المحاضرات الثلاث الأول في البرامج الإحصائية٢
مقدمة في لغة R

(1-1)مقدمة:

لغة البرمجة الإحصائية R لغة مفتوحة المصدر Open Source Language ابتكرها روس إيهاكا وروبيرت جنتلمان في جامعة أوكلاند في نيوزيلاندا، ويعود سبب تسميتها بلغةR إلى اسم مبتكريها. وصدرت أول نسخة للغة عام 2000 .

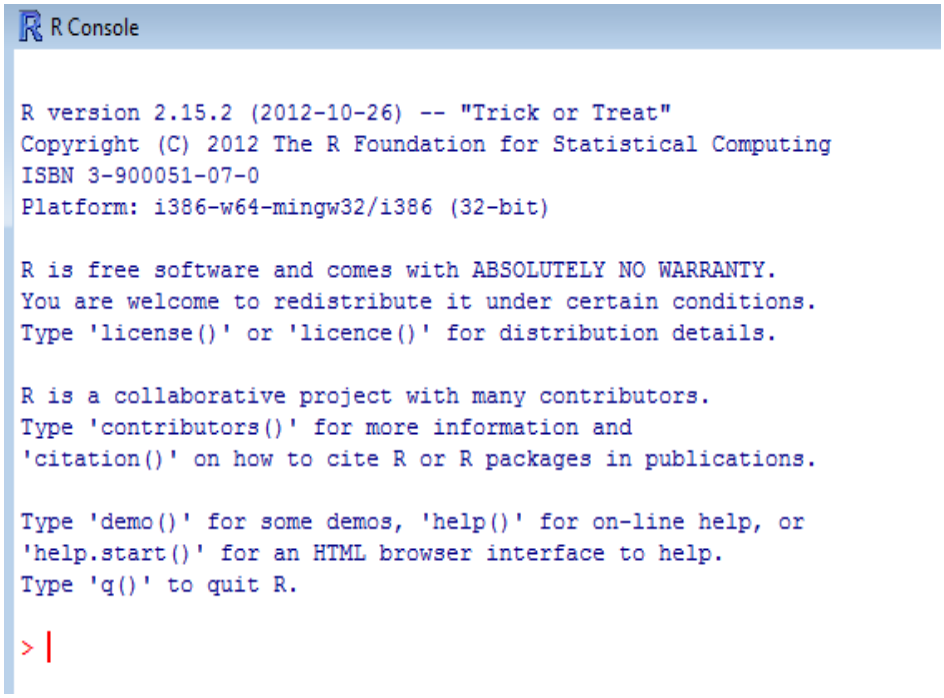
كما أن لغة R من أهم لغات تحليل المعطيات مبنية عل التوابع (Function language) أي أنها مزودة بكم هائل من التوابع التي تحل معظم المشاكل التي قد تواجهك، وتعتمد أوامرها على الكتابة المباشرة على شاشة الحاسب (Command Line) .

تتيح لك الفرصة بإضافة التوابع التي تناسبك أو التي هي من ابداعك الشخصي وبإمكانك عمل حزم (Package) خاصة بك من التوابع والخوارزميات وإضافتها إلى لغة R رسمياً حتى يستفيد منها غيرك وهذا المقصود بكون R مفتوحة المصدر.

تعمل لغة R على أشياء تسمى Objects وتتعامل مع كافة أنواع البيانات عددي، مركب، منطقي، و نصي.

عند تحميل اللغة على شاشة الحاسب تظهر في أعلى الصفحة عدة سطور تبين نموذج اللغة و آخر تعديل لها وتاريخ النموذج الحالي. وآخر سطر يكون فيه سهم باتجاه اليمين <-

الشكل العام لصفحة اللغة:



شكل(1-1) يبين واجهة اللغة عند بدء الإقلاع

يمكن تحميل اللغة من المواقع التالية وذلك حسب النظام الذي تعمل عليه:

1 – من أجل نظام الويندوز

MicrosoftWindows: http://cran.r-project.org/bin/windows/base/

2 – من أجل نظام ماكنتوش

MacOS: http://cran.r-project.org/bin/macosx/

3- من أجل نظام لينكس

Linux: http://cran.r-project.org/bin/linux/

يمكن معرفة الحزم الأساسية المحملة في النسخة كمايلي:

```
> getOption("defaultPackages")
[1] "datasets"  "utils"     "grDevices" "graphics"  "stats"     "methods   "
```

