لغة البرمجة R

(1) الحوسبة الإحصائية Statistical Computing

تمثل الاستخدام الواسع من الطرق والاساليب والتي تشمل مختلف مهارات علم الاحصاء وعلم الحاسبات لحل المشاكل وتحليل الظواهر الطبيعية أو الاجتماعية.

كما تؤكد على تأثير الحوسبة على الإحصاءات والعكس بالعكس. والعمل في مجموعة متنوعة من المجالات في الإحصاءات، بما في ذلك القياسات الحيوية، الاقتصاد القياسي، وتحليل البيانات، والرسومات، والمحاكاة، والخوارزميات، والنظم القائمة على المعرفة، والحوسبة النظرية الافتراضية. اضافة الى الأساليب الإحصائية المكثفة حسابيا بما في ذلك اختزال طرق، سلسلة ماركوف مونتي كارلو الأساليب والانحدار ، الشبكات العصبية الاصطناعية .

وان الاصدار الاول للغة R عام 1993، حيث صُممت من قبل روس إلهاكا، وروبرت جينتلمان خبيري الاحصاء بجامعة اوكلاند في نيوزيلندا وقد سُميت بلغة R لانها ترمز الى الحرف الاول من اسميهما ومن البرامج المتخصصة بالحوسبة الاحصائية هي : Matlab , Minitab , Excel , SPSS , S , R

الاحصاءات الحسابية 'computational statistics' الذي يهدف الى تصميم خوارزمية لتنفيذ الاساليب الاحصائية على اجهزة الكمبيوتر.

(2) مميزات لغة البرمجة R

R هو أقوى لغة برمجة في العالم للحوسبة الإحصائية، والتعلم الآلي والرسومات ومن ميزاتها :

1- متوفرة في موقع مجاني ومفتوح المصدر لغرض تنصيبها على حاسبتك ،ومتاحة على شبكة الانترنت من مصادر وكتب الكترونية وبرامج جاهزة تُنفذ مهمة ما ، وتحتوي على مجموعة حزم مفتوحة المصدر مطورة من طرف اناس ومجموعات ابحاث.

2- Rهي لغة وبيئة تطوير (Language and development environment) متخصصة في تحليل وتمثيل البيانات والحوسبة الإحصائية.

3- ان إحدى نقاط قوة لغة R هي سهولة توسعتها بالاستفادة من مجموعة الإضافات الهائلة المتاحة لها، والتي قام بتطويرها الآلاف من الجامعات والمراكز العلمية وحتى الباحثون المستقلون وطلاب الدراسات العليا

4- تتمتع بصفات لغات البرمجة الاخرى حيث يمكن القيام بالبرمجة الكائنية وحتى انشاء صفحات ويب.

5- أنها لغة سكريبت Scripting Language فلا تحتاج لكتابة دالة دخول كما هو في لغات البرمجة الكلاسيكية ، حيث يُعتبر اول سطر هو اول امر ويكون التنفيذ مباشرة بدون ترجمة

6- تعمل مع عدة أنظمة تشغيل مثل نظام لينوكس ويونكس وماك وويندوز .

7- لغة احصائية تعمل على حل جميع المسائل الاحصائية وباسلوب سهل مثلا استعمل الامرsum لحساب المجموع والامر var لحساب التباين والامر mean لحساب المعدل ....وهكذا

8- تحتوي على إجراءات إحصائية متقدمة غير متوفرة في مجموعات أخرى ولها قدرات رسومية عالية للمخططات ثلاثية الابعاد فضلاً على امكانية اظهار رسومات جذابة بسهولة

9- R حساسة لحالة الاحرف

10- لها قابلية لبرمجة الحسابات المتوازية parallel computing حيث يمكن حل عمليات الحوسبة المعقدة والنظم الاحيائية

11- تعطي النتائج التي تحتاجها وبشكل مبسط دون تكديس المخرجات خلافاً لبرمجيات احصائية اخرى مثل برنامج SPSS .

(3) مجالات الاستخدام

في مجالات الحياة اليومية نبحث دائما عن تقدير الاشياء والمتمثلة بالارقام والاحصائيات لتأكيد صحتها واتخاذ القرارات المناسبة لها. وفيما بعض الامثلة لمجالات عمل الحوسبة الاحصائية

1) التلوث الصناعي للتربة والهواء والماء ودراسة تأثيرها على صحة الانسان بطرق احصائية
2) دراسة وتحليل الجينات ومقارنتها وتصنيفها للسيطرة مستقبلا على امراض مستعصية مثل السرطان .
3) العمل في شركات العقاقير والادوية بالاعتماد على الدراسات الاحصائية لتاثير الدواء وفاعليته.
4) اعتماد الدراسة في الاقتصاد والاعمال وتوقعات السوق المالية على التحليل الاحصائي لمؤشرات العرض والطلب والاستهلاك .
5) تطبيقات الشبكات والانترنت تاخذ حيزاً كبيرا في مناهج الحوسبة الاحصائية.

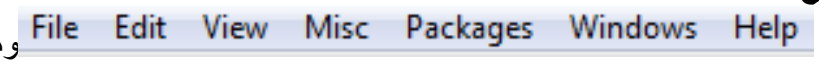
لغة البرمجة R

(4) تنصيب وتشغيل لغة R

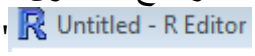
ويتم من الموقع الرسمي للغة على الرابط http://www.r-project.org

ويكون التشغيل بالنقر المزدوج على أيقونة R على سطح المكتب ، فتظهر الشاشة الافتتاحية بسطر الاوامر الذي يبدأ بعلامة التنصيص (>)

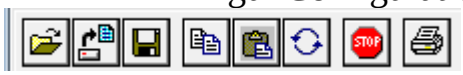
(5) واجهة برنامج R

1- شريط القوائم  ومن اهم الايعازات:

New script : لفتح مكان في البرنامج R لكتابة عبارات برنامج فقط دون تنفيذ :

- من قائمة File نختار New script ، تظهر شاشة "  " ، حيث يمكن نقل العبارات الى فضاء العمل بايعازي النسخ واللصق او ctrl+r .

GUI preferences… : لتغيير مواصفات شاشة العرض في R، من القائمة Edit نختار GUI preferences… يظهر مربع حوار "Rgui Configuration Editor "

2- شريط الادوات Toolbar 

ويتكون من الايقونات المهمة والاكثر استخدام وهي :

(1) Open script وتعني فتح الملف في برنامج R

(2) Load workspace تحميل فضاء العمل

(3) Save workspace حفظ فضاء العمل

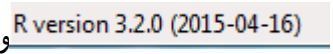
(4) Copy تسخ

(5) Paste لصق

(6) Copy and paste نسخ ولصق

(7) Stop current computation وقف الحساب الحالي

(8) Print طباعة

3- شريط الحالة statusbar  ويظهر اسفل الشاشة يبين نوع الاصدار لـR والسنة

(6) الاوامر الخاصة في لغة البرمجة R

لكل لغة من لغات البرمجة عدد من الرموز الخاصة التي تُميزها عن اللغات الاخرى ، وتعتبر قواعد هامة يجب الالتزام باستخدامها الصحيح وهي :

1- الرمز (<-) : يشار الى عملية الإسناد في لغة R ، وهي الطريقة الأكثر شيوعا مقارنة برمز المساواة = والذي يصح استخدامه ايضاً

2- الفارزة المنقوطة (;) تستخدم لفصل الاوامر في السطر الواحد ، او يكتب الامر التالي في سطر جديد

3- العبارة التعريفية : كل شيء بعد الرمز # (هاش) الى نهاية السطر يعتبر عبارة تعريفية او تعليق للمستخدم ويتم تجاهله من قبل لغة R . لكن الغريب أن لغة R تفتقر إلى طريقة لجعل مقطع كامل يعامل معاملة التعليقات (كما هو حال استخدام أسلوب التأطير /* ... */ في العديد من لغات البرمجة الأخرى).

