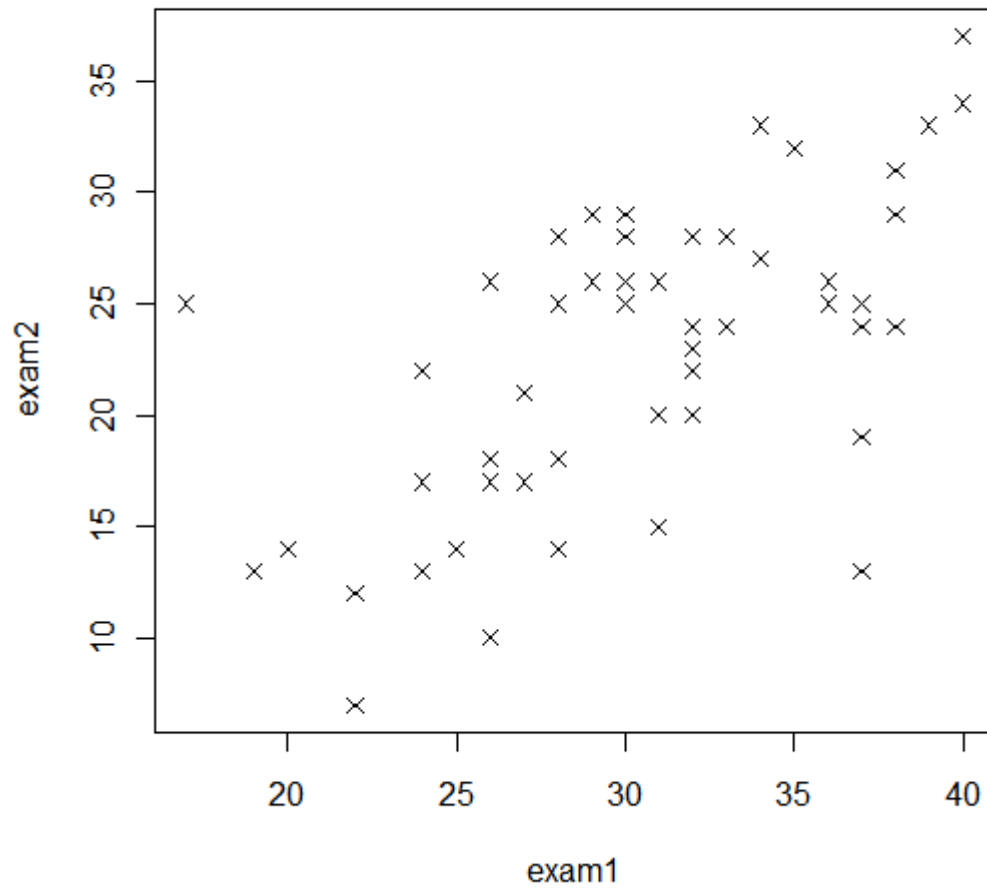
```
> plot(exam1, exam2, pch=3)
```

و



```
> plot(exam1, exam2, pch=4)
```

و



إذا اردنا تغيير معلم رسم بصورة دائمة نستخدم الدالة `par(...)` وكذلك يستخدم لسرد معالم الرسم

```
> par()
$xlog
[1] FALSE

$ylog
[1] FALSE

$adj
[1] 0.5

$ann
[1] TRUE
```

```
$ask
[1] FALSE

$bg
[1] "transparent"

$btty
[1] "o"

$cex
[1] 1

$cex.axis
[1] 1

$cex.lab
[1] 1

$cex.main
[1] 1.2

$cex.sub
[1] 1

$cin
[1] 0.15 0.20

$col
[1] "black"

$col.axis
[1] "black"

$col.lab
[1] "black"

$col.main
[1] "black"

$col.sub
```



```
[1] "black"

$cra
[1] 14.4 19.2

$crt
[1] 0

$csi
[1] 0.2

$cxy
[1] 0.03437083 0.05328007

$din
[1] 5.604166 5.593749

$err
[1] 0

$family
[1] ""

$fg
[1] "black"

$fig
[1] 0 1 0 1

$fin
[1] 5.604166 5.593749

$font
[1] 1

$font.axis
[1] 1

$font.lab
[1] 1
```

\$font.main

[1] 2

\$font.sub

[1] 1

\$lab

[1] 5 5 7

\$las

[1] 0

\$lend

[1] "round"

\$lheight

[1] 1

\$ljoin

[1] "round"

\$lmitre

[1] 10

\$lty

[1] "solid"

\$lwd

[1] 1

\$mai

[1] 1.02 0.82 0.82 0.42

\$mar

[1] 5.1 4.1 4.1 2.1

\$mex

[1] 1

\$mfcoll  
[1] 1 1

\$mfg  
[1] 1 1 1 1

\$mfrow  
[1] 1 1

\$mgp  
[1] 3 1 0

\$mkh  
[1] 0.001

\$new  
[1] FALSE

\$oma  
[1] 0 0 0 0

\$omd  
[1] 0 1 0 1

\$omi  
[1] 0 0 0 0

\$pch  
[1] 1

\$pin  
[1] 4.364166 3.753749

\$plt  
[1] 0.1463197 0.9250557 0.1823464 0.8534078

\$ps  
[1] 12

\$pty

```
[1] "m"
```

```
$smo
```

```
[1] 1
```

```
$srt
```

```
[1] 0
```

```
$tck
```

```
[1] NA
```

```
$tcl
```

```
[1] -0.5
```

```
$usr
```

```
[1] 0 1 0 1
```

```
$xaxp
```

```
[1] 0 1 5
```

```
$xaxs
```

```
[1] "r"
```

```
$xaxt
```

```
[1] "s"
```

```
$xpd
```

```
[1] FALSE
```

```
$yaxp
```

```
[1] 0 1 5
```

```
$yaxs
```

```
[1] "r"
```

```
$yaxt
```

```
[1] "s"
```

```
>
```

لمعرفة القيم الافتراضية للون والحرف

```
> par(c("pch", "col"))
```

```
$pch  
[1] 1
```

```
$col  
[1] "black"
```

```
>
```

لتغيير الحرف الافتراضي إلى \*

```
> par(pch="*")
```

### الرسومات التفاعلية:

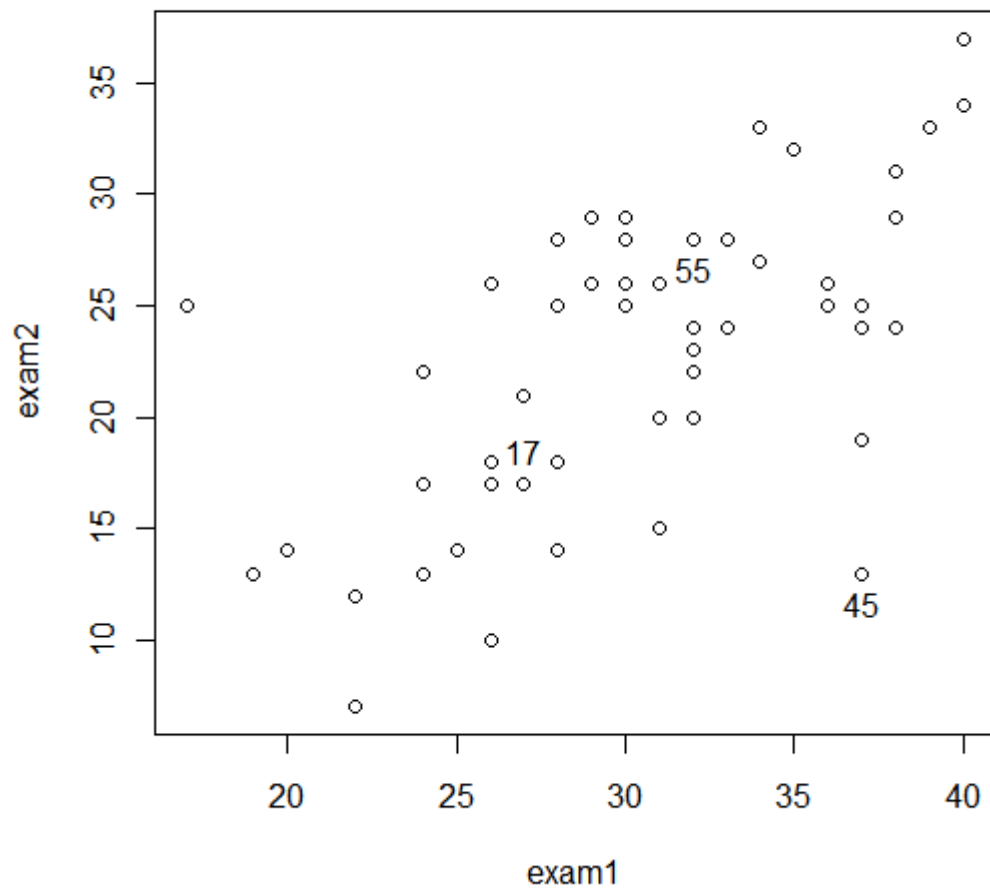
تمتلك R عدد من إمكانيات الرسم التفاعلي وأحد هذه هو الدالة `identify(...)` والتي تمكن من تأشير نقاط مهمة في الرسم. لاحظ بعد إدخال الدالة `identify(...)` سوف ينتظر البرنامج حتى يستطيع المستخدم إختيار النقاط مستخدما الفارة وتوقف العملية بإستخدام الزر الأيمن للفارة

```
> plot(exam1, exam2)
```

```
> identify(exam1, exam2)
```

```
[1] 17 45 55
```