لإضافة أية حزمة أخرى ولتكن foo نكتب مايلي:

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

أما لتحميل الحزمة السابقة مع كل المكتبات التي تعتمد عليها نكتب:

```
install.packages("foo", depends = TRUE).
```

(1-2)المتحولات (المتغيرات) Variables

لإعطاء القيمة 1 للمتحول x نكتب أحد الأشكال التالية:

```
------------------------------------------------
>x<-1
>x=1
```

ويفضل استخدام الشكل الأول.

لإعطاء للمتحول m قيمة منطقية نكتب:

```
------------------------------------------------
>m<-TRUE
------------------------------------------------
```

ويمكننا بنفس الطريقة إعطاء للمتحول z قيمة مركبة (عقدية)

```
------------------------------------------------
>z<-2i
------------------------------------------------
```

وبنفس الطريقة يمكننا تخزين قيمة نصية في المتحول a

```
------------------------------------------------
>a<-"Ahmed"
------------------------------------------------
```

ملاحظة: يجب أن تكون حروف كلمة TRUE أحرف كبيرة والا لن يتحسسها الكمبيوتر

(1-3) بعض العمليات الحسابية المباشرة

عملية جمع مباشرة

```
------------------------------------------------
> 2 + 3 #
[1]      5
------------------------------------------------
> 4 * 5 / 6       # ضرب ثم قسمة
[1]      3.333333
------------------------------------------------
> 7^8             # 7مرفوعة للقوة ٨
[1]      5764801
------------------------------------------------
```

لاحظ أنه بعد اشارة # فإن الحاسب يهمله كأنه غير موجود.

نلاحظ أيضاً أن عدد الخانات بعد الفاصلة فقط ستة، ويمكن أن نزيد هذا العدد كمايلي:

```
------------------------------------------------
> options(digits = 16)
> 10/3 #      لاحظ خانات أكثر
[1] 3.333333333333333
------------------------------------------------
```

```
> sqrt(2)        #    الجذر التربيعي
[1] 1.414213562373095
-----------------------------------------------
> exp(1)         #    قيمة العدد النيبري e
[1] 2.718281828459045
-----------------------------------------------
> pi
[1] 3.141592653589793
-----------------------------------------------
```

العودة الى الشكل الافتراضي للخانات بعد الفاصلة

```
> options(digits = 7)   #
-----------------------------------------------
```

ملاحظة: يمكن زيادة عدد الخانات الى ?22/

(4-1)أنواع البيانات في R

1 – البيانات الصحيحة integer وهي من أصل لغة r

2 – البيانات من نوع Double وهي تمثل الأرقام الحقيقية

3 – البيانات من نوع character كل شيء موضوع ضمن اشارتي "" أو إشارتي ' '

4 – البيانات من نوع منطقي logical وهي البيانات التي تأخذ TRUE أو FALSE أو NA

```
-----------------------------------------------
> sqrt(-1)                    #   غير معرف
[1] NaN
-----------------------------------------------
> sqrt(-1+0i)                 #   معرف عدد عقدي
[1] 0+1i
-----------------------------------------------
> sqrt(as.complex(-1))        #   نفس الأمر السابق تماما
[1] 0+1i
-----------------------------------------------
> (0 + 1i)^2                  #   النتيجة يجب أن تكون -1
[1] -1+0i
-----------------------------------------------
> typeof((0 + 1i)^2)
[1] "complex"
-----------------------------------------------
```

لاحظ أن (1i)^2 تعطي نفس الجواب -1

NaN تعني أنه ليس رقم.

```
-----------------------------------------------
> LETTERS[1:5]
```

```
[1] "A" "B" "C" "D" "E"
-----------------------------------------------
> letters[-(6:24)]
[1] "a" "b" "c" "d" "e" "y" "z"
-----------------------------------------------
```

(5-1)الأمر mode

يستخدم لمعرفة نوعية المتحولات التي أدخلناها والشكل العام له هو:

Mode(اسم متحول)

سنطبق الأمر السابق على المتحولات السابقة سنجد:

```
-----------------------------------------------
>mode(x)
[1]      "numeric"
-----------------------------------------------
>mode(a)
[1]      "character"
-----------------------------------------------
>mode(m)
[1]      "logical"
-----------------------------------------------
>mode(z)
[1]      "complex"
-----------------------------------------------
```

(6-1)ادخال البيانات بشكل شعاع

والشعاع هو متحول يأخذ عدة قيم من نوع واحد. في لغة R نستخدم الحرف C لهذا الغرض ونضع القيم بين قوسين.

