كلية العلوم

الفصل الثاني
٢٠١٩-٢٠٢٠

قسم الإحصاء الرياضي-السنة الرابعة

برامج إحصائية (٢)
عملي

المحاضرة (١+٢+٣)

أ. محمد العوض
أ. بشرى السعيد

– لغة البرمجة الإحصائية R –

لغة البرمجة آر هي بيئة برمجية إحصائية للتنهيج والرسم، اخترعها روس ايهاكا وروبيرت جنتلمان وسميت بلغة R تيمناً بإسميهما.

لغة R هي بيئة عمل برمجية للإحصاء الحاسوبي تسمح بإجراء التطبيقات الإحصائية من جهة وبناء البرامج الإحصائية من جهة أخرى. هي مفتوحة المصدر Open Source، وهناك نسخة مجانية لها من لغة البرمجة الإحصائية +S (Splus) والتي تعتمد على اللغة آس (S). يمكن إدخال المعطيات إلى لغة R يمكن أن تكون أشعة أو مصفوفات أو سلاسل زمنية وقد تكون توابع أو رسوم بيانية. إنّ لغة R هي لغة تابعية يعني كل أمر من أوامرها هو تابع يقبل العديد من الوسطاء ومن مزاياها أنها تمتلك واجهة لغة HTML وتدعم كافة أنظمة التشغيل (Windows, Linux,...) مما يساعد على تنفيذ الأوامر البرمجية بسهولة.

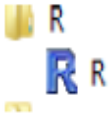
تم تصنيف لغة R ضمن المرتبة التاسعة للغات الأكثر شعبية حسب معيار IEEE، كما أنها تصنف اللغة الأكثر شعبية في مجال علوم البيانات. بالإضافة إلى استخدامها من قبل آلاف الشركات المهتمة بعلوم البيانات.

وهذه بعض الأمثلة القليلة عن استخدام R:

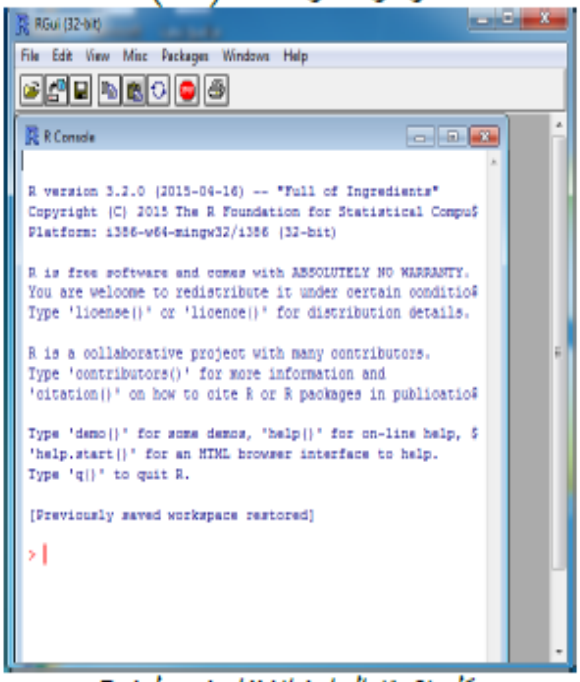
- تستخدم جوجل مشروع R لحساب عائدات الاستثمار في الحملات الإعلانية.
- تستخدم فورد R لتحسين تصميم سياراتها.
- يستخدم تويتر R لمراقبة تجارب المستخدمين.
- تستخدم المنظمة الوطنية للطقس المشروع R من أجل التنبؤ بالفيضانات الشديدة.
- يستخدم معهد روكفلر الحكومي R من أجل تطوير نماذج لمحاكاة أموال صناديق التعاقد العامة.
- تستخدم مجموعة حقوق الإنسان لتحليل البيانات مشروع R من أجل قياس تأثيرات الحروب.
- يُستخدم مشروع R كثيراً من قِبل صحيفة نيويورك تايمز من أجل إنشاء الرسوم البيانية والتطبيقات الصحفية ذات البيانات التفاعلية.