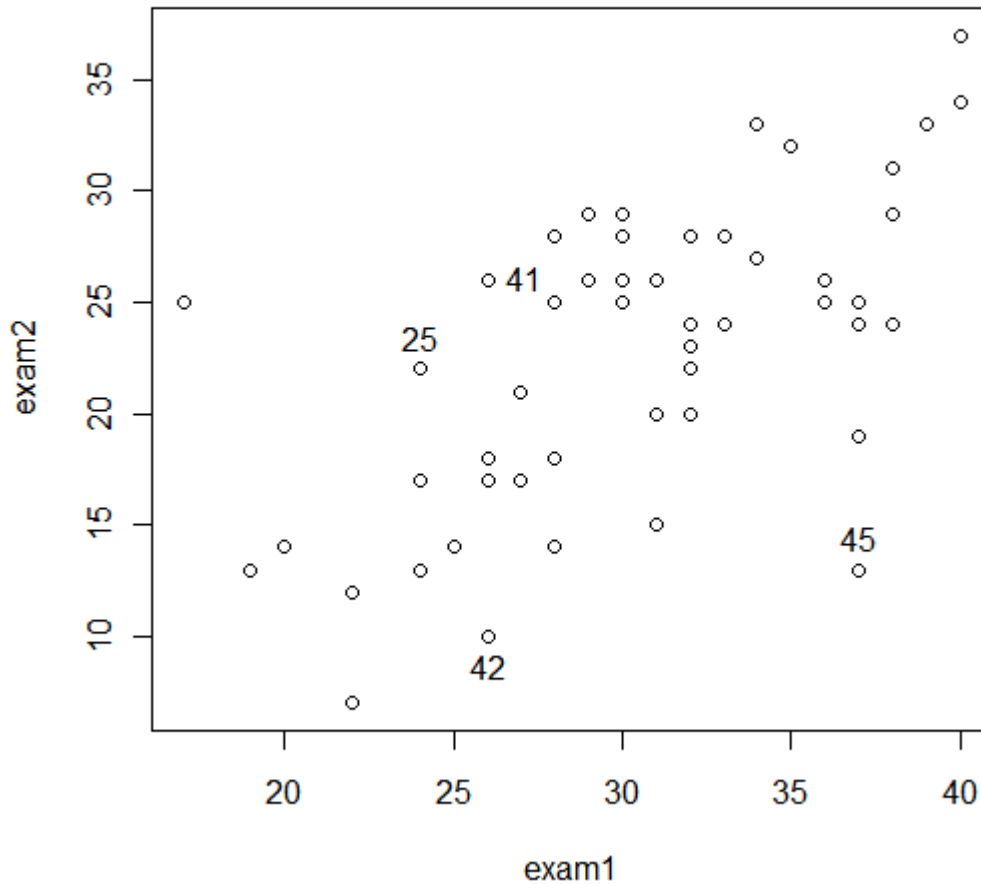
```
>
```



```

> plot(exam1,exam2)
> identify(exam1,exam2,row.names(mk2nd))
[1] 25 41 42 45
>

```



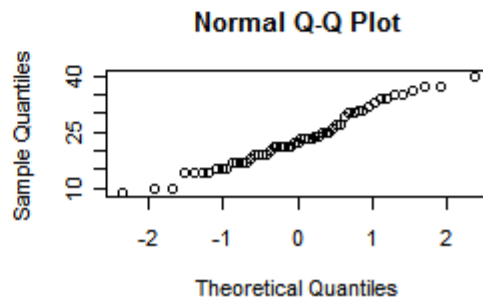
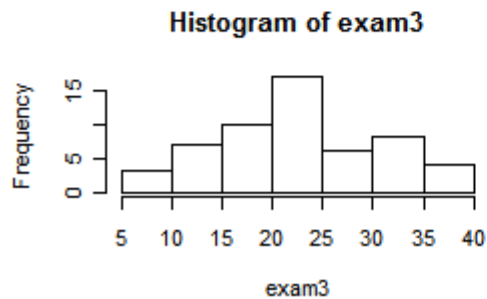
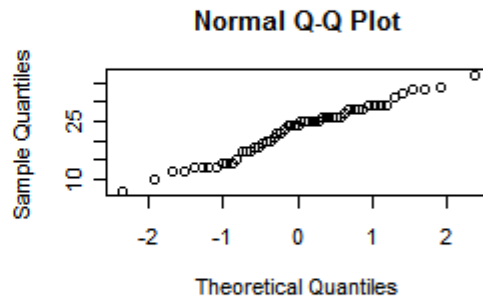
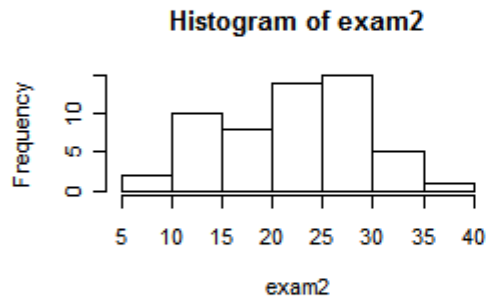
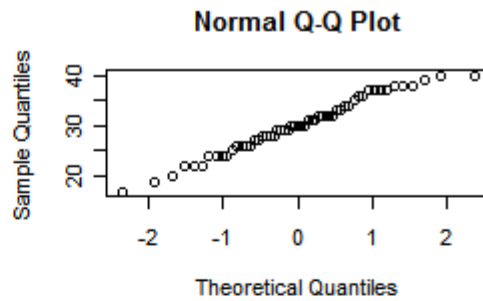
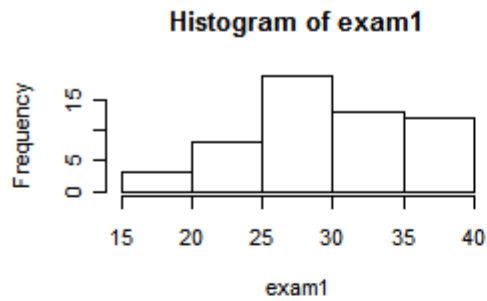
في الإمكان وضع أكثر من رسم واحد في الصفحة وذلك باستخدام المعلم `mfrow` والذي تكون قيمته لمتجة عددي صحيح ذا بعدين يعطي عدد الأسطر والأعمدة لاحظ نتيجة التالي:

```

> par(mfrow=c(3,2))
> hist(exam1)
> qqnorm(exam1)
> hist(exam2)
> qqnorm(exam2)
> hist(exam3)
> qqnorm(exam3)

```

```
> par(mfrow=c(1,1))
```



لاحظ الأمر الأخير الذي يعيد الرسم للقيمة الافتراضية.  
جرب التالي:

```
> par(mfcol=c(3,2))
```

والذي يستخدم `mfcol` بدلا من `mfrow`.  
تمرين: أدرس التالي

```
> citynames <- c("Athens", "Jo'burg", "London",  
"NYC", "Shanghai")  
> longitude <- c(23.72, 28.07, -0.08, -73.47,  
121.47)  
> latitude <- c(37.97, -26.20, 53.42, 40.78, 31.17)  
> cities <- data.frame(latitude,longitude)  
> row.names(cities) <- citynames  
> cities
```