(1-1)مثال:

لدينا البيانات التالية:

X:3 4 5 9

والمطلوب ادخالها الى لغة R .

```
>x<-c(3,4,5,9)
```

وهذا يعني تماما أن: x1=3 x2=4 x3=5 x4=9

الآن لوكتبنا: >x

سنحصل على :

```
[1]      3        4        5        9
```

بنفس الطريقة يمكن ادخال مجموعة قيم نصية تحت اسم متحول بشكل شعاع.

```
>y<-c("Ali,"Ahmad","omar")
```

(7-1)حساب عدد القيم في الشعاع

يعطى بمساعدة الأمر length ويملك الشكل العام التالي: length(name)

```
-----------------------------------------------
>length(x)
[1]     4                يعني اربع قيم
-----------------------------------------------
>length(y)
[1]     3                يعني ثلاث قيم
-----------------------------------------------
```

ملاحظة: لغة R حساسة للأحرف الكبيرة والصغيرة مثلاً mx تختلف عن Mx

ملاحظة: يفضل عدم استخدام الرموز التالية: c q D F I T

لأنها مفاتيح لكلمات مفتاحية في اللغة.

(8-1)فضاء العمل ودليل العمل work space

في أي جزء من القرص الصلب ننشئ مجلدا خاصاً بالعمل الذي نقوم به ليتم تسجيل كافة العمليات والنتائج داخله.

يمكن معرفة مكان مجلد العمل كمايلي:

```
>getwd()
```

طبعا سوف يعطيك مكان مجلد العمل الافتراضي، لنفرض أننا أنشأنا مجلد عمل باسم test على القرص الصلب F ولأرشاد التطبيق الى هذا المجلد يجب من بداية العمل أن نكتب:

```
>setwd("F:/test")
```

سوف يقوم التطبيق بتخزين كافة أعمالك في الجلسة الحالية ضمن المجلد المشار له. الآن لوكتبنا:

```
>getwd()
```

نحصل على:

```
[1]     "F:/test"
```

(9-1)جمع شعاعين

يتم بالجمع المباشر أو بوضع الناتج في متحول ثالث وذلك بجمع كل عنصر من الشعاع الأول الى العنصر المقابل له من الشعاع الثاني وبنفس الترتيب.

مثال(1-2)

```
-----------------------------------------------
>x1<-c(1,2,3)
>x2<c(3,4,5)
>x3<-c(1,2,3)+c(3,4,5)
-----------------------------------------------
```

أو:

```
-----------------------------------------------
>x3<-x1+x2
>x3
[1]     2       6       8
-----------------------------------------------
```

أما لوكتبنا:

```
-----------------------------------------------
>x1-c(8,7,3)
-----------------------------------------------
```

نحصل على:

```
-----------------------------------------------
[1]     -7      -5      0
-----------------------------------------------
>x1*x2
[1]     3       8       15
-----------------------------------------------
>c(6,2,4)/x1
[1]     6       1       1.33
-----------------------------------------------
```

لاحظ أيضا أنه في عملية الضرب والقسمة والطرح تم أيضا ضرب أو قسمة أو طرح كل عنصر من الشعاع الأول بما يقابله من الشعاع الثاني.

ملاحظة: يمكن العودة الى الأوامر السابقة بالضغط على سهم للأعلى وكذلك يمكن استخدام السهم للأسفل للعملية المعاكسة.

(1-10)التعامل مع شعاعين لا يملكان نفس العدد من العناصر

عندما يتم التعامل مع شعاعين لا يملكان نفس العدد من العناصر فإن الحاسب سوف يزيد طول الشعاع القصير من نفس عناصره وبنفس الترتيب حتى تصبح مساوية للشعاع الطويل.