6.2 تشغيل برنامج R

1) النقر المزدوج على أيقونة R على سطح المكتب

2) او من start ⬅ All Programs ⬅ 

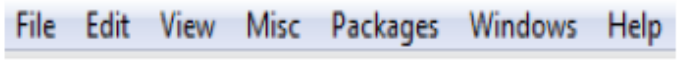
وحينها ستظهر الشاشة الافتتاحية بسطر الاوامر الخاصة بلغة R حيث تظهر علامة التنصيص (>) للبدأ بكتابة الاوامر.انظر الشكل (2-1) :



شكل (2-1) الشاشة الافتتاحية في لغة R

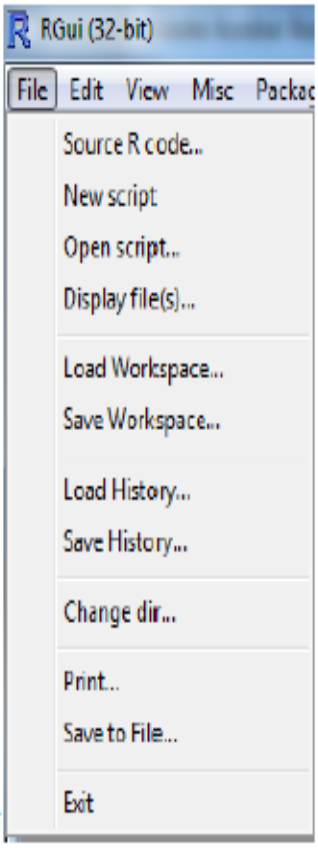
7.2 واجهة برنامج R

1.7.2 شريط القوائم



يظهر فيه اسماء القوائم للبرنامج وفيما يلي الاوامر في كل قائمة :

(1) File ويحتوي الاوامر التالية كما في الشكل (2-2):

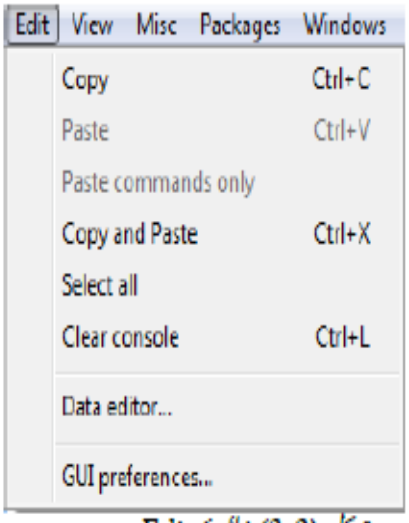


شكل (2-2) قائمة File

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

- من قائمة File نختار New script
- او نضرب المفاتيح ctrl+shift+N معاً

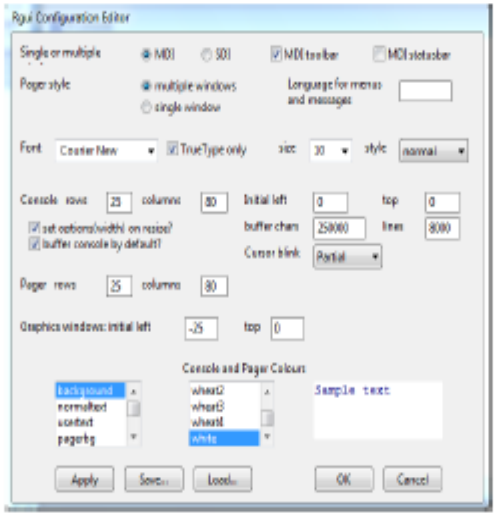
تظهر شاشة " Untitled - R Editor " ، ثم يمكن نقل تلك العبارات الى فضاء العمل بايعازي النسخ واللصق او ctrl+r .



شكل (2-3) قائمة Edit

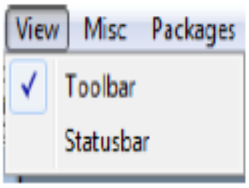
(2) Edit : ويحتوي الاوامر التالية :

GUI preferences... : لتغيير مواصفات شاشة العرض في R، من القائمة Edit في الشكل (2-3) نختار GUI preferences... يظهر مربع حوار " Rgui Configuration Editor" وكما في الشكل (2-4) :



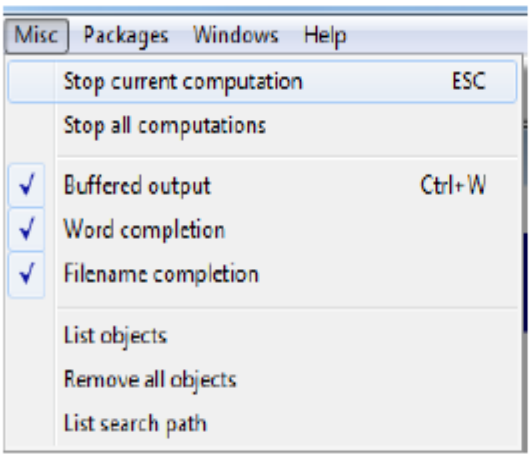
GUI preferences... مربع حوار (4-2) شكل

(3) View : ومحتوياتها كما في الشكل (5-2)