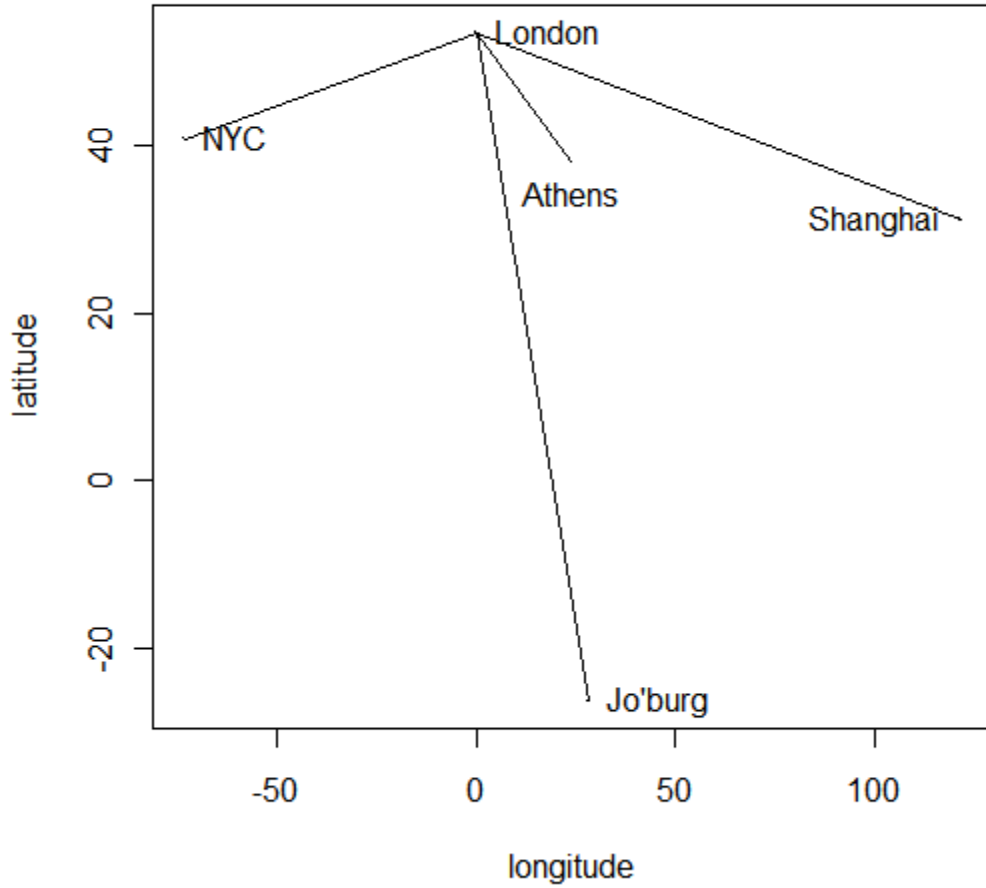


```

      latitude longitude
Athens      37.97      23.72
Jo'burg    -26.20      28.07
London      53.42      -0.08
NYC         40.78     -73.47
Shanghai    31.17     121.47
> rm(latitude,longitude,citynames)
> attach(cities)
> plot(longitude,latitude,type="n")
> points(longitude,latitude,pch=".")
> text(longitude[1], latitude[1],
row.names(cities)[1],pos=1)
> text(longitude[2], latitude[2],
row.names(cities)[2],pos=4)
> text(longitude[3], latitude[3],
row.names(cities)[3],pos=4)
> text(longitude[4], latitude[4],
row.names(cities)[4],pos=4)
> text(longitude[5], latitude[5],
row.names(cities)[5],pos=2)
> lines(c(longitude[1],longitude[3]),
c(latitude[1],latitude[3]))
> lines(c(longitude[2],longitude[3]),
c(latitude[2],latitude[3]))
> lines(c(longitude[4],longitude[3]),
c(latitude[4],latitude[3]))
> lines(c(longitude[5],longitude[3]),
c(latitude[5],latitude[3]))
>

```

الرسم الناتج:



لاحظ التالي:

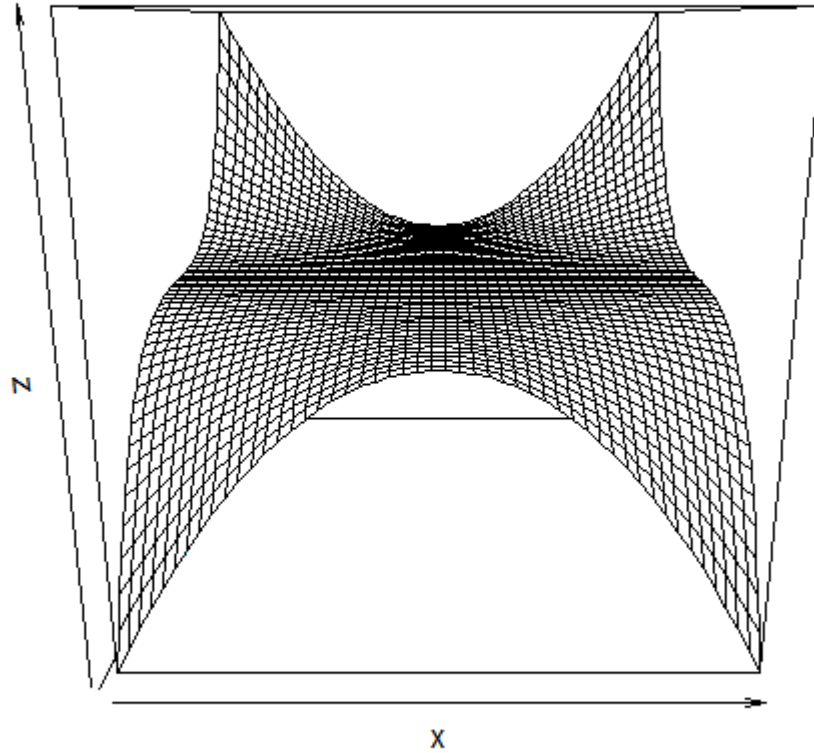
- 1- أمر الرسم `plot(longitude, latitude, type="n")` سوف يقوم برسم الإحداثيات ولكن `type = "n"` سوف يمنع رسم أي شيء آخر.
- 2- الحجة `pos` تأخذ القيمة 1 لأعلى و 2 لليساار و 3 لأسفل و 4 لليمين. لتنسيق وضع النصوص يمكن استخدام الحجج `adj` و `offset` وتترك كتمرين.

### الرسم ثلاثي الأبعاد 3-D:

سوف نقوم برسم السطح  $f(x, y) = x^2 y^3$  في المجال  $x \in [-1, 1]$  و  $y \in [-1, 1]$  ولكي نقوم بهذا نولد متتابعة منسقة من 50 نقطة للفترة  $[-1, 1]$  ثم نستخدم الدالة `outer` (للضرب الخارجي) لتوليد مصفوفة 50X50 لنقاط على السطح

```
> x <- seq(-1,1,length=50)
> y <- seq(-1,1,length=50)
```

```
> z <- outer(x^2,y^3)
> contour(x,y,z)
> image(x,y,z)
> persp(x,y,z)
```



يمكن تحسين شكل الرسم وذلك بتغيير بعض القيم الافتراضية.

## أساسيات R essentials :

في أي جلسة من جلسات R يقوم البرنامج بحفظ الأوامر المدخلة من لوحة المفاتيح في منطقة من الذاكرة `buffer` ويمكن إستعادة هذه الأوامر بأحد المفاتيح `↑` أو `↓` وقد يكون من المفيد تخزين هذه الأوامر للإستخدام في جلسة أخرى لهذا نحفظ هذا التاريخ بالأمر:

```
> savehistory("filename.Rhistory")
```

وتستعاد بالأمر:

```
> loadhistory("filename.Rhistory")
```

بعد إستعادة الأوامر يمكن إستخدامها بأحد المفاتيح `↑` أو `↓` أما إذا شئنا إجراء هذه الأوامر نستخدم الأمر:

```
> source("filename ", echo=TRUE)
```

## مسار البحث The search path

عند إمرار معرف لدالة تقوم R بالبحث عن الشيء التابع للمعرف. أول مكان تبحث فيه R هو فضاء العمل ( والذي يعرف أيضا بالإسم البيئة الشاملة *global environment* ويرمز له `GlobalEnv` ). إذا لم يوجد هذا الشيء في فضاء العمل سيقوم R بالبحث في أي قائمة مضافة (بما فيها أطر البيانات) والحزم المحملة بواسطة المستخدم والحزم المحملة ذاتيا ومن هنا تتضح عمل `attach(...)` الحقيقي حيث أنها تضع قائمة في مسار البحث و `detach(...)` تزيلها من المسار. الدالة `search()` تعطي مسار البحث وإعطاء إسم أي بيئة أو موقعها لدالة الأشياء سوف يقوم بسرد الأشياء في تلك البيئة

```
> search()
```

```
> objects(package:MASS)
```

```
> objects(4)
```

## حزم R packages:

وهي عبارة عن ملفات تحوي برامج إضافية متخصصة تضاف إلى R لإعطائها مقدرات لحل مجال أكبر من المشاكل لا تتوفر في أساس R. وفي مقدورنا سرد قائمة بالحزم المتوفرة وكذلك تحميل هذه الحزم وتجديدها بإستخدام الأمر `library(...)`. الأمر التالي يسرد كل الحزم المتوفرة

```
> library()
```

```
Packages in library 'C:/PROGRA~1/R/R-28~1.1/library':
```

base	The R Base Package
boot	Bootstrap R (S-Plus) Functions (Canty)
class	Functions for Classification
cluster	Cluster Analysis Extended Rousseeuw et al.
codetools	Code Analysis Tools for R
datasets	The R Datasets Package
foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...
graphics	The R Graphics Package
grDevices	The R Graphics Devices and Support for Colours and Fonts
grid	The Grid Graphics Package
KernSmooth	Functions for kernel smoothing for Wand & Jones (1995)
lattice	Lattice Graphics
MASS	Main Package of Venables and Ripley's MASS
methods	Formal Methods and Classes
mgcv	GAMs with GCV/AIC/REML smoothness
estimation	and GAMMs by PQL
nlme	Linear and Nonlinear Mixed Effects Models
nnet	Feed-forward Neural Networks and
Multinomial	Log-Linear Models
R.methodsS3	Utility function for defining S3 methods
R.oo	R object-oriented programming with or without references
rpart	Recursive Partitioning
Sim101	Sim 101 Package for WSC 2008
spatial	Functions for Kriging and Point Pattern Analysis
splines	Regression Spline Functions and Classes
stats	The R Stats Package
stats4	Statistical Functions using S4 Classes
survival	Survival analysis, including penalised likelihood.
tcltk	Tcl/Tk Interface
tools	Tools for Package Development
utils	The R Utils Package