4- يمكن ان يكون الامر او العبارة في لغة R على عدة اسطر حيث تعطي R العلامة + في بداية السطر للدلالة على أن المدخل حتى الآن لم يكتمل تعريفه ،كما في المثال التالي:

```
> for (x in 1:3){
 +print(x)
+}
```

5- يمكن كتابة الاوامر على الدوال داخل اقواس مستديرة ،على سبيل المثال :

```
> (2^(1000) rnorm)
```

6- تُطبع قبل النتيجة العلامة "[1]" وهذا لأن R يعتبر افتراضيا كل شيء بمثابة جدول "vector "والرقم واحد هو مؤشر عن العنصر الأول في الجدول.

```
> 1:70
 [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 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
[51] 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70
```

7- ان ناتج تنفيذ أي أمر أو دالة بلغة R يظهر بعدها مباشرة، وهكذا تتكون جلسة العمل الاعتيادية من تنفيذ لتتالي من الأوامر والتعليمات وصولا إلى إنجاز العمل أو التحليل المطلوب.

8- يوفر R آلية لإعادة تنفيذ-الأوامر السابقة وذلك باستخدام مفاتيح الأسهم على لوحة المفاتيح للتنقل إلى الأمام والخلف من خلال الأوامر السابقة الاستخدام ،كما يمكن للمؤشر ان ينتقل ضمن الأوامر باستخدام مفاتيح الأسهم الأفقية، والاحرف يمكن ازالتها مع مفتاح DEL او اضافتها من مفاتيح اخرى .ويمكنك باستخدام أزرار الأسهم إلى الأعلى وإلى الأسفل للتنقل عبر مجموعة الأوامر التي تم تنفيذها .

9- دالة الربط والتي لها البناء اللغوي التالي: c(object1,….) ، تُستخدم بشكل كبير لوضع قيم ذات نوع واحد في شكل متجه

e.x: >x<-c(1,2,3,"a")
> x

(للتنفيذ او الانتقال الى سطر جديد نضرب مفتاح الادخال Enter)

[1] "1" "2" "3" "a"

(10) R حساسة لحالة الاحرف فـa و A نوعان من كائنات مختلفة وعلى سبيل المثال :

```
> x=1:50
> MEAN(x)
Error: could not find function "MEAN"
> mean(x)
[1] 25.5
```

(7) انهاء R

لانهاء برنامج R نطبع الامر q()< يظهر مربع حوار يطلب حفظ البيانات

(8) ايعازات عامة

1- للاستفسار عن ايعاز ما او أي دالة اسمها محدد نستخدم الامر : help(…)< حيث نضع بين القوسين الايعاز او الكلمة التي نُريد الاستفسار عنها. e.x.>help(mean) او help(solve)<

او نكتب علامة الاستفهام في البداية. e.x.>?mean او ?solve<

2- الأمر example(…) لرؤية الامثلة. e.x.>example(mean)

3- الدالة history() : لاستعراض آخر 25 أمر .

4- الدالة ls() لمشاهدة كل الكائنات في الجلسة الجارية والمخزونة في R

5- الاختصار ctrl+L : لمسح الشاشة المعروضة بدون مسح اي بيانات في الذاكرة.اما الدالة rm(x) < فهو لمسح x من الذاكرة ، حيث ان x قد تكون متغير او متجه او مصفوفة او اطار بيانات والدالة

rm(list=ls())< لمسح كل ما يوجد الان في ذاكرة العمليات الحالية من بيانات تم تحميلها فيما سبق

6- الدالة getwd() : لمعرفة دليل العمل working directory

7- الدالة dir() والدالةlist.files() لعرض كل الملفات في دليل العمل .

8- الدالة options(…) لتحديد عرض خط النتائج وكما يلي:

```
> options(width=40)
> 1:50
 [1]  1  2  3  4  5  6  7  8  9 10 11 12
[13] 13 14 15 16 17 18 19 20 21 22 23 24
[25] 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] 49 50
```

لغة البرمجة R

(9) العمليات الحسابية

تُستخدم R كالة حاسبة لتنفيذ العمليات الحسابية (+ للجمع ،- للطرح، * للضرب ،/للقسمة ، ^او ** للاس، %% الباقي من القسمة، %/% للقسمة الصحيحة .

```
> 5 + 3
[1] 8
> 15.3 * 23.4
[1] 358
> sqrt(16)
[1] 4
```

```
> v=c(2,5,7)
> t=c(8,4,9)
> v+t
[1] 10  9 16
> v-t
[1] -6  1 -2
> v*t
[1] 16 20 63
> v/t
[1] 0.2500000 1.2500000 0.7777778
> v%%t
[1] 2 1 7
> v%/%t
[1] 0 1 0
> v^t
[1]       256       625 40353607
```

```
> 2+7.2-5*3/2+2^4
[1] 17.7
> 31%%7
[1] 3
> 16%/%5
[1] 3
```

(10) الدوال الرياضية

- abs(x) والتي تعيد القيمة المطلقة
- sqrt(x) التي تحسب الجذر التربيعي او x^0.5 .
- الدوال المثلثية المختلفة مثل sin(x) و cos(x) وغيرهما.

e.x.>x=1:5
>y=sin(x)+cos(x)

- دوال التحويل مثل log(x) التي تحسب اللوغاريتم الطبيعي للمقدار x.
- log10(x) اللوغاريتم العشري لذات المقدار x.
- log(x,n) والتي تحسب اللوغاريتم لأي أساس يحدده المبرمج من خلال المقدار n، فمثلا يمكنك حساب اللوغاريتم الثنائي للمقدار x باستخدام التعليمة log(x,2).
- exp(0) >

```
> sqrt(2)
[1]1.414214
> cos(pi)
[1] -1
> sin(20)^2+cos(20)^2
[1] 1
> log(1)
[1] 0
> log10(10)
[1] 1
> exp(0)
[1] 1
```

(11) دوال احصائية

- Sum() دالة الجمع
- mean() لطباعة المتوسط الحسابي
- max() لطباعة القيمة العظمى
- min() لطباعة القيمة الصغرى
- median() الوسيط وهو القيمة التي تقع في المنتصف عند ترتيب القيم تصاعديا، وعادة ما يستخدم الوسيط للدلالة على مركز المجموعة).
- var() تحسب مقدار التباين من العلاقة $\sum (x_i - \bar{x})^2$
- sd() لطباعة الانحراف المعياري
- **الدالة summary()** فهي عامة الاستخدام ويكون ناتج تنفيذها ملخص لبعض الدوال الاحصائية

```
> x=c(1,2,3,4,5)
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
      1       2       3       3       4       5
```

(12) عمليات المقارنة Comparison Operations

تستخدم للمقارنة بين القيم. كما في الجدول المجاور:

العملية	الوصف
<	اقل من
>	اكبر من
<=	اقل من او يساوي
>=	اكبر من او يساوي
==	يساوي
!=	لا يساوي

امثلة :

```
> v=c(2,5,7)
> t=c(8,4,9)
> v>t
[1] FALSE  TRUE FALSE
> v<t
[1]  TRUE FALSE  TRUE
> v==t
[1] FALSE FALSE FALSE
> v>=t
[1] FALSE  TRUE FALSE
> v<=t
[1]  TRUE FALSE  TRUE
> v!=t
[1] TRUE TRUE TRUE
```

```
> m<-1:5; m==4
 [1] FALSE FALSE FALSE  TRUE FALSE
```