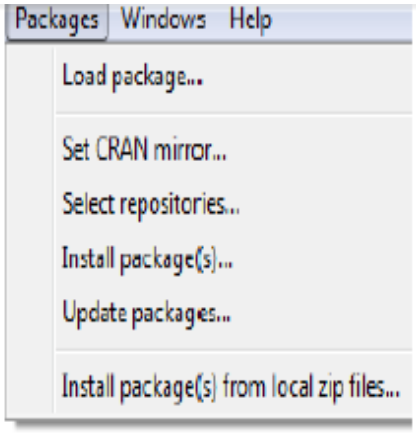
View قائمة (5-2) شكل

(4) Misc : وتحتوي على عدة اوامر كما في الشكل (6-2)



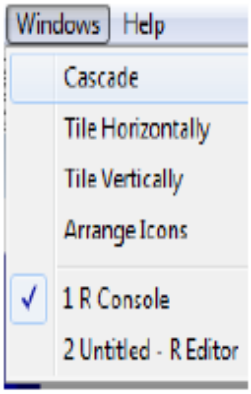
Misc قائمة (6-2)شكل

(5) Packages : وتحتوي على اوامر لاستدعاء الحزم وكما في الشكل (7-2)



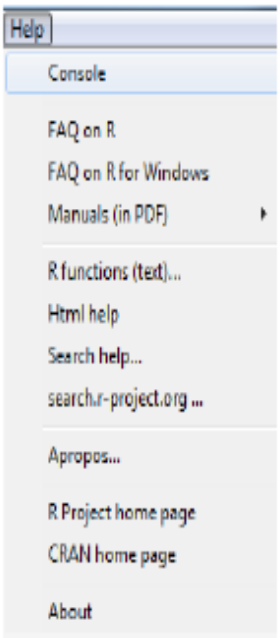
شكل(2-7) قائمة *Packages*

(6) Windows : لاظهار النوافذ وترتيبها في اوامر كما في الشكل (2-8)



شكل (2-8) قائمة *Windows*

(7) Help : للحصول على التعليمات لبرنامج R وضمن اوامر عدة كما في شكل (2-9)



شكل (2-9) قائمة *Help*

وفي بند لاحق سنوضح الحصول على المساعدة لاوامر او ايعازات في برنامج R

2.7.2 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 نسخ

(5) Paste لصق

(6) Copy and paste نسخ ولصق

(7) Stop current computation وقف الحساب الحالي

(8) Print طباعة

3.7.2 شريط الحالة statusbar

ويظهر اسفل الشاشة ويُكتب فيه نوع الاصدار لـ R والسنة

الاوامر الخاصة في لغة البرمجة R:

لكل لغة من لغات البرمجة عدد من الرموز الخاصة التي تميزها عن اللغات الاخرى، وتعتبر قواعد هامة يجب الالتزام باستخدامها الصحيح وفيما يلي أهم وأكثر الأوامر الخاصة في لغة البرمجة R:

1) الرمز (<-): يشـــار الى عملية الإســـناد في لغة R، وهي الطريقة الأكثر شـــيوعاً مقارنة برمز المســـاواة = والذي يصح استخدامه أيضاً. (سنرى الفرق لاحقاً)

مثال:

```
x<-14
x=14
14->x
```

2) الفاصلة المنقوطة (;): تستخدم لفصل الأوامر في السطر الواحد، أو يكتب الأمر التالي في سطر جديد.
3) العبارة التعريفية: كل شيء بعد الرمز # (هاش) إلى نهاية السطر يعتبر عبارة تعريفية أو تعليق للمستخدم ويتم تجاهله من قبل لغة R.
4) يمكن ان يكون الأمر أو العبارة في لغة R على عدة أسطر.
5) تُطبع قبل النتيجة العلامة "[1]" وهذا لأن R يعتبر افتراضياً كل شيء بمثابة جدول "vector " والرقم واحد هو مؤشر عن العنصر الأول في الجدول، مثال:

```
> 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
```