(1-3)مثال:

```
-----------------------------------------------
>x1<-c(1,2)
>x2<-c(2,1,1,1,1,1)
>x1+x2
[1]     3       3       2       3       2       3
-----------------------------------------------
```

وقد تمت العملية كمايلي:

```
        X1      1       2       1       2       1       2
+       X2      2       1       1       1       1       1
                3       3       2       3       2       3
```

ومن أجل حذف المتغيرx2 نتبع التعليمة:

```
>rm(x2)
```

ومن أجل حذف جميع المتغيرات الموجودة سابقاً نكتب التعليمة:

```
>rm(list=objects())
```

(1-11)بعض التوابع للتعامل مع الأشعة:

التابع sum ويملك الشكل العام التالي: sum(vector)

التابع mean له الشكل العام التالي: mean(vector)

(1-4)مثال:

لدينا البيانات التالية والمطلوب أكتب الكود اللازم لحساب المتوسط لها.

X:50 60 45 45 60

الحل:

```
-------------------------------------------------
>x<-c(50,60,45,45,60)
>s1<-sum(x)
>n<-length(x)
>mn<-s1/n
>mn
[1]  52
-------------------------------------------------
```

ويمكن اخراج قيمة المتوسط مباشرة عن طريق استخدام التابع mean

```
>mean(x)
[1]  52
-------------------------------------------------
```

(1-12) اطار البيانات Data Frame

هو جمع لعدة متغيرات ضمن موضوع واحد ويمكن تشكيله وفق الصيغة التالية:

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

(1-5)مثال:

لدينا البيانات التالية:

X1:3 4 5 9

X2:1 2 3 6

المطلوب ادخال هذه البيانات الى لغة R وعمل اطار بيانات لها أسمه db1 .

الحل:

```
-------------------------------------------------
>x1<- 3,4,5,9
>x2<- 1,2,3,6
>db1<-data.frame(x1,x2)
-------------------------------------------------
```

الأن يمكننا استعراض محتويات الاطار كمايلي:

```
-------------------------------------------------
```

```
>db1
      X1    X2
[1]    3     1
[2]    4     2
[3]    5     3
[4]    9     6
```

للتعامل مع أي شعاع ضمن اطار البيانات يجب أن يكتب أسم اطار البيانات أولا ثم اسم الشعاع المطلوب.

مثال(1-6)

أحسب المتوسط للشعاع x1 في اطار البيانات السابق.

الحل:

```
-----------------------------------------------
>mean(db1$x1)
[1]    5.25
-----------------------------------------------
```

(1-13)إضافة متحول الى اطار البيانات

١. الطريقة الأولى

لنفرض أننا نريد اضافة الشعاع x3:2,4,6,8 الى اطار البيانات السابق.

الحل:

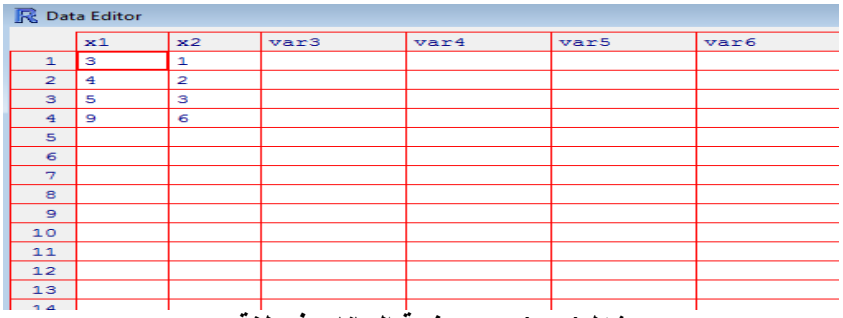
```
-----------------------------------------------
>db1$x3<-c(2,3,4,6,8)
>db1
      X1    X2    X3
[1]    3     1     2
[2]    4     2     4
[3]    5     3     6
[4]    9     6     8
```

٢. الطريقة الثانية

نفتح صفحة عمل تضم المتحولات السابقة ويمكن أن نضيف الى هذه الصفحة ما نشاء من متحولات جديدة، حتى يمكننا تعديل المتحولات القديمة.

```
>fix(db1)
```

سنحصل على:


Data Editor

	x1	x2	var3	var4	var5	var6
1	3	1				
2	4	2				
3	5	3				
4	9	6				
5						
6						
7						
8						
9						
10						
11						
12						
13						

شكل(1-2) يبن صفحة البيانات في لغة R

يفيدنا اطار البيانات في اختصار العمليات الاحصائية فمثلا يمكننا الحصول على احصاءات وصفية لكل اطار بما فيه من متحولات كمايلي:

>summary(db1)

	X1	X2
Min	3	1
1stQ1	3.75	1.75
Median	4.5	2.5
Mean	5.25	3
3rd Q4	6	3.75
Max	9	6

ملاحظة: حساب الربيعات يتم بعدد من الطرق منها:

- طريقة أولى
- نرتب القيم تصاعديا
- الربيع الأول هو القيمة ذات الترتيب 0.25(n+1) (n عدد قيم الشعاع)
- الربيع الثالث هو القيمة ذات الترتيب 0.75(n+1)

وحسب هذه الطريقة فإن الربيع الأول للشعاع الأول هو القيمة ذات الترتيب 0.25(4+1) =1.25

Q1=(3+4)/2=3.5

الربيع الثالث هو القيمة ذات الترتيب 0.75(4+1)=3.75 وبالتالي الربيع الثاث هو: Q3=(5+9)/2=7

- طريقة ثانية

إذا كان عدد قيم العينة يكتب بالشكل 4k+1 فإن الربيع الأول والثالث هما :

Q1=0.25(رقم القيمةk) +0.75(القيمةk+1)

Q3=0.75(القيمةk+1) +0.75(القيمة3k+1)

- طريقة ثالثة

الربيع الأول هو وسيط النصف الأول من البيانات

الربيع الثالث هو وسيط النصف الثاني من البيانات

- طريقة رابعة

الربيع الأول هو وسيط النصف الأول من البيانات بعد حذف الوسيط العام
الربيع الثالث هو وسيط النصف الثاني من البيانات بعد حذف الوسيط العام
نظام المساعدة في لغة R
للحصول على أية مساعدة فيما يخص أي أمر نكتب إشارة استفهام أمام الأمر المطلوب.
مثال: ?mean> يعطينا الشكل العام لهذا الأمر وكيف يستخدم

Default S3 method:
mean(x, trim = 0, na.rm = FALSE, ...)

Arguments

X	An **R** object. Currently there are methods for numeric/logical vectors and date, date-time and time interval objects, and for data frames all of whose columns have a method. Complex vectors are allowed for trim = 0, only.
trim	the fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to or from other methods.

Value

If trim is zero (the default), the arithmetic mean of the values in x is computed, as a numeric or complex vector of length one. If x is not logical (coerced to numeric), numeric (including integer) or complex, NA_real_ is returned, with a warning.

If trim is non-zero, a symmetrically trimmed mean is computed with a fraction of trim observations deleted from each end before the mean is computed.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) The New S Language. Wadsworth & Brooks/Cole.

See Also

weighted.mean, mean.POSIXct, colMeans for row and column means.

Examples

```
x <- c(0:10, 50)
xm <- mean(x)
c(xm, mean(x, trim = 0.10))
```

(14-1)معرفة المتحولات الداخلة في اطار البيانات

تعطى أسماء المتحولات الداخلة في اطار البيانات بمساعدة Objects

---

```
>objects()
[1]   "db1"    "x1"    "x2"
```

(15-1)حفظ البيانات على الديسك في مجلد العمل

لنشرح هذه العملية من خلال الخطوات والمثال الأتي:

```
>x<C(1,2,3,4)
```

وضع تعريف لمجلد العمل على القرص

```
>setwd("F:/rcource")
```

تشكيل اطار بيانات فيه متحول واحد هو x

```
>db<-data.frame(x)
>db
X
1
2
3
4
```

نضيف متحول جديد y الى اطار البيانات السابق

```
>db$y<-c(2,4,6,8)
>db
        X       y
1       1       2
2       2       4
3       3       6
4       4       8
```

نحفظ البيانات على الديسك وفقا للمسار المكتوب في أمر setwd

```
>save.image("db")
```

نقوم الآن بالخروج من لغة R ونفتحها من جديد ونكتب:

```
>db
 Error object db not found
```

وذلك لأن البيانات غير موجودة في الذاكرة بل على الديسك.