الأمر التالي يحمل الحزمة MASS

```
> library(MASS)
```

ونستطيع مشاهدة محتواه

```
> library(help=MASS)
```

```
Information on package 'MASS'
```

```
Description:
```

```

Bundle:          VR
Priority:        recommended
Contains:       MASS class nnet spatial
Version:        7.2-45
Date:           2008-12-07
Depends:        R (>= 2.4.0), grDevices, graphics, stats,
               utils
Suggests:       lattice, nlme, survival
Author:         S original by Venables & Ripley. R port by
               Brian Ripley <ripley@stats.ox.ac.uk>,
               following earlier work by Kurt Hornik and
               Albrecht Gebhardt.
Maintainer:     Brian Ripley <ripley@stats.ox.ac.uk>
BundleDescription: Functions and datasets to support Venables
               and Ripley, 'Modern Applied Statistics with
               S' (4th edition).
License:        GPL-2 | GPL-3
URL:            http://www.stats.ox.ac.uk/pub/MASS4/
Packaged:       Tue Dec 9 11:11:17 2008; riple
Package:        MASS
Description:    The main library and the datasets
Title:          Main Package of Venables and Ripley's MASS
LazyLoad:       yes
LazyData:       yes
Built:          R 2.8.1; i386-pc-mingw32; 2008-12-22
               09:22:56; windows

```

Index:

Functions:  
=====

Null	Null Spaces of Matrices
addterm	Try All One-Term Additions to a Model
anova.negbin	Likelihood Ratio Tests for Negative
Binomial GLMs	
area	Adaptive Numerical Integration
bandwidth.nrd	Bandwidth for density() via Normal
Reference	
	Distribution
bcv	Biased Cross-Validation for Bandwidth
Selection	
boxcox	Box-Cox Transformations for Linear Models
con2tr	Convert Lists to Data Frames for use by
Trellis	
confint-MASS	Confidence Intervals for Model Parameters
contr.sdif	Successive Differences contrast coding
corresp	Simple Correspondence Analysis
cov.rob	Resistant Estimation of Multivariate
Location and	
	Scatter

cov.trob Distribution	Covariance Estimation for Multivariate t
denumerate 'loglm'	Transform an Allowable Formula for  into one for 'terms'
dose.p	Predict Doses for Binomial Assay model
dropterm	Try All One-Term Deletions from a Model
eqscplot	Plots with Geometrically Equal Scales
fitdistr Distributions	Maximum-likelihood Fitting of Univariate
fractions	Rational Approximation
gamma.dispersion Parameter	Calculate the MLE of the Gamma Dispersion  in a GLM Fit
gamma.shape Distribution	Estimate the Shape Parameter of the Gamma  in a GLM Fit
ginv	Generalized Inverse of a Matrix
glm.convert fit	Change a Negative Binomial fit to a GLM
glm.nb Linear Model	Fit a Negative Binomial Generalized
glmmPQL PQL	Fit Generalized Linear Mixed Models via
hist.scott Selection	Plot a Histogram with Automatic Bin Width
huber Scale	Huber M-estimator of Location with MAD
hubers Location	Huber Proposal 2 Robust Estimator of  and/or Scale
isoMDS Scaling	Kruskal's Non-metric Multidimensional
kde2d	Two-Dimensional Kernel Density Estimation
lda	Linear Discriminant Analysis
ldahist Groups	Histograms or Density Plots of Multiple
lm.gls Squares	Fit Linear Models by Generalized Least
lm.ridge	Ridge Regression
loglm	Fit Log-Linear Models by Iterative Proportional Scaling
logtrans	Estimate log Transformation Parameter
lqs	Resistant Regression
mca	Multiple Correspondence Analysis
mvrnorm Distribution	Simulate from a Multivariate Normal
negative.binomial GLMs	Family function for Negative Binomial
newcomb Time of Light	Newcomb's Measurements of the Passage

pairs.lda	Produce Pairwise Scatterplots from an
'lda' Fit	
parcoord	Parallel Coordinates Plot
plot.lda	Plot Method for Class 'lda'
plot.mca	Plot Method for Objects of Class 'mca'
polr	Proportional Odds Logistic Regression
predict.lda	Classify Multivariate Observations by
Linear	
	Discrimination
predict.lqs	Predict from an lqs Fit
predict.mca	Predict Method for Class 'mca'
predict.qda	Classify from Quadratic Discriminant
Analysis	
qda	Quadratic Discriminant Analysis
rational	Rational Approximation
renumerate	Convert a Formula Transformed by
'denumerate'	
rlm	Robust Fitting of Linear Models
rms.curv	Relative Curvature Measures for Non-
Linear Regression	
rnegbin	Simulate Negative Binomial Variates
sammon	Sammon's Non-Linear Mapping
stdres	Extract Standardized Residuals from a
Linear Model	
stepAIC	Choose a model by AIC in a Stepwise
Algorithm	
studres	Extract Studentized Residuals from a
Linear Model	
summary.loglm	Summary Method Function for Objects of
Class 'loglm'	
summary.negbin	Summary Method Function for Objects of
Class 'negbin'	
summary.rlm	Summary Method for Robust Linear Models
synth.tr	Synthetic Classification Problem
theta.md	Estimate theta of the Negative Binomial
by Deviance	
theta.ml	Estimate theta of the Negative Binomial
by Maximum	
	Likelihood
theta.mm	Estimate theta of the Negative Binomial
by Moments	
truehist	Plot a Histogram
ucv	Unbiased Cross-Validation for Bandwidth
Selection	
width.SJ	Bandwidth Selection by Pilot Estimation
of Derivatives	
write.matrix	Write a Matrix or Data Frame

Datasets:  
=====



Aids2	Australian AIDS Survival Data
Animals	Brain and Body Weights for 28 Species
Belgian-phones	Belgium Phone Calls 1950-1973
Boston	Housing Values in Suburbs of Boston
Cars93	Data from 93 Cars on Sale in the USA in
1993	
Cushings	Diagnostic Tests on Patients with
Cushing's Syndrome	
DDT	DDT in Kale
GAGurine	Level of GAG in Urine of Children
Insurance	Numbers of Car Insurance claims
Melanoma	Survival from Malignant Melanoma
OME	Tests of Auditory Perception in Children
with OME	
Pima.tr	Diabetes in Pima Indian Women
Rabbit	Blood Pressure in Rabbits
Rubber	Accelerated Testing of Tyre Rubber
SP500	Returns of the Standard and Poors 500
Sitka	Growth Curves for Sitka Spruce Trees in
1988	
Sitka89	Growth Curves for Sitka Spruce Trees in
1989	
Skye	AFM Compositions of Aphyric Skye Lavas
Traffic	Effect of Swedish Speed Limits on
Accidents	
UScereal	Nutritional and Marketing Information on
US Cereals	
UScrime	The Effect of Punishment Regimes on Crime
Rates	
VA	Veteran's Administration Lung Cancer
Trial	
abbey	Determinations of Nickel Content
accdeaths	Accidental Deaths in the US 1973-1978
anorexia	Anorexia Data on Weight Change
bacteria	Presence of Bacteria after Drug
Treatments	
beav1	Body Temperature Series of Beaver 1
beav2	Body Temperature Series of Beaver 2
biopsy	Biopsy Data on Breast Cancer Patients
birthwt	Risk Factors Associated with Low Infant
Birth Weight	
cabbages	Data from a cabbage field trial
caith	Colours of Eyes and Hair of People in
Caithness	
cats	Anatomical Data from Domestic Cats
cement	Heat Evolved by Setting Cements
chem	Copper in Wholemeal Flour
coop	Co-operative Trial in Analytical
Chemistry	
cpus	Performance of Computer CPUs
crabs	Morphological Measurements on
Leptograpsus Crabs	