6) إن ناتج تنفيذ أي أمر أو دالة بلغة R يظهر بعدها مباشرة (التنفيذ لحظي).
7) يوفر R آلية لإعادة تنفيذ الأوامر السابقة وذلك باستخدام مفاتيح الأسهم على لوحة المفاتيح حيث يمكن أن تستخدم للتنقل إلى الأمام والخلف من خلال الأوامر السابقة الاستخدام، كما يمكن للمؤشر أن ينتقل ضمن الأوامر باستخدام مفاتيح الأسهم الأفقية، والأحرف يمكن ازالتها مع مفتاح DEL أو اضافتها من مفاتيح اخرى.
يمكنك باستخدام أزرار الأسهم إلى الأعلى وإلى الأسفل للتنقل عبر مجموعة الأوامر التي تم تنفيذها.
8) R حساسة لحالة الاحرف فمثلاً a و A نوعان من كائنات مختلفة.
9) تعطي R العلامة + في بداية السطر للدلالة على أن المدخل حتى الآن لم يكتمل تعريفه.
10) يفضل عدم بدأ اسم المتغير بخط سفلي (_) أو رقم لأنه قد يؤدي إلى رسالة خطأ، وإذا بدأ بنقطة فيفضل ألا يتبع النقطة رقم.
11) لأنهاء برنامج R نطبع الامر:

```
>q ()
```

يظهر مربع حوار يطلب منك ما إذا كنت تريد حفظ البيانات من جلسة R الخاصة بك. أما في حالة كتابة q بدون اقواس يظهر خطأ وكما يلي:

```
> q
function (save = "default", status = 0, runLast = TRUE)
.Internal(quit(save, status, runLast))
<bytecode: 0x06960ebc>
<environment: namespace:base>
```

12) للاستفسار عن ايعاز ما او أي دالة اسمها محدد نستخدم الامر:

```
>help (…)
```

حيث نضع بين القوسين الايعاز او الكلمة التي نريد الاستفسار عنها، مثلاً:

```
>help(solve) أو help(mean)
```

أو نكتب علامة الاستفهام ونذكر بعدها الايعاز أو الكلمة التي نريد الاستفسار عنها

```
>?solve أو ?mean
```

وفيما يلي بعض الاوامر القياسية لإدارة العمل الخاص بك:

```
> getwd()
```
لمعرفة دليل العمل الحالي

```
> ls()
```
لمعرفة الكائنات الموجودة في ساحة العمل

```
> setwd(my_directory)
```
لتغيير دليل العمل

العمليات الحسابية:

العملية	الوصف
+	الجمع
-	الطرح
*	الضرب
/	القسمة
^ أو **	القوة
%%	باقي القسمة
%/%	القسمة الصحيحة

العمليات المنطقية:

العملية	الوصف
==	المساواة
!=	عدم المساواة
>	أكبر
>=	أكبر أو يساوي
<	أصغر
<=	أصغر أو يساوي
&	و
|	أو

```
> 5 + 3
[1] 8
> 15.3 * 23.4
[1] 358
>2+7.2-5*3/2+2^4
[1] 17.7
> 31%%7
[1] 3
> 16%/%5
[1] 3
```

والجدير بالذكر هنا أنه كأي لغة برمجة أخرى يوجد عدة أنواع من المتغيرات ومنها الصحيح integer والرقمي numeric والمركب complex والمحرف character والمنطقي logical ...إلخ

لمعرفة نوع متغير ما وليكن x يمكن أن نستخدم إحدى التوابع:

```
typeof(x)
class(x)
mode(x)
```

ويجب معرفة أن المتغير من النوع المنطقي يأخذ إحدى القيمتين TRUE(T) و FALSE(F) (بأحرف كبيرة).

يمكن التحكم في عدد الخانات لرقم عشري ما عن طريق التابع:

```
> options(digits=الرقم المطلوب)
```

مثلاً:

```
> 10/3
[1] 3.333333
> options(digits=16)
> 10/3
[1] 3.333333333333333
```

بعض التوابع الشهيرة:

العملية	الوصف
`sqrt()`	الجذر التربيعي
`exp()`	التابع الأسي
`log()`	التابع اللوغاريتمي

أمثلة:

```
> sqrt(-1)
[1] NaN   (أي ليس موجود)
> sqrt(-1+0i)
[1] 0+1i
> sqrt(as.complex(-1))
[1] 0+1i
> (0+1i)^2
[1] -1+0i
```

الأشعة (Vector):

الشعاع هو عبارة عن متحول يأخذ عدّة قيم من نوع واحد. ولتعريف الشعاع بلغة R يتم استخدام التابع c (حرف c صغير)، فمثلاً لوضع القيم 2,3,4,5 في شعاع x نكتب:

```
> x<-c(2,3,4,5)
```

حيث يرمز الحرف c إلى الكلمة concatenate أي "تسلسل".