(13) العمليات المنطقية Logical Operations

في الجدول ادناه المعاملات المنطقية ووصف لكل منها :

العملية	الوصف
!	لا
&	و
|	او

```
> x <- c(TRUE,FALSE,0,6)
 > y <- c(FALSE,TRUE,FALSE,TRUE)
> !x
 [1] FALSE  TRUE  TRUE FALSE
 > x&y
 > x|y
 [1]  TRUE  TRUE FALSE  TRUE
```

(14) أسبقية التشغيل

ان تنفيذ العمليات الحسابية او الدوال في التعبير الواحد، يتم حسب تسلسل العملية التي ستجرى. وتحدد الأسبقية في ترتيب التنفيذ من قبل لغة R ، وفي حالة وجود تعبيران او اكثر بنفس الاسبقية فيتم حساب التعبير بين الاقواس اولاً ان وجدت وبعدها نأخذ بنظر الاعتبار الترتيب في التنفيذ مثلاً من جهة اليسار الى يمين التعبير والجدول التالي يوضح اولويات التنفيذ للعمليات:

الوصف	العملية
الأس	^
باقي القسمة	%%
الضرب، القسمة	*, /
الجمع، الطرح	+, -
مقارنات	<, >, <=, >=, ==, !=
المنطقي NOT	!
منطقية AND	&
منطقية OR	|
احالة نحو اليمين	->, ->>
احالة نحو اليسار	<-, <<- , =

e.x.: > 3 / 4 / 5
[1] 0.15
> 3 / (4 / 5)
[1] 3.75

في المثال الاول، يكون الناتج حسب اسبقية التنفيذ من اليسار الى اليمين لنفس عملية القسمة اما في المثال الثاني فالناتج يختلف بسبب تنفيذ ناتج الاقواس اولا.

لغة البرمجة R

(15) البيانات في R

في كل لغة برمجة يتم التعامل مع انواع مختلفة من البيانات والتي تكتب بصيغ خاصة بتلك اللغة وهي المتجهات Vectors ، القوائم Lists ، المصفوفات والمجموعات Matrices and Arrays الجداول Tables ، العوامل Factor وفيما يلي اهم البيانات الموجودة في R والصيغ التي تتعامل فيها .

1// المتجهات Vectors

```
a<- c(1,2,5,3,6,-2,4)   # متجه رقمي
b<- c("one","two","three")   # متجه حرفي
c<-c(TRUE,TRUE,FALSE,TRUE,FALSE)  # متجه منطقي
```

الدالة ()typeof: لمعرفة نوع المتجه

الدالة ()length: لمعرفة عدد العناصر في المتجه.

الدالة ()c : تستخدم لوضع الاشياء معاً في المتجه

```
>a[5]          #    لطباعة عنصر معين
[1]  6
>b[2]
[1] "two"
>a[c(2,4)]   #  العنصر الثاني والرابع في المتجه
[1]  2  3
>a[-3]         #لاختيار العناصر جميعا ما عدا العنصر الثاني
[1] 1 2 3 6 -2 4
> c(0, 7, 8)
[1] 0 7 8
> x <- c(7.2, 3, 9)
> x
[1] 7.2 3.0 9.0
```

الرمز لانشاء تسلسل تصاعدي او تنازلي للقيم في مثال كما يلي :

```
> a<- 5:20
> a
[1] 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

ممكن ربط المتجهات معاً :

```
> y=c(0,7,8)
> c(a, y)
[1] 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 0 7 8
 [1] 5 13 17
```

الدالة ()diff: لمعرفة الفرق بين العناصر **:**

```
x <- c(1, 4, 6, 9, 23)
diff(x)
[1]  3  2  3 14
```

الدالة **()setdiff** : لمعرفة الفرق بين عناصر متجهين :

```
x <- c(1, 4, 6, 9, 23)
y <- c(12, 4, 6, 78, 44)
setdiff(x,y)
[1]  1  9 23
```

مثال مخزون لبيانات عن الجزر : > islands

- **الدالة sort():** لترتيب البيانات

```
> x=c(2,6,1,9,4,78,67,90)
> sort(x)
[1]  1  2  4  6  9 67 78 90
> sort(c(5,0,7,1,3))
[1] 0 1 3 5 7
```

- **الدالة objects() :** لسرد الكائنات في مساحة العمل

2// القوائم Lists

القائمة هي متجه عناصره مختلفة النوع ، كما في المثال التالي :

```
> x=1:10   # a vector
> y=matrix(1:12,nrow=3)   # a matrix
> z="Bill"        # a character variable
> my.list=list(x,y,z)    # creat the list
> my.list          #view the list
[[1]]
 [1]  1  2  3  4  5  6  7  8  9 10

[[2]]
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

[[3]]
[1] "Bill"
```

> my.list[[3]] # العنصر الثالث في my.list .
[1] "Bill"

- لتسمية العناصر في القائمة

```
> names(my.list) = c("my.vector","my.matrix","my.name")
> my list
Error: unexpected symbol in "my list"
> my.list
$my.vector
 [1]  1  2  3  4  5  6  7  8  9 10

$my.matrix
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

$my.name
[1] "Bill"
```

يتم استخدام $ لسحب العناصر من القوائم بالاسم. اكتب اولاً اسم القائمة، تليها $، يليه اسم العنصر في القائمة.

```
> my.list$my.name
[1] "Bill"
```

3// المصفوفات Matrices

جميع الاعمدة في المصفوفة يجب ان تكون بنفس النوع (رقمي ،او حرفي ، الخ) وكذلك بنفس الطول ، والشكل العام للمصفوفة هو :

mymatrix -> matrix(vector , nrow= r , ncol= c , byrow= FALSE)

يشير byrow= TRUE الى ملئ الصفوف اولا اما byrow=FALSE فان المصفوفة تملأ من قبل الاعمدة (الافتراضي).

```
> y<-matrix(1:20, nrow=5,ncol=4)
> y
     [,1] [,2] [,3] [,4]
[1,]    1    6   11   16
[2,]    2    7   12   17
[3,]    3    8   13   18
[4,]    4    9   14   19
[5,]    5   10   15   20
```

- العنوان [1,] يعني الصف الاول لكل الاعمدة و [,1] يعني العمود الاول لكل الصفوف
- لمعرفة قيمة العنصر في المصفوفة : يكون بتحديد رقم الصف والعمود

```
> m[1,2]
[1] 9
```

```
> y[1,]                    # جميع القيم في الصف الاول
[1]  1  6 11 16
```

```
> y[,3]                    # جميع القيم في العمود الثالث
[1] 11 12 13 14 15
```

```
>y[-1,]                    # جميع قيم المصفوفة ماعدا الصف الاول
     [,1] [,2] [,3] [,4]
[1,]    2    7   12   17
[2,]    3    8   13   18
[3,]    4    9   14   19
[4,]    5   10   15   20
```

```
>y[2:4,1:3]                # الصفوف 2،3،4 للاعمدة 1،2،3
     [,1] [,2] [,3]
[1,]    2    7   12
[2,]    3    8   13
[3,]    4    9   14
```

- لمعرفة ابعاد المصفوفة نستخدم عبارة dim() ثم اسم المصفوفة

```
> dim(m)
[1] 2 3
```

- لجمع المصفوفتين m ,n يُكتب الايعاز m+n > ولطرح المصفوفتين يكتب الايعاز m-n >