لتحميل البيانات الموجودة في اطار البيانات db نتبع الخطوات التالية:

```
-----------------------------------------------
>setwd("F:/rcource")
>load("db")
>db
X       y
1       1       2
2       2       4
3       3       6
4       4       8
-----------------------------------------------
```

(16-1)استعراض أسماء الاطارات الموجودة في مجلد العمل

يعطى بمساعدة dir وتملك الشكل العام التالي: ()dir

```
-----------------------------------------------
>dir()
[1]     "db"
-----------------------------------------------
>x1<-c(2,4)
>y1<-c(3,6)
>db2<-data.frame(x1,y1)
>save.image("db2")
-----------------------------------------------
>dir()
[1]     "db"    "db2"
```

نلاحظ ظهور اطار البيانات الثاني بعد تخزينه على الديسك.

(17-1)المساعدة العامة

تعطى بالأمر help وتملك الشكل العام التالي:

```
>help.start
```

(18-1)استيراد البيانات من Excel I

١. نفتح صفحة بيانات Excel ونسجل البيانات التي نريد ولتكن كمايلي:

X1	X2
4	5
8	8
8	2
12	5
15	3

٢. من قائمة file نختار save as ونكتب اسم الملف الذي نريد حفظ البيانات تحته وليكن rrr (يجب أن نختار نوع الملف هو من نوع : text(tab delimited)

٣. ننسخ الملف المذكور ونلصقه في مجلد العمل الذي تم انشاؤه من قبلنا.
٤. الأن من داخل لغة R نكتب:

```
>db3<-read.table(rrr.text,header=TRUE)
```

سيتم تحميل البيانات وتسميتها باطار جديد هو db3 .
أو يمكننا اختيار ملف البيانات بعد تطبيق التعليمة التالية:

```
>db4<-read.table(file.choose(),header=TRUE)
```

ملاحظة: إذا أهملنا كلمة header=TRUE سيعتبر الحاسب أن كل البيانات هي نصية وبالتالي لن نستطيع اجراء عمليات حسابية عليها.

(19-1)استيراد البيانات من Excel II

١. من ورقة excel نسجل الملف الذي يحوي البيانات وعند الحفظ نختار نوع الملف من النوع: csv(comma, delimited)
٢. ننقل الملف الى مجلد العمل ولنفرض أننا سجلناه باسم test
٣. من داخل لغة R نكتب

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

(20-1)العمليات على الأرقام

١. التابع floor يعطي الرقم الصحيح الذي ليس أكبر من الرقم المعالج(الرقم الصحيح الأصغر مباشرة من الرقم المعالج).
٢.

```
>floor(2.23)
[1]  2
-----------------------------------------------
>floor(-2.23)
[1]  -3
-----------------------------------------------
```

٣. التابع ceiling يعطي الرقم الصحيح الذي ليس أصغر من الرقم المعالج(الرقم الصحيح الأكبر مباشرة من الرقم المعالج).
٤.

```
>ceiling(2.23)
[1]  3
-----------------------------------------------
>ceiling(-2.23)
[1]  -2
-----------------------------------------------
```

ملاحظة: يمكن تطبيق التوابع السابقة على شعاع كامل دفعة واحدة.

(7-1)مثال:

```
>x<-c(1.2,3.25,-5.4)
```

```
>floor(x)
[1]  1  3  -6
-----------------------------------------------
>ceiling(x)
[1]  2  4  -5
-----------------------------------------------
```

٥. التابع round يستخدم للتقريب الحسابي لعدد من الأرقام بعد الفاصلة.