ولمعرفة عدد عناصر الشعاع نستخدم التابع length بالشكل:

```
> length(x)
```

ففي المثال السابق سيظهر لدينا:

```
[1] 4
```

يمكن استخدام التوابع seq و rep لنفس الغرض ولكن بشكل آخر **فللتابع seq** الشكل العام التالي:

```
> seq(from,to,by)
```

ويقوم هذا التابع بتوليد متتالية من الأرقام من from إلى to وبخطوة قدرها by فمثلاً لو كتبنا:

```
> seq(from=2,to=30,by=3)
[1] 2 5 8 11 14 17 20 23 26 29
```

ويمكن كتابتها بالشكل:

```
> seq(2,30,3)
```

إذا كنا نريد توليد متتالية من from إلى to وبخطوة واحدة يمكن اختصار التابع السابق بالشكل:

```
> from:to
```

فمثلاً لتوليد أعداد من 2 إلى 30 نكتب:

```
> 2:30
[1]  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20
[2] 21 22 23 24 25 26 27 28 29 30
```

وللتابع rep الشكل العام التالي:

```
> rep(x,each)
```

ويقوم هذا التابع بتكرار عناصر الشعاع x بعدد من المرات قدرها each فمثلاً لتوليد العناصر:

```
1 1 2 2 3 3 4 4 5 5 6 6 7 7
```

نكتب:

```
> rep(1:7,each=2)
```

وإذا أردنا توليد الشعاع ككتلة واحدة نكتب:

```
> rep(1:7,2)
[1] 1 2 3 4 5 6 7 1 2 3 4 5 6 7
```

أمثلة: ولد العناصر الآتية:

4 6 9 4 6 9 4 6 9
4 4 6 6 6 6 9 9 9 9 9 9
4 6 6 9 9 9

الحل:

```
> rep(c(4,6,9),3)
> rep(c(4,6,9),c(2,4,6))
> rep(c(4,6,9),1:3)
```

الفلترة (Filtering) وبعض التطبيقات على الأشعة:

نقصد بالفلترة الوصول إلى بيانات ما وفق شرط معين، ولنأخذ مثال تطبيقي الآن:

1) املأ الشعاع x بالعناصر 7,5,6,7,8,9,11.
2) اوجد حجم الشعاع x.
3) اوجد العنصر الخامس من الشعاع x.
4) ضع الشعاع x في الشعاع y مضيفاً له القيم 4,8,9,10,13.
5) اطبع قيم الشعاع y عدا العنصر رقم 12.
6) اطبع أول 3 قيم من الشعاع y.
7) اطبع القيمة الأولى والخامسة والتاسعة من الشعاع y.

8) استبدل القيمة الأولى من الشعاع y بالقيمة 12.

9) استبدل القيمة الثالثة من الشعاع y بمربعها.

10) استبدل أول 3 قيم من الشعاع y بالقيمة 7.

11) استبدل القيمة السابعة والثامنة والتاسعة من الشعاع y بالقيم 7,8,9.

12) أوجد معكوس الشعاع y.

13) استبدل القيم التي هي أكبر من 8 في الشعاع y بالقيمة 4.

14) اطبع آخر 9 قيم من الشعاع y.

15) أضف للعناصر الزوجية في الشعاع y بالقيمة 1.

16) استبدل العناصر الفردية التي هي أقل من 7 بالقيمة 2.

17) اوجد كلاً من المجموع والمتوسط والوسيط والانحراف المعياري والانحراف المتوسط للشعاع y.

18) اسند نصف قيم الشعاع y للشعاع x1 والنصف الآخر للشعاع x2.

19) ولد الشعاع rp المكون من العناصر 5 5 5 4 4 3 3 3 3 3 3 2 2 2 2 2 1 1 1 بطريقة مختصرة.

الحل:

```
1)  > x<-c(7,5,6,7,8,9,11)
2)  > length(x)
3)  > x[5]
4)  > y<-c(x,4,8,9,10,13)
5)  > y[-12]
6)  > y[1:3]
7)  > y[c(1,5,9)]
8)  > y[1]=12
9)  > y[3]=y[3]^2
10) > y[1:3]<-7
11) > y[c(7,8,9)]<-c(7,8,9)
12) > rev(y)
13) > y[y>8]<-4
```

```
14) > y[(length(y)-8):length(y)]
15) > y[y%%2==0]<-y[y%%2==0]+1
16) > y[y%%2!=0 & y<7]<-2
17) > sum(y)
    > mean(y)
    > median(y)
    > sd(y)
    > sum(abs(y-mean(y)))/length(y)
18) > x1<-y[1:6]
    > x2<-y[7,12]
19) > rp<-c(rep(1,3),rep(2,5),rep(3,6),rep(4,2),rep(5,3))
```

التعليمة summary:

تستخدم التعليمة summary(x) بوسيط x (x شعاع أو مصفوفة أو كائن اختبار إحصائي ما) لحساب الإحصاءات الوصفية.