```
> m=matrix(c(2,4,6,3,5,0),nrow=2,ncol=3)
> n=matrix(c(1,5,3,7,0,9),nrow=2,ncol=3)
> m;n
     [,1] [,2] [,3]
[1,]    2    6    5
[2,]    4    3    0
     [,1] [,2] [,3]
[1,]    1    3    0
[2,]    5    7    9
> m+n
     [,1] [,2] [,3]
[1,]    3    9    5
[2,]    9   10    9
> m-n
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]   -1   -4   -9
```

- ضرب المصفوفات اي ضرب عنصر بعنصر ويكون الايعاز : m*n >

```
> m*n
     [,1] [,2] [,3]
[1,]    2   18    0
[2,]   20   21    0
```

- الدالة t() مدور المصفوفة الناتجة عن تبديل الاعمدة بالاسطر للمصفوفة يرمز لها بالرمز $\mathbf{A}^T$

```
> m
     [,1] [,2] [,3]
[1,]    2    6    5
[2,]    4    3    0
> t(m)
     [,1] [,2]
[1,]    2    4
[2,]    6    3
[3,]    5    0
```

- المصفوفة الاحادية : تستخدم الدالة diag()

```
> m=diag(3)
> m
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

- المصفوفة الصفرية : لانشاء مصفوفة باربعة اعمدة وثلاث صفوف ، تحتوي على قيمة 0 ، نكتب الايعاز : > matrix(0,nrow=3,ncol=4)

او بصورة اخرى يُكتب كالاتي :

```
> matrix(0,3,4)
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
```

- تستخدم الدالة (solve) لحل المعادلات الخطية في R كما في المثال التالي :

$3x_1+2x_2-x_3=1$
$2x_1-2x_2+4x_3=-2$
$-x_1+0.5x_2-x_3=0$

$$\begin{pmatrix} 3 & 2 & -1 \\ 2 & -2 & 4 \\ -1 & 0.5 & -1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}=\begin{pmatrix} 1 \\ -2 \\ 0 \end{pmatrix}$$, A*x=b ,A^{-1}*b=x

```
>A=array(c(3,2,-1,2,-2,0.5,-1,4,-1),c(3,3))
>b=c(1,-2,0)
>solve(A,b)
[1]  1 -2 -2
```

(16) حفظ واستدعاء كائن

في R يمكن تخزين كل كائن واستعادته من ملف مع الأوامر حفظ save وتحميل load

```
>x <- 1:4
 >save(x,file="x.Rdata")          لخزن x
 rm(x)                  لمسح x من الذاكرة
  x
                                  Error :object 'x' not found      يظهر الخطأ : الكائن " x " لا يوجد
load("x.Rdata")                   لتحميل x
 x
 [1] 1 2 3 4
```

الدالة ()getwd لمعرفة المسار التي تحصل فيه على النتائج او المجلد التي قمت بتخزين بياناتك فيه.
الدالة ()library لعرض جميع الحزم المتوفرة في R ، ويُعرض محتوى الحزمة
الدالة ()data فهي لعرض كل مجموعات البيانات في حزمة

```
> data(CO2)    #  CO2 لتحميل واستخدام البيانات وهي في مثالنا
> head(CO2)    #  لتحميل البيانات لست صفوف في المقدمة
  Plant   Type  Treatment conc uptake
1   Qn1 Quebec nonchilled   95   16.0
2   Qn1 Quebec nonchilled  175   30.4
3   Qn1 Quebec nonchilled  250   34.8
4   Qn1 Quebec nonchilled  350   37.2
5   Qn1 Quebec nonchilled  500   35.3
6   Qn1 Quebec nonchilled  675   39.2
>tail(CO2)    # لتحميل البيانات لست صفوف في النهاية
```

(17) الرسم البياني في R

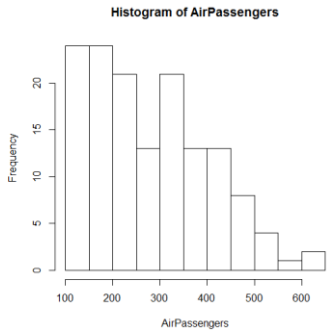
هو تمثيل مرئي لتوزيع البيانات. حيث ان شكل الرسم البياني هو السمة المفيدة والأكثر وضوحا فهو يسمح لك أن ترى بسهولة حيث يقع على كمية كبيرة نسبيا من البيانات وحيث يوجد القليل جدا من البيانات التي يمكن العثور عليها. وبعبارة أخرى، يمكنك ان ترى فيها منتصف توزيع البيانات ومدى قرب هذه البيانات حول هذا الوسط، حيث يمكن العثور على القيم المتطرفة الممكنة. وبسبب كل هذا، فالرسوم البيانية هي طريقة رائعة للتعرف على البيانات . وباختصار، فإن الرسم البياني يتكون من المحور السيني x، والمحور الصادي y ونقاط متعددة من ارتفاعات مختلفة. ويوضح المحور y عدد المرات التي تحدث القيم على المحور x في البيانات.

لغة البرمجة R

1// الدالة hist() تمثل الرسم البياني للبيانات الفئوية

مثال : hist(AirPassengers)

والذي يعطي الرسم البياني :



كما ويمكن اضافة المعلمات التالية لتحسين صورة المخطط حيث:

main تعطي العنوان

xlab و ylab لتوفير تسميات المحاور

xlim وylim لتوفير مجموعة من المحاور

col لتحديد اللون

breaks لتحديد عدد الفواصل

hist(AirPassengers, main"=Histogram for Air Passengers ،"xlab"=Passengers ،" border"=blue ،"col"=green ،"xlim=c ،(100,700)las ،1=breaks=5)

مثال :

```
> x <- c(12, 15, 13, 20, 14, 16, 10, 10, 8, 15)
> hist(x)
```

>hist(rnorm(100))

>hist(rnorm(100),breaks=50)

2// الدالة plot(…)

وهي دالة التخطيط الاكثر استخداماً في R ،وفي ابسط الحالات يمكن رسم المتجهات بشكل مبعثر ، وعلى سبيل المثال :

التعبير : >plot(c(1,2),c(3,5))

يرسم النقطتين : (1,3) و(2,5)

```
> x <- seq(1, 10)
> y <- x**2 - 10 * x
> plot(x, y)
```

- تغيير لون ونوع الرسم

يكون الرسم عبارة عن نقاط دائرية وسوداء اللون. هذا هو الافتراضي. ويمكن تغيير نوع الرسم ولونه وفيما يلي كل الرموز الممكن استخدامها والشكل الذي يرسمه:

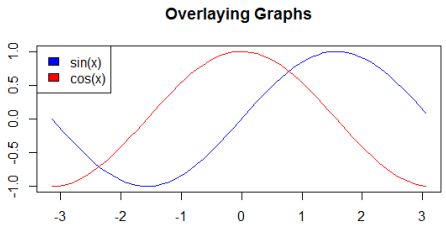
P	نقاط	o	دوائر صغيرة على نقاط الرسم وخطوط تصل بينها
I	خطوط	S و s	خطوط درج
b	نقاط وخطوط معاً	h	الخطوط العمودية مثل الرسم البياني
c	نقاط فارغة تربط بخطوط	n	لا ترسم اي نقاط او خطوط

وبالمثل يمكن ان نحدد اللون باستخدام col

>plot(x, sin(x),main="The Sine Function",ylab="sin(x)", type="l",col="blue")

- الدالة legend() تضيف وسيلة ايضاح الى الرسم ، حيث يحدد فيها موقع وسيلة الايضاح والتسمية لخطوط الرسم واللون لكل رسم، والمثال التالي يوضح ذلك (انظر الشكل التالي :

legend("topleft",c("sin(x)","cos(x)"),fill=c("blue","red"))