drivers	Deaths of Car Drivers in Great Britain
1969-84	
eagles	Foraging Ecology of Bald Eagles
epil	Seizure Counts for Epileptics
farms	Ecological Factors in Farm Management
fgl	Measurements of Forensic Glass Fragments
forbes	Forbes' Data on Boiling Points in the
Alps	
galaxies	Velocities for 82 Galaxies
gehan	Remission Times of Leukaemia Patients
genotype	Rat Genotype Data
geyser	Old Faithful Geyser Data
gilgais	Line Transect of Soil in Gilgai Territory
hills	Record Times in Scottish Hill Races
housing	Frequency Table from a Copenhagen Housing
Conditions	
	Survey
immer	Yields from a Barley Field Trial
leuk	Survival Times and White Blood Counts for
	Leukaemia Patients
mammals	Brain and Body Weights for 62 Species of
Land Mammals	
mcycle	Data from a Simulated Motorcycle Accident
menarche	Age of Menarche data
michelson	Michelson's Speed of Light Data
minn38	Minnesota High School Graduates of 1938
motors	Accelerated Life Testing of Motorettes
muscle	Effect of Calcium Chloride on Muscle
Contraction	
	in Rat Hearts
nlschools	Eighth-Grade Pupils in the Netherlands
npk	Classical N, P, K Factorial Experiment
npri	US Naval Petroleum Reserve No. 1 data
oats	Data from an Oats Field Trial
painters	The Painter's Data of de Piles
petrol	N. L. Prater's Petrol Refinery Data
quine	Absenteeism from School in Rural New
South Wales	
road	Road Accident Deaths in US States
rotifer	Numbers of Rotifers by Fluid Density
ships	Ships Damage Data
shoes	Shoe wear data of Box, Hunter and Hunter
shrimp	Percentage of Shrimp in Shrimp Cocktail
shuttle	Space Shuttle Autolander Problem
snails	Snail Mortality Data
steam	The Saturated Steam Pressure Data
stormer	The Stormer Viscometer Data
survey	Student Survey Data
topo	Spatial Topographic Data
waders	Counts of Waders at 15 Sites in South
Africa	
whiteside	House Insulation: Whiteside's Data

wtloss

Weight Loss Data from an Obese Patient

يلاحظ أن بعض الحزم تحوي مجاميع بيانات ونستطيع معرفة البيانات الموجودة بالأمر `data()` والذي يسرد مجاميع البيانات للحزم المحملة حاليا

**> data()**

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 Indomethicin
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
Seatbelts	Road Casualties in Great Britain 1969-84
Theoph	Pharmacokinetics of theophylline
Titanic	Survival of passengers on the Titanic
ToothGrowth	The Effect of Vitamin C on Tooth Growth in Guinea Pigs
UCBAdmissions	Student Admissions at UC Berkeley
UKDriverDeaths	Road Casualties in Great Britain 1969-84
UKgas	UK Quarterly Gas Consumption
USAccDeaths	Accidental Deaths in the US 1973-1978
USArrests	Violent Crime Rates by US State
USJudgeRatings	Lawyers' Ratings of State Judges in the US Superior Court
USPersonalExpenditure	Personal Expenditure Data
VADeaths	Death Rates in Virginia (1940)

WWWusage	Internet Usage per Minute
WorldPhones	The World's Telephones
ability.cov	Ability and Intelligence Tests
airmiles	Passenger Miles on Commercial US Airlines, 1937-1960
airquality	New York Air Quality Measurements
anscombe	Anscombe's Quartet of "Identical" Simple Linear Regressions
attenu	The Joyner-Boore Attenuation Data
attitude	The Chatterjee-Price Attitude Data
austres	Quarterly Time Series of the Number of Australian Residents
beaver1 (beavers)	Body Temperature Series of Two Beavers
beaver2 (beavers)	Body Temperature Series of Two Beavers
cars	Speed and Stopping Distances of Cars
chickwts	Chicken Weights by Feed Type
co2	Mauna Loa Atmospheric CO2 Concentration
crimtab	Student's 3000 Criminals Data
discoveries	Yearly Numbers of Important Discoveries
esoph	Smoking, Alcohol and (O)esophageal Cancer
euro	Conversion Rates of Euro Currencies
euro.cross (euro)	Conversion Rates of Euro Currencies
eurodist	Distances Between European Cities
faithful	Old Faithful Geyser Data
fdeaths (UKLungDeaths)	Monthly Deaths from Lung Diseases in the UK
freeny	Freeny's Revenue Data
freeny.x (freeny)	Freeny's Revenue Data
freeny.y (freeny)	Freeny's Revenue Data
infert	Infertility after Spontaneous and Induced Abortion
iris	Edgar Anderson's Iris Data
iris3	Edgar Anderson's Iris Data
islands	Areas of the World's Major Landmasses
ldeaths (UKLungDeaths)	Monthly Deaths from Lung Diseases in the UK
lh	Luteinizing Hormone in Blood Samples
longley	Longley's Economic Regression Data
lynx	Annual Canadian Lynx trappings 1821-1934
mdeaths (UKLungDeaths)	Monthly Deaths from Lung Diseases in the UK
morley	Michaelson-Morley Speed of Light Data
mtcars	Motor Trend Car Road Tests
nhtemp	Average Yearly Temperatures in New Haven
nottem	Average Monthly Temperatures at Nottingham, 1920-1939
occupationalStatus	Occupational Status of Fathers and their Sons
precip	Annual Precipitation in US Cities
presidents	Quarterly Approval Ratings of US Presidents
pressure	Vapor Pressure of Mercury as a Function of Temperature

quakes	Locations of Earthquakes off Fiji
randu	Random Numbers from Congruential Generator RANDU
rivers	Lengths of Major North American Rivers
rock	Measurements on Petroleum Rock Samples
sleep	Student's Sleep Data
stack.loss (stackloss)	Brownlee's Stack Loss Plant Data
stack.x (stackloss)	Brownlee's Stack Loss Plant Data
stackloss	Brownlee's Stack Loss Plant Data
state.abb (state)	US State Facts and Figures
state.area (state)	US State Facts and Figures
state.center (state)	US State Facts and Figures
state.division (state)	US State Facts and Figures
state.name (state)	US State Facts and Figures
state.region (state)	US State Facts and Figures
state.x77 (state)	US State Facts and Figures
sunspot.month	Monthly Sunspot Data, 1749-1997
sunspot.year	Yearly Sunspot Data, 1700-1988
sunspots	Monthly Sunspot Numbers, 1749-1983
swiss	Swiss Fertility and Socioeconomic
Indicators	(1888) Data
treering	Yearly Treering Data, -6000-1979
trees	Girth, Height and Volume for Black Cherry Trees
uspop	Populations Recorded by the US Census
volcano	Topographic Information on Auckland's
Maunga	Whau Volcano
warpbreaks	The Number of Breaks in Yarn during Weaving
women	Average Heights and Weights for American Women
Data sets in package 'MASS':	
Aids2	Australian AIDS Survival Data
Animals	Brain and Body Weights for 28 Species
Boston	Housing Values in Suburbs of Boston
Cars93	Data from 93 Cars on Sale in the USA in
1993	
Cushings	Diagnostic Tests on Patients with Cushing's Syndrome
DDT	DDT in Kale
GAGurine	Level of GAG in Urine of Children
Insurance	Numbers of Car Insurance claims
Melanoma	Survival from Malignant Melanoma
OME	Tests of Auditory Perception in Children with OME

Pima.te	Diabetes in Pima Indian Women
Pima.tr	Diabetes in Pima Indian Women
Pima.tr2	Diabetes in Pima Indian Women
Rabbit	Blood Pressure in Rabbits
Rubber	Accelerated Testing of Tyre Rubber
SP500	Returns of the Standard and Poors 500
Sitka	Growth Curves for Sitka Spruce Trees in
1988	
Sitka89	Growth Curves for Sitka Spruce Trees in
1989	
Skye	AFM Compositions of Aphyric Skye Lavas
Traffic	Effect of Swedish Speed Limits on Accidents
UScereal	Nutritional and Marketing Information on US Cereals
UScrime	The Effect of Punishment Regimes on Crime Rates
VA	Veteran's Administration Lung Cancer Trial
abbey	Determinations of Nickel Content
accdeaths	Accidental Deaths in the US 1973-1978
anorexia	Anorexia Data on Weight Change
bacteria	Presence of Bacteria after Drug Treatments
beav1	Body Temperature Series of Beaver 1
beav2	Body Temperature Series of Beaver 2
biopsy	Biopsy Data on Breast Cancer Patients
birthwt	Risk Factors Associated with Low Infant Birth Weight
cabbages	Data from a cabbage field trial
caith	Colours of Eyes and Hair of People in Caithness
cats	Anatomical Data from Domestic Cats
cement	Heat Evolved by Setting Cements
chem	Copper in Wholemeal Flour
coop	Co-operative Trial in Analytical Chemistry
cpus	Performance of Computer CPUs
crabs	Morphological Measurements on Leptograpsus Crabs
deaths	Monthly Deaths from Lung Diseases in the UK
drivers	Deaths of Car Drivers in Great Britain 1969-84
eagles	Foraging Ecology of Bald Eagles
epil	Seizure Counts for Epileptics
farms	Ecological Factors in Farm Management
fgl	Measurements of Forensic Glass Fragments
forbes	Forbes' Data on Boiling Points in the Alps
galaxies	Velocities for 82 Galaxies
gehan	Remission Times of Leukaemia Patients
genotype	Rat Genotype Data
geyser	Old Faithful Geyser Data
gilgais	Line Transect of Soil in Gilgai Territory
hills	Record Times in Scottish Hill Races
housing	Frequency Table from a Copenhagen Housing Conditions Survey