فلو كان x شعاع أو مصفوفة مثلاً:

```
> x<-rep(1:5,c(3,5,6,2,3))
> summary(x)
   Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
  1.000     2.000     3.000     2.842     3.500     5.000
```

أي يظهر لنا أصغر قيمة min وأكبر قيمة max والربيع الأول 1st Quartile والربيع الثاني 2st Quartile أو ما يعرف بالوسيط median والربيع الثالث 3st Quartile والمتوسط الحسابي mean.

وظيفة:

تمرين:

باستخدام لغة البرمجة الإحصائية R اكتب التعليمات التي توجد معاملات معادلة الانحدار التي من الشكل:

$$y = a + bx$$

حيث أن b تعطى بالعلاقة:

$$b = \frac{\sum x_i y_i - n\bar{x}\,\bar{y}}{\sum (x_i - \bar{x})^2}$$

و a تعطى بالعلاقة:

$$a = \bar{y} - b\bar{x}$$

حيث القيم المقدرة لـ y تعطى بالشكل:

$$pred_i = a + bx_i$$

معامل الارتباط r يعطى بالعلاقة:

$$r = \frac{\overline{xy} - \bar{x}\,\bar{y}}{\sigma_x . \sigma_y}$$

الخطأ $Error$ يعطى بالعلاقة:

$$error = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n - 2}}$$

العمليات على الأشعة:

تتم العمليات بشكل مباشر أو بإسناد الناتج لمتحول ثالث، فمثلاً:

```
> x<-c(1,3,5,7)
> y<-c(2,4,6,8)
> x+y
[1]  3  7 11 15
> x-y
[1] -1 -1 -1 -1
> x*y
[1]  2 12 30 56
> x/y
[1] 0.5000000 0.7500000 0.8333333 0.8750000
> x%%y
[1] 1 3 5 7
> y%/%y
[1] 1 1 1 1
```

ملاحظة: إذا كانت الأشعة ليست من نفس الطول فإن المترجم سوف يزيد طول الشعاع القصير من نفس عناصره وبنفس الترتيب حتى تصبح مساوية للشعاع الطويل.

مثال:

```
> x<-c(1,2)
> y<-c(3,4,5,6)
> x+y
[1] 4 6 6 8
```

توليد أرقام عشوائية:

لتوليد أرقام عشوائية من k إلى m نكتب:

```
> sample(k:m)
```

مثلاً:

```
> x <-1:40
> sample(1:40)
```

لتوليد n مجموعة بأرقام عشوائية من k إلى m نكتب:

```
> replicate (n, sample(k:m))
```

مثلاً:

```
> x<-5:15
> z<-replicate(4,sample(x))
> z
      [,1] [,2] [,3] [,4]
 [1,]   13   15    5   12
 [2,]    6   11   14    8
 [3,]    9    8    8   14
 [4,]    7    6    7   10
 [5,]   15    5   15    9
 [6,]   14   12   10   13
 [7,]   12    7    6    6
 [8,]   10   10   11   11
 [9,]    8   13    9   15
[10,]    5   14   12    5
[11,]   11    9   13    7
```

المقارنة بين الأشعة:

```
> x<-c(3,5,7,10)
> y<-c(2,5,8,9)
> x>y
[1]  TRUE FALSE FALSE  TRUE
> x!=y
[1]  TRUE FALSE  TRUE  TRUE
> x<y
[1] FALSE FALSE  TRUE FALSE
```

التعليمتان any و all:

```
> any(x==y)
[1] TRUE
> any(x>y)
[1] TRUE
> all(x==y)
[1] FALSE
```

```
> any(x!=y)
[1] TRUE
```

القيمة NULL (فراغ) والقيمة NA الغير موجودة:

```
> z<-c(1,4,6,NA)
> z
[1]  1  4  6 NA
> mean(z)
[1] NA
> mean(z,na.rm=TRUE)
[1] 3.666667
> sum(z)
[1] NA
> sum(z,na.rm=T)
[1] 11
```

```
> z<-c(1,4,6,NULL)
> z
[1] 1 4 6
> mean(z)
[1] 3.666667
> sum(z)
[1] 11
```