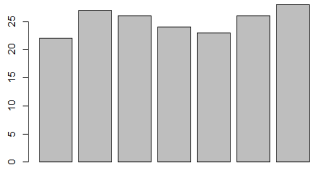
3// الدالة barplot()

تستخدم في R للرسم الشريطي ، نفترض، لدينا متجه من درجات الحرارة القصوى (في درجة مئوية) لمدة سبعة أيام على النحو التالي:

>max.temp <- c(22, 27, 26, 24, 23, 26, 28)

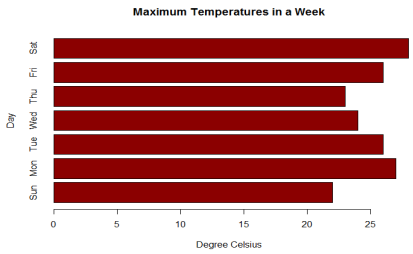
يمكن رسم شريطي من هذه البيانات :

>barplot(max.temp)

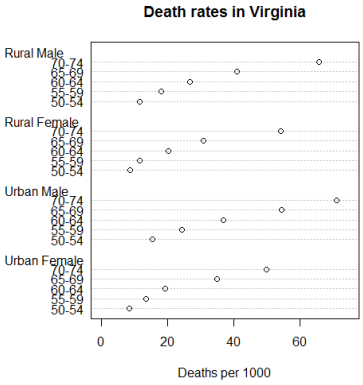


ويمكن اضافة معلمات رسومية الى تلك الدالة في اعطاء
عنوان رئيسي وتسميات للمحاور وتسمية كل شريط بيانات في الرسم وتحديد اللون وحتى يمكن الرسم الافقي من خلال المعلمة horiz=TRUE

>barplot(max.temp,main="Maximum Temperatures in a Week",Celsius",ylab="Day", names.arg =c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat"), col="darkred",horiz=TRUE)



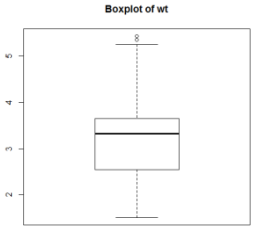
4// الدالة dotchart



> dotchart(VADeaths, xlim=c(0, 75),xlab="Deaths per 1000",
+ main="Death rates in Virginia")

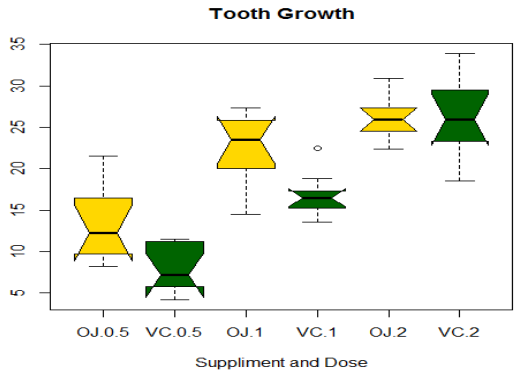
5// الدالة Box plots

>attach(mtcars)
> boxplot(wt,main="Boxplot of wt")



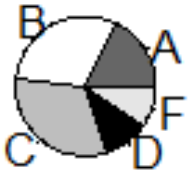
المربعات الملونة لسهولة التفسير

>boxplot(len~supp*dose,data=ToothGrowth, notch=TRUE, col=(c("gold","darkgreen")), main="Tooth Growth", xlab= "Suppliment and hgfdhkhjDose")



6// الدالة Pie

تستخدم للتخطيطات الدائرية تعرض متجه من الأرقام بتقطيع قرص دائري إلى قطع حيث الزاوية يتناسب مع كل عدد. على سبيل المثال، الدرجات إلكتروني المخصصة لفئة قد تنشأ في النسب، الذي يُرسم بإيعازات R التالية :



> groupsizes <- c(18, 30, 32, 10, 10)
> labels <- c("A", "B", "C", "D", "F")
> pie(groupsizes, labels, col=c("grey40", "white", "grey", "black", "grey90"))

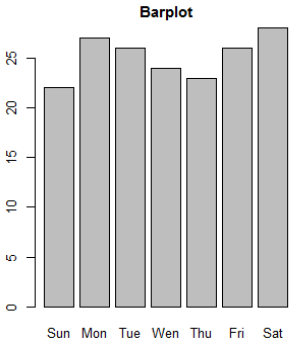
7// دالة par()

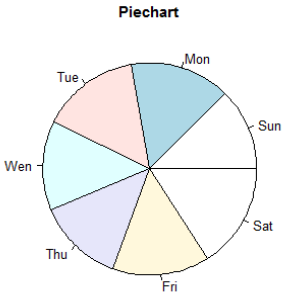
تعرض لائحة طويلة من المعلمات ولمعرفة عمل كل واحدة يمكن مراجعة التعليمات. اضافة الى تقسيم صفحة النتائج وان المعلمة الرسومية mfrow للرسم في الصفوف اولا و mfcol للرسم في الاعمدة اولاً ، وان c(m,n) تحدد العدد المطلوب :

par(mfrow=c(2,2)) # تعين منطقة الرسم الى 2*2

```
> max.temp=c(22,27,26,24,23,26,28)
> names(max.temp)=c("Sun","Mun","Tue","Wen","Thu","Fri","Sat")
> max.temp
Sun Mun Tue Wen Thu Fri Sat
 22  27  26  24  23  26  28
```

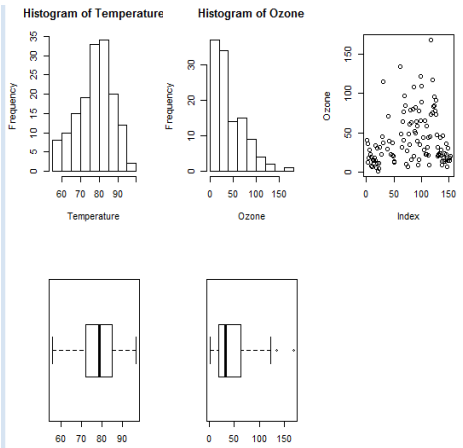
barplot(max.temp, main="Barplot")
pie(max.temp, main="Piechart", radius=1)

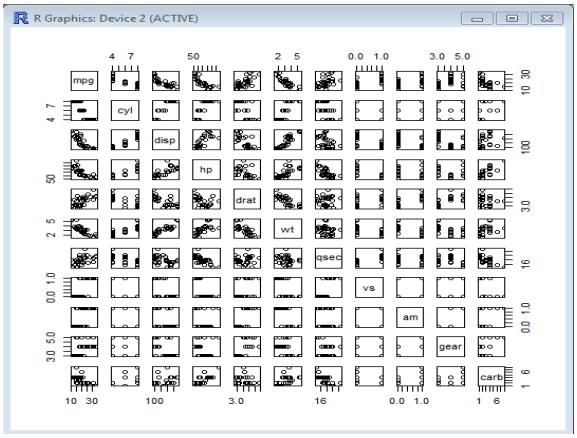




فيما يلي ايعازات برسم 5 رسوم مختلفة في صفين وثلاث اعمدة :

>Temperature <- airquality$Temp
>Ozone <- airquality$Ozone
>par(mfcol=c(2,3))
>hist(Temperature)
>boxplot(Temperature, horizontal=TRUE)
>hist(Ozone)
>boxplot(Ozone, horizontal=TRUE)
>plot(Ozone)





8// الرسوم المزدوجة pairs()

استخدمنا اطار البيانات "mtcars" مع 32 حالة على 11 متغير، لرسم مجموعة من الخرائط لكل زوج ممكن من هذه البيانات في شكل رسم بياني مبعثر بحيث كل زوج في مخطط متبادل الموقع حيث يظهر التمثيل في محاور x وy، يوضح الشكل التالي مثالا لناتج تنفيذ هذا القانون عند تطبيقه على اطار البيانات mtcars

>pairs(mtcars)

9// رسم ثلاثي الابعاد 3D plot

يتمثل في ثلاث متغيرات وهي x، y، z ، حيث x وy متجهان يحددان على المستوى اما الارتفاع فهو المتجه z والذي يتكون من مصفوفة للبعدين x و y.