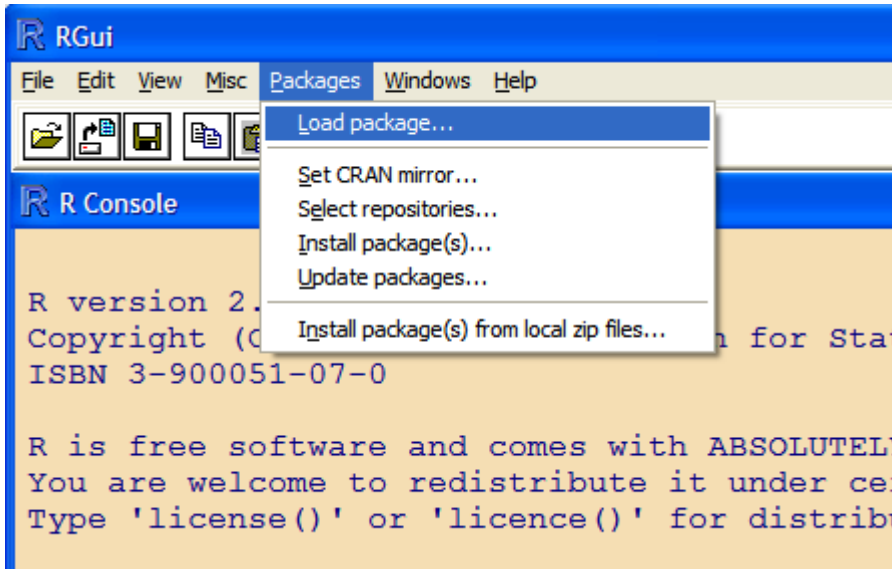
immer	Yields from a Barley Field Trial
leuk	Survival Times and White Blood Counts for Leukaemia Patients
mammals	Brain and Body Weights for 62 Species of Land Mammals
mcycle	Data from a Simulated Motorcycle Accident
menarche	Age of Menarche in Warsaw
michelson	Michelson's Speed of Light Data
minn38	Minnesota High School Graduates of 1938
motors	Accelerated Life Testing of Motorettes
muscle	Effect of Calcium Chloride on Muscle Contraction in Rat Hearts
newcomb	Newcomb's Measurements of the Passage Time of Light
nlschools	Eighth-Grade Pupils in the Netherlands
npk	Classical N, P, K Factorial Experiment
npr1	US Naval Petroleum Reserve No. 1 data
oats	Data from an Oats Field Trial
painters	The Painter's Data of de Piles
petrol	N. L. Prater's Petrol Refinery Data
phones	Belgium Phone Calls 1950-1973
quine	Absenteeism from School in Rural New South Wales
road	Road Accident Deaths in US States
rotifer	Numbers of Rotifers by Fluid Density
ships	Ships Damage Data
shoes	Shoe wear data of Box, Hunter and Hunter
shrimp	Percentage of Shrimp in Shrimp Cocktail
shuttle	Space Shuttle Autolander Problem
snails	Snail Mortality Data
steam	The Saturated Steam Pressure Data
stormer	The Stormer Viscometer Data
survey	Student Survey Data
synth.te	Synthetic Classification Problem
synth.tr	Synthetic Classification Problem
topo	Spatial Topographic Data
waders	Counts of Waders at 15 Sites in South Africa
whiteside	House Insulation: Whiteside's Data
wtloss	Weight Loss Data from an Obese Patient

Use `'data(package = .packages(all.available = TRUE))'`  
to list the data sets in all `*available*` packages.

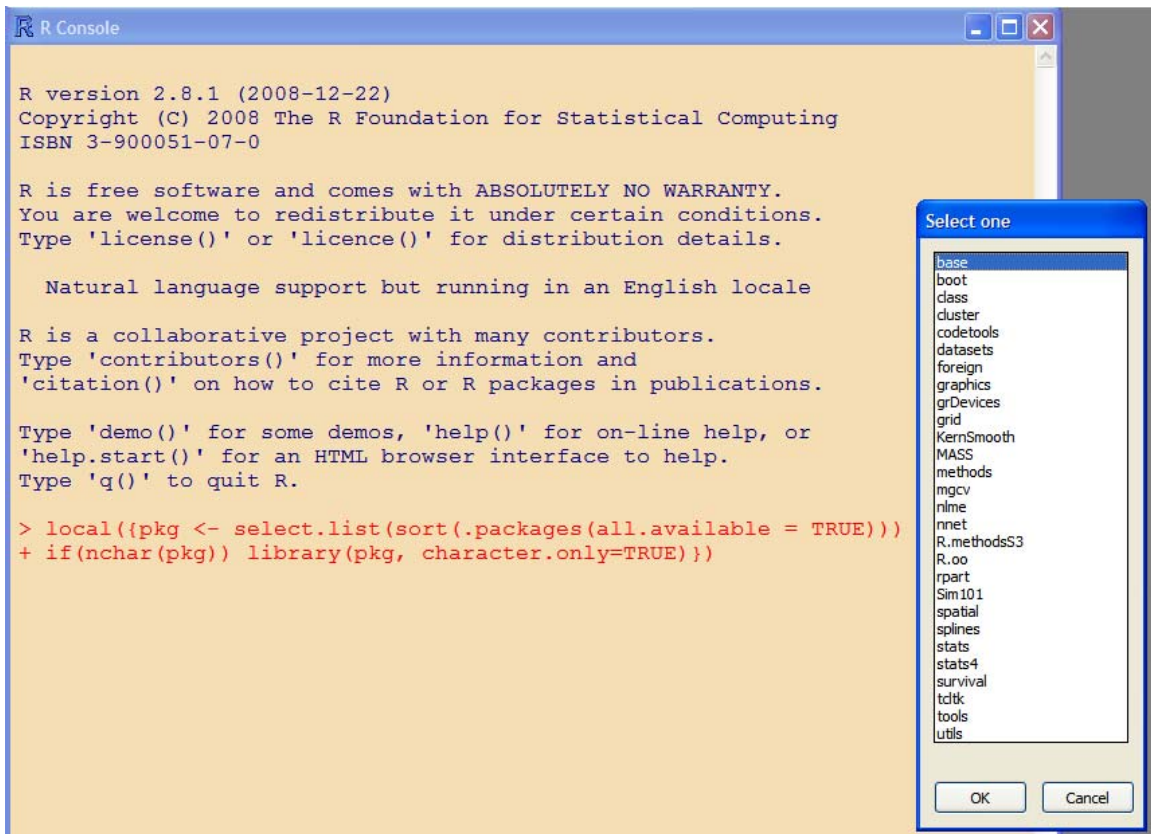
وإضافة مجموعة بيانات إلى حزمة

**> data(mydata, package=MASS)**

الحوار التالي لتحميل حزمة



تظهر نافذة تعرض جميع الحزم المتوفرة للاختيار منها



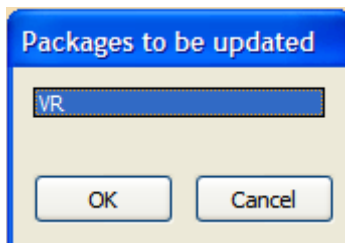
```
> update.packages(ask='graphics')
--- Please select a CRAN mirror for use in this session ---
```



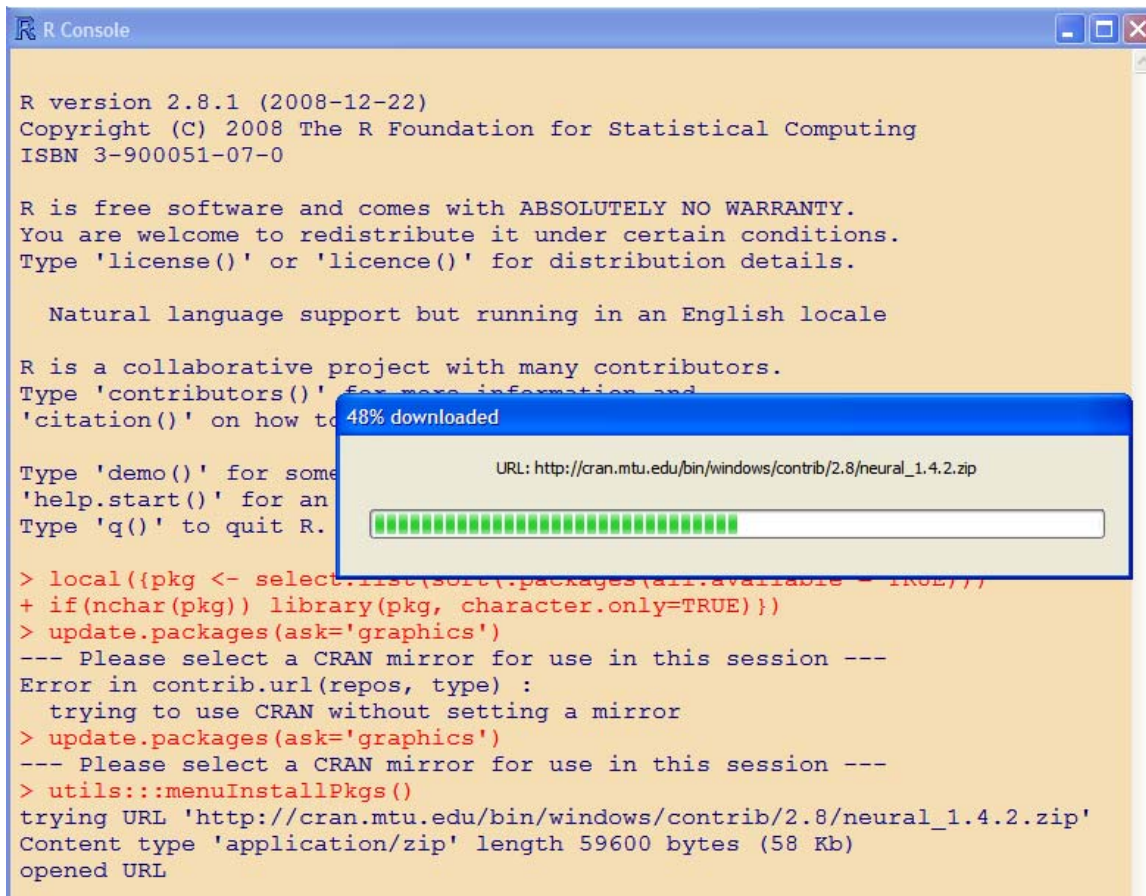
بعد إختيار الحزمة يطلب البرنامج إختيار الجهة التي سيتم التحميل منها