بعض التوابع المهمة:

التابع	الوظيفة
الرقم الصحيح الأصغر مباشرة من x	floor(x)
الرقم الصحيح الأكبر مباشرة من x	ceiling(x)
لتقريب العدد x لعدد n بعد الفاصلة	round(x, digits=n)

أمثلة:

```
> floor(5.56)
[1] 5
> floor(-3.45)
[1] -4
> ceiling(5.56)
[1] 6
> ceiling(-3.45)
[1] -3
> round(10/3,digits=3)
[1] 3.333
```

إطار البيانات Data Frame:

يمكن تعريف إطار البيانات Data Frame على أنه جدول يحتوي عدة أسطر وعدة أعمدة حيث يمثل كل عمود نوع محدد من البيانات وكل سطر بيانات فرد محدد، وله الصيغة العامة التالية:

```
> data_frame_name<-data.frame(vector1,vector2,…)
```

مثال:

```
> x <- c(2, 3, 5)
> y <- c("red", "green", "white")
> z <- c(T, F, T)
> df <- data.frame(n, s, b)
> df
    x     y     z
[1] 2   red  TRUE
[2] 3 green FALSE
[3] 5 white  TRUE
```

للتعامل مع أي شعاع من إطار البيانات نكتب اسم إطار البيانات متبوعاً بإشارة $ ثم اسم الشعاع المطلوب، مثلاً:

```
> df$y
[1] red   green white
```

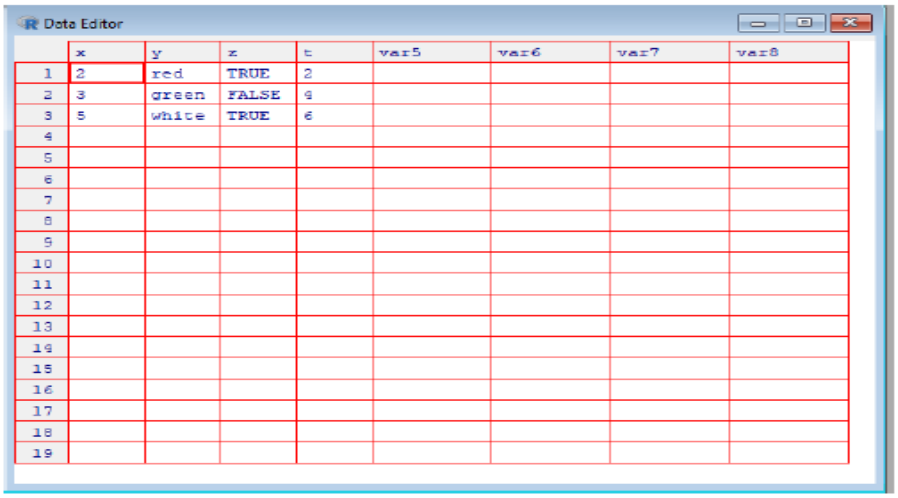
ويمكن إضافة متحول إلى إطار البيانات السابق بإحدى الأشكال الآتية:

```
> df$t<-c(2,4,6)
> df
     x      y     z  t
[1]  2    red  TRUE  2
[2]  3  green FALSE  4
[3]  5  white  TRUE  6
```

أو:

```
> fix(df)
```

فيفتح لنا محرر إطار بيانات بالشكل:



	x	y	z	t	var5	var6	var7	var8
1	2	red	TRUE	2				
2	3	green	FALSE	4				
3	5	white	TRUE	6				
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								

ويجب التنويه هنا أنه تمكننا التعلمية السابقة fix بالتعديل على إطار البيانات أيضاً.

يمكن إيجاد إحصاءات عامة حول Data Frame السابقة بالشكل:

```
> summary(df)
       x             y          z                t
 Min.   :2.000   green:1   Mode :logical   Min.   :2
 1st Qu.:2.500   red  :1   FALSE:1         1st Qu.:3
 Median :3.000   white:1   TRUE :2         Median :4
 Mean   :3.333                             Mean   :4
 3rd Qu.:4.000                             3rd Qu.:5
 Max.   :5.000                             Max.   :6
```

للوصول للسطر رقم r نكتب:

```
df[r,]
```

للوصول للعمود رقم c نكتب:

```
df[,c]
```

مثلاً:

```
> df[3,]
    x     y    z  t
[3] 5 white TRUE  6
> df[,3]
[1]  TRUE FALSE  TRUE
```

حفظ البيانات على القرص في مجلد العمل:

لنفهم هذا عن طريق مثال:

```
> x1<-c(1,4,9,13)
> db<-data.frame(x1)
> db$x2<-c(2,8,5,10)
> setwd("D:/myfile")
> save.image("db")
```

بإغلاق لغة R ونفتحها من جديد:

```
> db
Error: object 'db' not found
> setwd("D:/myfile")
> load("db")
> db
```

```
   x1 x2
[1]  1  2
[2]  4  8
[3]  9  5
[4] 13 10
```

استيراد البيانات من EXCEL:

يتم ذلك بإحدى الطريقتين:

- من ورقة EXCEL نسجل البيانات ونحفظ الملف باسم ما وليكن myfile من النوع csv(comma, delimited) ثم ننقل الملف إلى مجلد العمل الذي أنشأناه، ثم من داخل لغة R نكتب:

```
> db <-read.csv("test=csv", header=T,sep=";")
```

- من ورقة EXCEL نسجل البيانات ونحفظ الملف باسم ما وليكن myfile ومن النوع text(tab delimited) ثم ننقل الملف إلى مجلد العمل الذي أنشأناه، ثم من داخل لغة R نكتب:

```
> db <-read.table(myfile.text,header=T)
```

العامل ">" والعامل "<":

```
> ">"(4,2)
[1] TRUE
> ">"(5,9)
[1] FALSE
> "<"(4,2)
[1] FALSE
> "<"(5,9)
[1] TRUE
```

التابع which والتابع subset:

```
> x<-c(4,6,1,NA,5,NULL)
> x[x>3]
[1]  4  6 NA  5
> subset(x,x>3)
[1] 4 6 5
> which(x+x>4)
[1] 1 2 5
```

التابع ifelse:

للتابع ifelse الشكل العام الآتي:

```
> ifelse(condition,statement1,statement2)
```

والذي يختبر الشرط condition ففي حال تحققه تنفذ التعلمية statement1 وفي حال عدم تحققه تنفذ التعليمة statement2.

كما يمكن وضع تابع ifelse جديد مكان statement2 لإجراء مقارنة جديدة.

وكمثال على التابع ifelse سنعيد كتابة التعلمية التي تختبر وضع عدد فيما إذا كان موجباً أو سالباً أو صفراً:

```
> x<-5
> ifelse(x>0, "Positive", ifelse(x<0,"Negative","Zero"))
[1] "Positive"
```

مثال: برنامج يطبع العدد 1 إذا وجد الحرف p ويطبع 2 إذا وجد الحرف f ويطبع 0 فيما عدا ذلك:

```
> y<-c("f","p","p","p","f","f","a","a")
> ifelse(y=="p",1,ifelse(y=="f",2,0))
[1] 2 1 1 1 2 2 0 0
```

تابع الفرق diff:

هو تابع يساعد على حساب الفروق بين قيم شعاع x بخطوة قدرها lag وبدرجة فروق قدرها d ولها الشكل العام التالي:

```
> diff(x,lag,d)
```

مثال:

```
> x<-c(9,5,1,7,3,2,1,7,5,3)
> diff(x,1,1)
[1] -4 -4  6 -4 -1 -1  6 -2 -2
> diff(x,2,1)
[1] -8  2  2 -5 -2  5  4 -4
> diff(x,1,2)
[1]   0  10 -10   3   0   7  -8   0
```

ملاحظات حول dataframes: في حال كان حجم الـ dataframe ضخماً جداً، يمكن استخدام التعلمية head لاستعراض أو 6 أسطر فقط أو التعليمة tail لاستعراض آخر 6 أسطر... كما يمكننا إضافة وسيط للتعليمتين السابقتين لتحديد عدد أسطر آخر غير 6.

مثال:

```
> x<-c(5,7,4,2,9,1,-3,5,-8,10)
> y<-rep(c(1,2,5,3),1:4)
> db<-data.frame(x,y)
> db$z<-1:10
> db
```

```
    x  y  z
1   5  1  1
2   7  2  2
3   4  2  3
4   2  5  4
5   9  5  5
6   1  5  6
7  -3  3  7
8   5  3  8
9  -8  3  9
10 10  3 10
```

```
> head(db,3)
  x  y z
1 5  1 1
2 7  2 2
3 4  2 3
```

```
> tail(db)
    x  y  z
5   9  5  5
6   1  5  6
7  -3  3  7
8   5  3  8
9  -8  3  9
10 10 3 10
```

```
> tail(db,3)
    x  y  z
8   5  3  8
9  -8  3  9
10 10  3 10
```

انتهت المحاضرة