Contour Plots (1

>x <- 1:50 ; y <- 1:70

```
>z <- matrix(expand.grid(x,y)$Var1^2+expand.grid(x,y) $Var2^2,50,70)
>contour(x,y,z)
```

image plots (2

```
>x <- y <- seq(-4*pi, 4*pi, len = 27)
>r <- sqrt(outer(x^2, y^2, "+")) ; z <- cos(r^2)*exp(-r/6)
>image(z = z <- cos(r^2)*exp(-r/6), col  = gray((0:32)/32))

>image(t(volcano)[ncol(volcano):1,]) رسم لبيانات بركان #
 >m = matrix(runif(100),10,10); image(m)                مثال اخر:
```

persp() (3

هذه الدالة ترسم قطع السطوح فوق مستوى x-y وكما في المثال التالي :

```
>x <- seq(-10, 10, length= 30) :y <- x
>f <- function(x,y) { r <- sqrt(x^2+y ^2);  10 * sin(r)/r }
>z <- outer(x, y, f)
>z[is.na(z)] <- 1
>persp (x, y, z, aspect=c(1, 1, 0.5), col = "lightblue",xlab = "X", ylab = "Y", zlab = "Sinc( r
)")
```

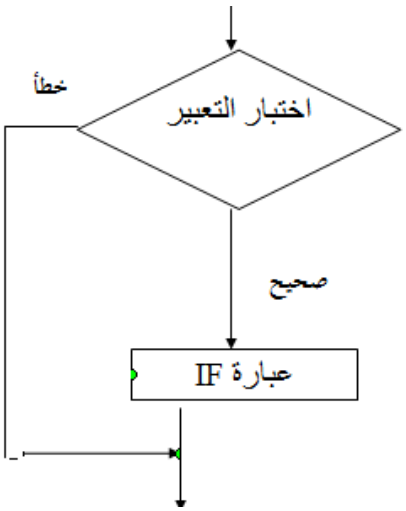
(18) البرمجة في R

اهم نقاط الجذب الرئيسية في استخدام بيئة البرمجة R هو السهولة التي يمكن للمستخدمين كتابة البرامج والمهام الخاصة بهم ، والوضوح في مراقبة هيكل البرنامج .

لغة البرمجة R مثل اي لغة برمجة اخرى تمتلك عبارات برمجية خاصة تُستخدم لحل مشكلة ما وتُحاكي الحاسب لتحصل على النتائج وفيما يلي عبارات البرمجة المستعملة في لغة R وبناء الجملة والمخطط لكل عبارة

if statement if عبارة //1

```
if (test_expression) {
  statement
}
```



إذا كان test_expression هو صحيح، يحصل تنفيذ العبارة. ولكن إذا كان خطأ، لا يحدث أي شيء.

- مثال :

```
> if(1==0) {
+     print(1)
+  } else {
+     print(2)
+  }
[1] 2
```

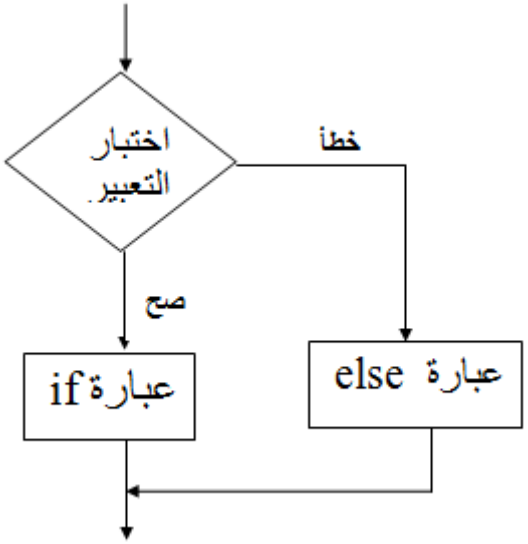
- مثال :

```
x <- 5
if(x > 0){
  print("Positive number")
}
 [1] "Positive number"
```

if…else عبارة //2

```
if (test_expression) {
  statement1
} else {
  statement2
}
```

يتم التقييم اذا (test_expression) هو صحيح، ينفذ ما بعده والموجود بين الاقواس {}، ومن المهم ان نلاحظ ان else يجب ان يكون في نفس خط القوس المغلق لعبارة if والذي يُنفذ ما بعده عندما يكون التعبير خطأ.




```
x <- -5
if(x > 0){
  print("Non-negative number")
} else {
  print("Negative number")
}
 [1] "Negative number"
```

يمكن ان يُكتب الشرط اعلاه في سطر واحد وكما يلي:

```
if(x > 0) print("Non-negative number") else print("Negative number")
```

- مثال

```
> even.odd = function(x) {
+ if (!is.numeric(x)) {
+ print("neither") }
+ else if (x%%2 == 0) {
+ print("زوجي") }
+ else {
+ print("فردي")
+ } }
> even.odd(3)
[1] "odd"
> even.odd(4)
[1] "even"
> even.odd("A")
[1] "neither"
```

Or in other way:

```
> a = c(5,7,2,9)
> ifelse(a %% 2 == 0,"even","odd")
[1] "odd"  "odd"  "even" "odd"
```

3// عبارة if…else المتداخلة

```
if ( test_expression1) {
  statement1
} else if ( test_expression2) {
  statement2
} else if ( test_expression3) {
  statement3
} else
  statement4
```

فقط جملة واحدة سوف تُنفذ اعتماداً على test_expressions.

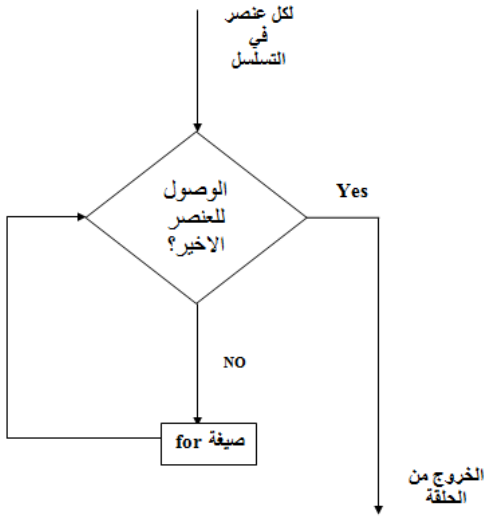
```
x <- 0
if (x < 0) {
  print("Negative number")
} else if (x > 0) {
  print("Positive number")
} else
  print("Zero")
 [1] "Zero"
```

4// عبارة for

الحلقة تُستخدم لتكرار المتجه في برمجة R

```
for (variable in sequence) {
   statements
}
```

هنا، sequence هو متجه وvariable يأخذ في كل مرة قيمته خلال الحلقة.



- مثال : فيما يلي مثال لحساب عدد الارقام الزوجية في المتجه التالي:

```
x <- c(2,5,3,9,8,11,6)
count <- 0
for (val in x) {
   if(val %% 2 == 0)  count = count+1
}
print(count)
 [1] 3
```

في المثال اعلاه لدينا 7 ارقام في التجه x ، اخذنا المتغير count ليكون عداد لحساب عدد الارقام الزوجية باختبار باقي القسمة على 2 اذا كان الناتج صفر فالعدد هو رقم زوجي ،وبذلك اصبح الناتج 3.