ثم يعرض أسماء الحزم الموجودة والتي تحتاج لتجديد



ثم يقوم البرنامج بإنزال وتجهيز الحزمة



```
R Console
R version 2.8.1 (2008-12-22)
Copyright (C) 2008 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some interactive examples.
Type 'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> local({pkg <- select.list(sort(.packages[all.available = TRUE]))
+ if(nchar(pkg)) library(pkg, character.only=TRUE)})
> update.packages(ask='graphics')
--- Please select a CRAN mirror for use in this session ---
Error in contrib.url(repos, type) :
trying to use CRAN without setting a mirror
> update.packages(ask='graphics')
--- Please select a CRAN mirror for use in this session ---
> utils:::menuInstallPkgs()
trying URL 'http://cran.mtu.edu/bin/windows/contrib/2.8/neural_1.4.2.zip'
Content type 'application/zip' length 59600 bytes (58 Kb)
opened URL
```

```
> update.packages(ask='graphics')
--- Please select a CRAN mirror for use in this session ---
Warning: package 'VR' is in use and will not be installed
trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/cluster_1.11.13.zip'
Content type 'application/zip' length 518967 bytes (506 Kb)
opened URL
downloaded 506 Kb

trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/lattice_0.17-22.zip'
Content type 'application/zip' length 926180 bytes (904 Kb)
opened URL
downloaded 904 Kb

trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/survival_2.35-4.zip'
Content type 'application/zip' length 2694031 bytes (2.6 Mb)
opened URL
```

downloaded 2.6 Mb

```
package 'cluster' successfully unpacked and MD5 sums checked
package 'lattice' successfully unpacked and MD5 sums checked
Warning: cannot remove prior installation of package 'lattice'
package 'survival' successfully unpacked and MD5 sums checked
```

```
The downloaded packages are in
      C:\Documents and Settings\amb\Local
Settings\Temp\Rtmp1dGcgZ\downloaded_packages
updating HTML package descriptions
>
```

```
> utils:::menuInstallPkgs()
```

```
trying URL
'http://cran.mtu.edu/bin/windows/contrib/2.8/neural_1.4.2.zip'
Content type 'application/zip' length 59600 bytes (58 Kb)
opened URL
downloaded 58 Kb
```

```
package 'neural' successfully unpacked and MD5 sums checked
```

```
The downloaded packages are in
      C:\Documents and Settings\amb\Local
Settings\Temp\Rtmp1dLn2q\downloaded_packages
updating HTML package descriptions
```

```
> library(neural)
> library(help = neural)
>
```

Information on package 'neural'

Description:

```
Package:      neural
Version:     1.4.2
Title:       Neural Networks
Author:      Ádám Nagy
Maintainer:  Ádám Nagy <sodzsus@yahoo.com>
Description: RBF and MLP neural networks with graphical user
             interface
License:     GPL version 2 or later
Built:      R 2.8.1; ; 2008-12-14 01:45:38; windows
```

Index:

```
mlp                MLP neural network recalling method
```

mlptrain	MLP neural network training method
rbf	RBF neural network recalling method
rbftrain	RBF neural network training method
letters_train	dataset for pattern recognizing
letters_out	dataset for pattern recognizing
letters_recall	dataset for pattern recognizing

## الصفوف والمصفوفات Arrays and matrices:

الصف في R يتكون من متجة بيانات والذي يعطي مكونات مصفوفة ومتجة بعد. والمصفوفة هي صف ذو بعدين. متجه البعد يمتلك عنصرين وهي عدد الأسطر وعدد الأعمدة. يمكن توليد مصفوفة بالدالة `array(...)` مع الحجة `dim` توضع كمتجه ذا طول 2. أو يمكن إستخدام الدالة `matrix(...)` مع ملاحظة أن الأعمدة تحدد أولاً

```
> m1 <- array(1:15,dim = c(3,5))
> m1
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15
> m2 <- matrix(1:15,3,5)
> m2
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15

> m3 <- 1:15
> dim(m3) <- c(3,5)
> m3
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15

> c(class(m1), class(m2), class(m3))
[1] "matrix" "matrix" "matrix"
> array(1:15, dim=c(3,6)) # recycling rule used
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    4    7   10   13    1
```

```

[2,] 2 5 8 11 14 2
[3,] 3 6 9 12 15 3
> array(1:15, dim=c(2,4)) # extra values discarded
      [,1] [,2] [,3] [,4]
[1,] 1 3 5 7
[2,] 2 4 6 8

```

إستخدام العمال الحسابية على صفوف متشابهة سينتج عنه إجراء حسابات عنصر بعنصر وهذا لأبأس به في جمع مصفوفة مع أخرى ولكنه غير صحيح في عملية الضرب

```

> m1+m1
      [,1] [,2] [,3] [,4] [,5]
[1,] 2 8 14 20 26
[2,] 4 10 16 22 28
[3,] 6 12 18 24 30
> m1*m1
      [,1] [,2] [,3] [,4] [,5]
[1,] 1 16 49 100 169
[2,] 4 25 64 121 196
[3,] 9 36 81 144 225

```

المصفوفات المتوافقة *conformal matrices* (وهي تلك التي يكون عدد الأعمدة في الأولى يساوي عدد السطور في الثانية) يمكن أن يوجد حاصل ضربها بإستخدام العامل `%*%`. قلب المصفوفة أو عكسها وكذلك حل نظام خطي من المعادلات يمكن أن يتم بإستخدام الدالة `solve(...)` فمثلا لكي نحل النظام  $ax = b$  نستطيع أن نستخدم `solve(A,b)` والدالة `solve(A)` تعطي مقلوب **A**.

```

> m4 <- array(1:3, c(4,2))
> m4
      [,1] [,2]
[1,] 1 2
[2,] 2 3
[3,] 3 1
[4,] 1 2

> m5 <- array(3:8, c(2,3))
> m5
      [,1] [,2] [,3]
[1,] 3 5 7
[2,] 4 6 8

```

```

> m4 %*% m5
      [,1] [,2] [,3]
[1,]   11   17   23
[2,]   18   28   38
[3,]   13   21   29
[4,]   11   17   23

> m6 <- array(c(1,3,2,1),c(2,2))
> m6
      [,1] [,2]
[1,]    1    2
[2,]    3    1
> v1 <- array(c(1,0), c(2,1))
> v1
      [,1]
[1,]    1
[2,]    0
> solve(m6,v1)
      [,1]
[1,] -0.2
[2,]  0.6

> solve(m6) # inverts m6
      [,1] [,2]
[1,] -0.2  0.4
[2,]  0.6 -0.2

> solve(m6) %*% v1 # does the same as solve(m6,v1)
      [,1]
[1,] -0.2
[2,]  0.6

```

الضرب الخارجي outer product لصفين هو مصفوفة تتولد من كل تراكيب ممكنة لعناصرهم. الحجتين الأولى والثانية في الدالة **outer(...)** هي المتجهين والحجة الثالثة هي العملية المطلوب القيام بها (القيمة الافتراضية هي الضرب) وهنا نحن نبسط الأمر لأن الضرب الخارجي يعمل على متجهات بأبعاد عامة (وليس متجهات وحسب) والمثال التالي يوضح ذلك. المطلوب توليد دالة الكتلة للمصفوفة 2X2 والتي عناصرها تأتي من توزيع متساوي متقطع على المجموعة {0,...,5} المحددة هي على الشكل AD - BC

حيث A و B و C و D موزعة توزيع متساوي. دالة الكتلة يمكن حسابها عن طريق إحصاء جميع القيم الممكنة الناتجة وتستخدم الدالة **table(...)** التي تكون جدول توزيع تكراري لحجته

```
> m7 <- outer(0:5,0:5) # every possible value of AD
and BC
```

```
> m7
```

```
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    0    0    0    0    0    0
[2,]    0    1    2    3    4    5
[3,]    0    2    4    6    8   10
[4,]    0    3    6    9   12   15
[5,]    0    4    8   12   16   20
[6,]    0    5   10   15   20   25
```