- مثال

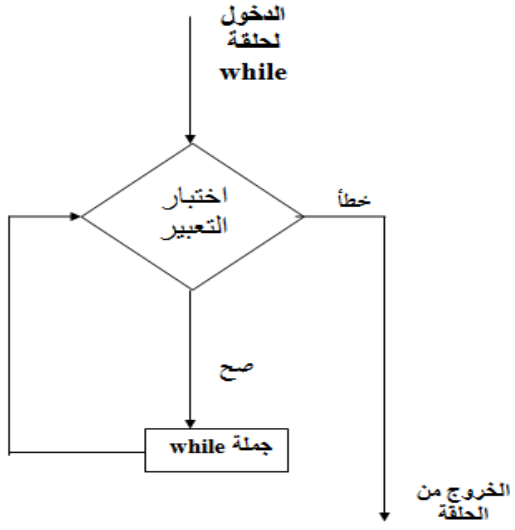
```
> for (x in 1:3) {
(print(x +
+ }
[1] 1
[1] 2
[1] 3
```

5//عبارة while

وهي على غرار حلقة for ، لكن يتم التحكم في التكرار من قبل عبارة شرطية :

```
while (test_expression) {
  statement
}
```

هنا، يتم تقييم test_expression فاذا كانت النتيجة صحيحة يتم تنفيذ التعليمات داخل الحلقة ويستمر تقييم التعبير حتى تصبح النتيجة خاطئة ، في هذه الحالة، يتم إنهاء الحلقة.

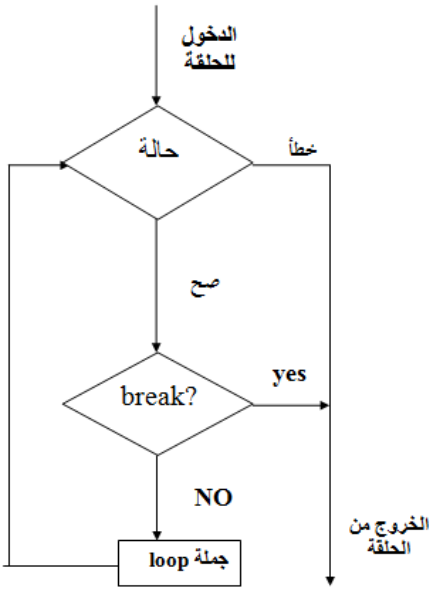



```
i <- 1
while (i < 6) {
  print(i)
  i = i+1
}
 [1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

في المثال أعلاه، يتم تهيئة i إلى 1. وهنا test_expression هو I 6> الذي يقيم صحيح كلما 1 هو أقل من 6. لذا، تم تنفيذ داخل الحلقة وطباعتها وتستمر الحلقة حتى i يأخذ قيمة 6. فيصبح الشرط 6 <6 وهذا يعطي النتيجة خطأ ويتم الخروج من الحلقة .

6// عبارة break

يستخدم عبارة break داخل حلقة لوقف تكرار وتدفق حالة الحلقة.




```
x <- 1:5
for (val in x) {
   if (val == 3){
      break
   }
   print(val)
}
 [1] 1
[1] 2
```

في المثال اعلاه التكرار لاعداد المتجه x من 1 الى 5،داخل الحلقة يوجد شرط لكسر الحلقة وهو القيمة تساوي 3 .

7// الدوال Functions

تقوم بوظيفة معينة او محددة ومن فوائد الدوال functions هو تصغير الكود البرمجي والتخلص من تكرار بعض المهام التي تتكرر باستمرار، وكذلك توجد إمكانية إنشاء دوال تؤدي وظيفة خاصة

- بناء الجملة function

```
Function_name<-functions(arg1,arg2,….){
      Function –body
}
```

القيمة التي تم ارجاعها بواسطة function هي قيمة Function –body والذي عادة ما يكون تعبير نهائي غير معين ، مثلاً : return()

```
 (1)> square = function(x) { return(x^2)}
> square(5)
[1] 25
> square(1:5)
[1] 1 4 9 16 25

 (2)>a=1:10
>a^2
>sum(a^2)
>Fct<-function(x){sum(x^2)}
>Fct(1:5)

(3)> cube = function(x) x^3
> cube(2)
[1] 8
> cube(1:5)
[1] 1 8 27 64 125

* (two parameter)
(4)>fct<-function(x,y){
>res<-x^2+y
>res}
>fct(2,1)

(5)> pow = function(x, y = 2) x^y
> pow(2)
[1] 4
> pow(2, 4)
[1] 16
> pow(y= 4,2)
[1] 16
> pow(y =3, x = 3)
 [1] 27

(6) f.bad <- function(x, y) {
 z1 <- 2*x + y
 z2 <- x + 2*y
 z3 <- 2*x + 2*y
 z4 <- x/y}
f.bad(1, 2)
[1] 0.5
```

لغة البرمجة R

RStudio (19)

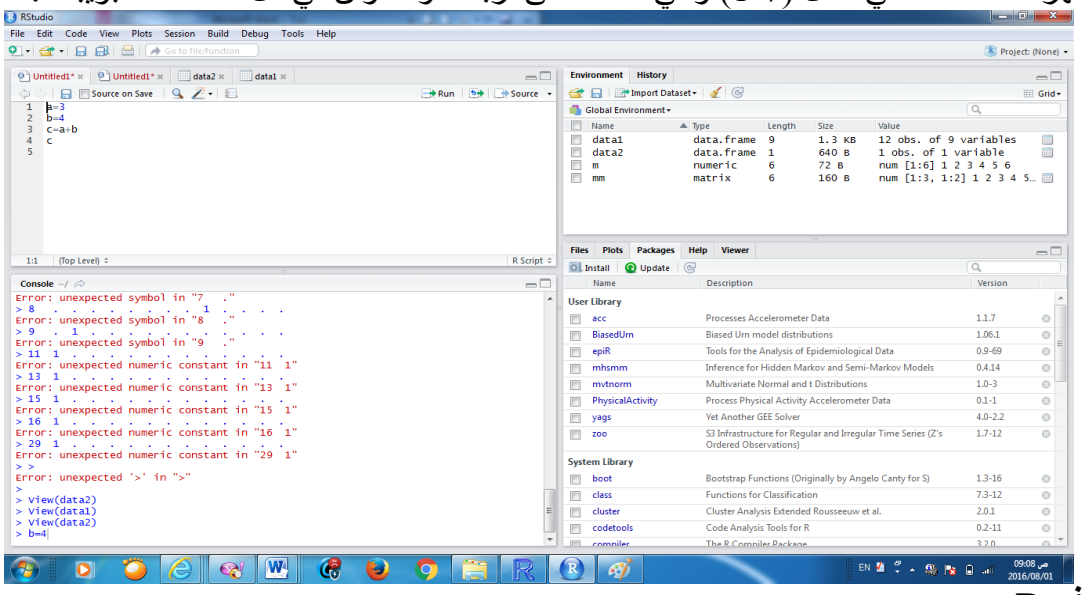
يعتبر RStudio واجهة بديلة لـ R حيث يمكن تثبيت RStudio كتطبيق سطح المكتب أو كتطبيق الخادم الذي هو في متناول الآخرين عبر الإنترنت. فهو مفتوح المصدر (اي مجاناً) ومتاح من الموقع http://www.rstudio.org .

RStudio يقدم واجهة احسن لبرنامج R فهو بيئة التطوير المتكاملة (IDE / Integrated development environmen) وتطبيق البرمجيات التي توفر مرافق شاملة لمبرمجي الكمبيوتر لتطوير البرمجيات. وIDE يتكون عادة من محرر شفرة المصدر، وبناء أدوات التشغيل الآلي ومصحح.

يمكنك استخدام R دون RStudio باستخدام أي محرر لكتابة البرامج النصية. ولكن RStudio في حد ذاته ليس من المفيد جدا دون R .وهو البرنامج المكتوب في لغة البرمجة C++ .

RStudio هو الواجهة لـR. والذي يجعل استخدام R أجمل كثيرا كما يضيف بعض المكونات السهلة الاستخدام حيث يتيح لك الكثير من الحزم بسهولة .كما يوفر الوصول الى R في متصفح الويب ، اضافة الى هذا فان RStudio يضم العديد من النوافذ المفتوحة في نفس الوقت وادوات لتنظيم العمل في المشاريع وغيرها.

وأن R و RStudio كلاهما برامج بُنيت للحوسبة الاحصائية ،ولها مصدر مفتوح ومجاني وفي حالة تشغيل برنامج RStudio تظهر الشاشة كما في شكل (1-7) وهي مقسمة الى اربعة نوافذ وان في كل نافذة عدة تبويبات:



الحزم في R (20)

الحزمة باقة من التعليمات البرمجية والبيانات والوثائق، و الاختبارات، اي ملفات اضافية جاهزة لحل مشاكل معقدة في R سهلة للمشاركة مع الآخرين. فقد اضاف الاف الخبراء في مجال العلوم الاحصائية وظائف على R في شكل حزم والتي هي ايضاً متاحة بحرية على صفحات الانترنت ، واعتبارا من بداية عام 2015، كان هناك أكثر من 6000 حزمة متوفرة على شبكة الارشيف الشامل R أو كرا CRAN، وان توفر مجموعة كبيرة من الحزم من بيانات وتطبيقات تمتد عبر العديد من التخصصات المختلفة في الدراسة هو احد الأسباب التي جعلت لغة R ناجحة للغاية ، حيث ان هناك احتمالات بأن شخصا ما قد حل بالفعل المشكلة التي نعمل عليها، ويمكنك الاستفادة من عملهم من خلال تحميل تلك الحزمة .

قبل ان تتمكن من استخدام الحزمة ، عليك اولاً تثبيتها ، فبعض الحزم هي اساسية ويتم تثبيتها تلقائياً عند تثبيت حزم R ، والحزم الاخرى لن تأتي مع تركيب R لذلك فهي تحتاج الى تثبيت .

توجد ثلاث دوال مهمة في استخدام الحزم (ليكن اسمها مثلاً : x) وهي :

(1) تحميل الحزمة من داخل R: >install.packages("x")

(2) استخدام الحزمة في R : >library("x")

(3) للحصول على المساعدة يكون اما بـ: >?x او help(x)

1// تحميل حزم البيانات

قد تكون الحزم المتوفرة في R هي بيانات مخزونة في R ولاستدعاءها نكتب الايعاز : >data()

فتظهر مجموعات البيانات في حزمة 'datasets' ضمن نافذة مجموعات بيانات R ولادراج وعرض اي من حزم البيانات المتوفرة ، نكتب في سطر الاوامر اسم الحزمة .

مثال : لتكن حزمة البيانات BOD ، نكتب : >BOD

2// تحميل حزم البرامج

يتضمن برنامج R برامج اضافية لها امكانية اوسع في الحل ، وللحصول على قائمة بجميع الحزم المثبتة في R نكتب الامر : library()< فتظهر نافذة قد تختلف تبعا للإعدادات المحلية في جهاز الكمبيوتر الخاص بك.

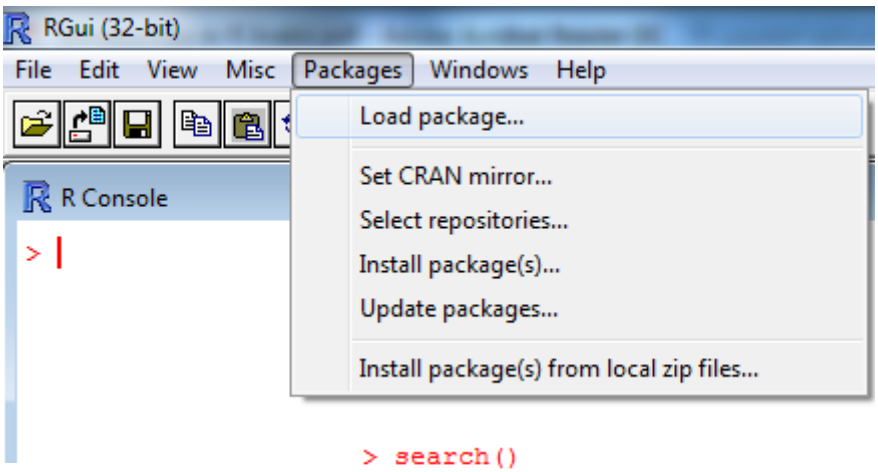
ولتحميل اي حزمة من الحزم المدرجة في النافذة ولتكن " boot" نستخدم الايعاز : library(boot)<

وللحصول على معلومات حول الحزمة المطلوبة وهي في مثالنا "boot" ، نكتب الايعاز : library(help=boot)<

وللحصول على بيانات تلك الحزمة نستخدم ايعاز البيانات : data()< فتظهر البيانات الخاصة بحزمة "boot" وكما في شكل(8-4) :

(21) قائمة Packages

يمكن ادراج الحزم المتوفرة على حاسبتك في لغة البرمجة R وذلك من الشاشة الافتتاحية لبرنامج R ومن شريط القوائم نفتح قائمة Packages كما في الشكل المجاور



حيث تحتوي على 6 اوامر .

1// Load package…

2// Set CRAN mirror…

3// Select repositories…

4//Install package(s)…

5//Update packages…

- الدالة search() للحصول على كافة الحزم التي تم تحميلها حاليا في بيئة R:

```
> search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
```

(22) مستودعات لحزم اضافية

وفيما يلي اسماء مخازن تحتوي على حزم اضافية ومجموعة ابحاث مفتوحة المصدر مطورة من قبل باحثين .

1// CRAN (Comprehensive R Archive Network)

وهو موقع لحزم تخدم في عدة مجالات مثل المعلومات الحيوية ويسرد مجموعة واسعة من المهام في العلوم الاجتماعية والاحصاء وعلم الوراثة وارتفاع اداء الحاسبات والة التعلم والتصوير الطبي كما وتُستخدم من قبل ادارة الاغذية والعقاقير باعتبارها مناسبة لتفسير البيانات في البحوث السريرية ولتحميل حزمة من موقع كرا

من قائمة packages نختار install packages… --- يظهر مربع حوار cran mirror لاختيار حزمة واحدة ثم يظهر مربع حوار packages ويمكن اختيار اكثر من حزمة. > install packages ("Packages Name")

2// Bioconductor

هو موقع مفتوح المصدر يوفر حزم R لتحليل بيانات الجينوم عالية الانتاجية الا انه يقبل الحزم التي تخضع لشروط الموقع من اضافة لتوضيح الحزمة والدوال مع امثلة وبترتيب خاص .. لهذا نجد اغلب الباحثين يتعاملوا مع الموقع CRAN لسرعة نشر برامجهم . يتم تحميل الحزم بخطوتين :

- اولاً : تحميل موقع bioconductor :

> source("http:// bioconductor.org/biocLite.R")

- ثانياً : تحميل الحزم باستعمال الدالة > biocLite("Packages Name")