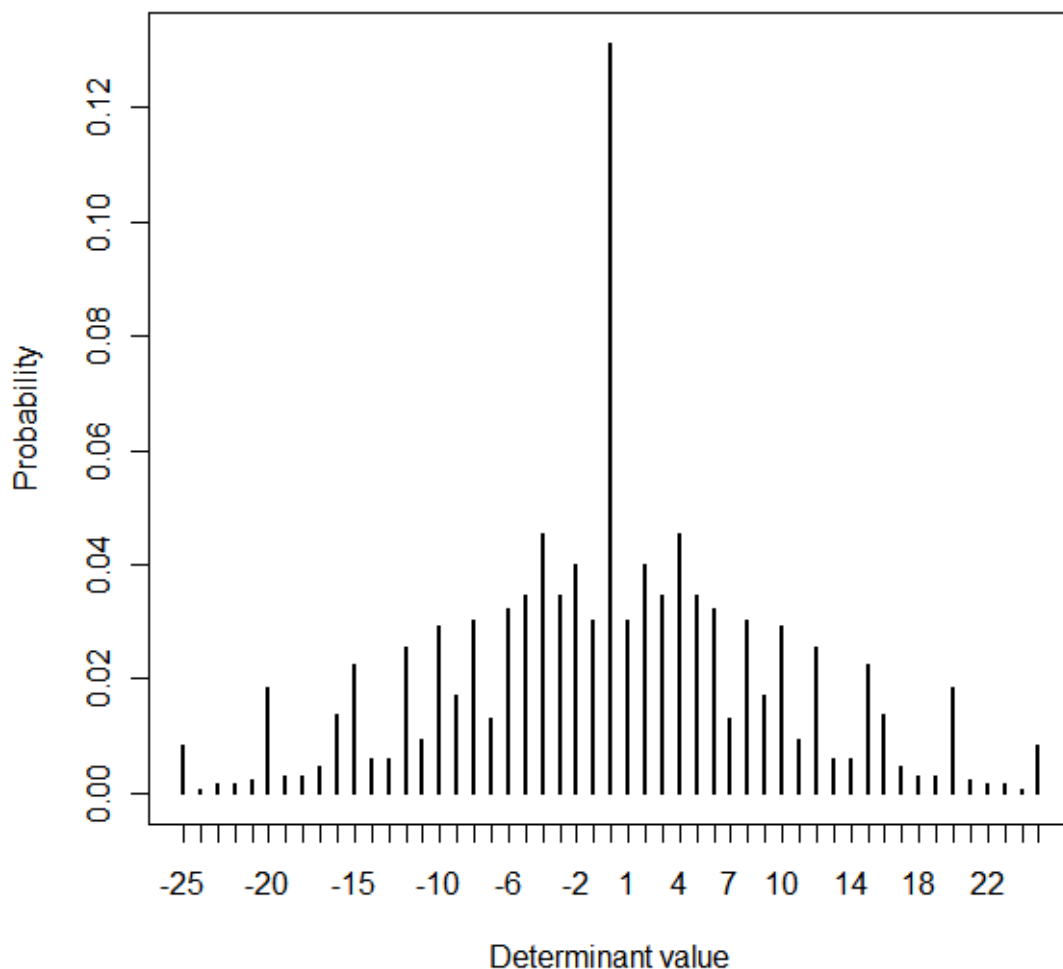
```
> freq <- table(outer(m7,m7,"-")) # frequency for
all values of AD-BC
```

```
> freq
```

```
> freq
```

```
-25 -24 -23 -22 -21 -20 -19 -18 -17 -16 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4
 11  1  2  2  3 24  4  4  6 18 29  8  8 33 12 38 22 39 17 42 45 59
-3 -2 -1  0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
 45 52 39 170 39 52 45 59 45 42 17 39 22 38 12 33  8  8 29 18  6  4
 19 20 21 22 23 24 25
  4 24  3  2  2  1 11
```

```
> plot(freq/sum(freq), xlab="Determinant value",
+ylab = "Probability")
```



يوجد العديد من العمليات على المصفوفات في R مثل **t(...)** لإيجاد المنقول و **transpose** و **nrow(...)** وتعطي عدد الأسطر و **ncol(...)** تعطي عدد الأعمدة والدالة **rbind(...)** و **cbind(...)** تربط الأسطر او الأعمدة معا. والدالة **eigen(...)** لإيجاد الجذور والمتجهات المميزة *eigenvectors and eigenvalues* للمصفوفة وتقوم بتفكيك القيمة الشاذة *singular value decomposition* مستخدمين الدالة **svd(...)** وهكذا.



## إدخال بيانات من الإنترنت :Data from the Internet

نستخدم **read.table()** لجلب وإدخال بيانات من الإنترنت كالتالي:

```
> data <-read.table("http://www.stat.ucla.edu/~vlew  
/stat130a/datasets/twins.csv",header=TRUE,sep=" , ")
```

حيث **header=TRUE** تخبر R بأن البيانات تشمل أسماء المتغيرات في أول سطر.

و **sep=" , "** تخبر R أن البيانات مفصولة بواسطة فاصلة ( " , " ).

## Glossary

- Basic arithmetic operators: `()` brackets in arithmetic calculations  
^ raise to the power of  
/ division  
\* multiplication  
+ addition  
- subtraction
- Assignment operator `<-` assigns the value of the expression on the right to the identifier on the left.
- Concatenation function containing those values. `c(...)` argument is a set of values; returns a vector
- Simple statistical functions **sum(...)**  
**mean(...)**  
**median(...)**  
**range(...)**  
**sd(...)** standard deviation  
**mad(...)** mean absolute deviation  
**IQR(...)** inter-quartile range  
**min(...)** minimum  
**max(...)** maximum
- Help `? ...` will provide information on function usage and arguments  
**help.start()** opens html help files
- Session management **objects()** lists objects in workspace  
**ls()** does the same as `objects()`  
**rm(...)** removes the objects listed from workspace  
**rm(list = objects())** removes all objects from workspace  
**getwd()** returns the current working directory  
**setwd(...)** set the working directory  
**dir()** list files in current directory  
**save.image(...)** saves workspace to specified file  
**quit()** ends R session  
**load(...)** loads a saved workspace from file
- Load data from text file **read.table(...)** argument is a file in the working directory. Return value is a data frame.
- Editing data **fix(...)** or **data.entry(...)** brings up spreadsheet like environment for editing data.
- Scatterplot **plot(...)** is a versatile plotting command, draws a scatterplot when two arguments are given.
- Linear model fitting **lm(...)** can fit a variety of linear models, argument is a model formula.
- Basic inference **t.test(...)** Student's t-test  
**var.test(...)** F test to compare two variances  
**chisq.test(...)** Pearson's  $\chi^2$  test for count data

- Object properties **cor.test(...)** test for association  
**length(...)**  
**mode(...)**  
**typeof(...)**  
**class(...)**
- Object contents **attributes(...)** returns all non-intrinsic attributes  
**summary(...)** information on an object; output depends on class  
**str(...)** compact details of object structure  
**dput(...)** text listing of object
- Regular sequences : sequence generator gives sequence with unit steps between left and right arguments  
**seq(...)** more general regular sequence generator, can set step (by) and/or length of sequence (length)  
**rep(...)** for generating sequence with repeated values
- Logical values TRUE and FALSE
- Comparison and logical operators == equality  
!= inequality  
< less than  
> greater than  
<= less than or equal to  
>= greater than or equal to  
! negation  
& and (&& scalar version)  
| or (|| scalar version)
- Accessing elements [...] contents of brackets a vector of integers, a vector of logical values or a vector of names.
- Accessing a list component [[...]] contents of brackets a number or a name.
- Accessing a named component \$
- Referring to list components directly **attach(...)** brings components of a list into view; use **detach()** to tidy up.
- Generating a vector of strings **paste(...)** converts any non-character arguments into characters and puts them together into a vector of mode character.
- Generating lists **list(...)** put arguments (which may be named) into a list.
- A selection of plots **hist(...)** histogram  
**qqnorm(...)** normal qq-plot  
**boxplot(...)**  
**contour(...)** 3D contours  
**image(...)** 3D coloured blocks  
**persp(...)** 3D wire frame
- Some low level plotting commands **points(...)**  
**lines(...)**  
**text(...)**

- Some graphics parameters **pch** plotting character  
**col** colour  
**mfrow** multi-figure page specification **mfc**
- Storing commands **savehistory(...)**  
**loadhistory(...)**  
**source(...)**
- Packages **library(...)** list and load packages  
**data(...)** add data from a package to workspace
- Search path **search()** lists environments in the search path in the order in which they will be searched. Passing the position in search path or name of an environment to **objects(...)** lists objects in that environment.
- Arrays **matrix(...)** to generate a matrix (2D array)  
**array(...)** to generate a general array.