٦.
```
-----------------------------------------------
>round(x,digits=1)
[1]  1.2  3.2  -5.4
-----------------------------------------------
```

(21-1)توليد الأرقام العشوائية:

ولد أرقام عشوائية من 1 الى 12 وخزنها في المتحول x

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

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

8	8	4
12	7	9
10	11	1
4	2	6

تشكيل شعاع واعطاؤه الأرقام من خمسة الى ثمانية

```
>x<-5:8
[1]     5       6       7       8
```

تشكيل شعاع واعطاؤه القيم من خمسة الى واحد

```
>x<-5:1
[1]     5       4       3       2       1
```

لاحظ أن الخطوة في كلا الحالتين هي واحد سواء بالزيادة أو بالنقصان.

```
>i<-5
>1:i-1
[1]     0       1       2       3       4
```

بينما لوكتبنا:

```
>i<-5
>1:(i-1)
[1]     1       2       3       4
```

(22-1)توليد الأرقام بواسطة seq

الشكل العام للتعليمة هو:

```
Seq(from=..,to=…,by=…)
>x<-seq(from=12,to=30,by=3)
>x
[1]     12      15      18      21      24      27      30
```

```
>seq(from=1.1,to 2,length=10)
```

```
[1]     1.1     1.2     1.3.    1.4.    1.5     1.6     1.7     1.8     1.9
        2
-----------------------------------------------
```

أما لوكتبنا:

```
-----------------------------------------------
>seq(from=1.1,to 2,length=11)
[1]     1.1     1.19    1.28    1.37    1.46    1.55    1.64    1.73    1.81
        1.9     2
-----------------------------------------------
```

(23-1)الإعادة repeating

```
>x<-rep(9,3)
>x
[1]     9       9       9
---------------------------------
>rep(c(1,2,2),3)
[1]     1 2 2   1 2 2   1 2 2
---------------------------------
>rep(1:3,2)
[1]     1 2 3   1 2 3
---------------------------------
>rep((2,3,4),each=2)
[1]     2 2     3 3     4 4
---------------------------------
>rep(c(2,3,4),2)
[1]     2 3 4   2 3 4
---------------------------------
```

(24-1)التعليمات any و all

تكون نتيجة استخدام هذان الأمران TRUE أو FALSE

```
-----------------------------------------------
>x<-1:10
>any(x>8)
[1]      TRUE
-------------------------------------
```

طبعا الشعاع x يحوي القيم من 1 الى 10 وبالتالي هناك قيم أكبر من الثمانية وهي القيم 9 و 10 لذلك كانت النتيجة صحيحة.

بينما:

```
-----------------------------------------------
>any(x>90)
[1]     FALSE
-----------------------------------------------
```

```
>all(x>40)
[1]     FALSE
-----------------------------------------------
>all(x>0)
[1]     TRUE
-----------------------------------------------
```

(25-1)أشعة الدخل والخرج vector in & vector out

ويقصد بها تطبيق عملية ما على كل عنصر من الشعاع.

```
-----------------------------------------------
>u<-c(5,2,8)
>v<-c(1,3,9)
>u>v
[1]     TRUE        FALSE           FALSE
-----------------------------------------------
```

تتم مقارنة العنصر الأول من u مع العنصر الأول من v والثاني من u مع الثاني من v

وهكذا حتى نهاية عناصر الشعاعين.

(26-1)القيم غير الموجودة وغير ذات القيمة NA & NULL

اذا اعتبرنا القيمة غير موجودة نسميها NA واذا اعتبرنا مكان القيمة فراغ لا شيء نسميها NULL.

```
-----------------------------------------------
>x<c(88,NA,12,168,13)
>x
[1]     88      NA      168     13
-----------------------------------------------
>mean(x)
[1]     NA
-----------------------------------------------
```

عند وجود قيمة NA لا يمكن تطبيق العمليات الحسابية على الشعاع، ويجب أزالتها من الشعاع.

```
-----------------------------------------------
>mean(x,na.rm=T)
[1]     70.25
-----------------------------------------------
```

في هذه الحالة اعتبرنا القيمة الثانية غير موجودة وبالتالي عند حساب المتوسط تمت القسمة على 4 أي 281/4=70.25(rm تعني remove)

```
-----------------------------------------------
>x<c(88,NULL,12,168,13)
>mean(x)
[1]     70.25
```

أي عند وجود قيمة من نوع Null يقوم الحاسب باهمالها مباشرة.

ملاحظة: عند وجود كلمة NA بين قيم عددية يتعامل معها الحاسب كقيمة عددية ، واذا كانت موجودة بين قيم نصية يعتبرها الحاسب نصاً.

(1-8)مثال:

```
---------------------------------------------
>x<-c(5,NA,10)
>mode(x[1])
[1]     "numeric"
---------------------------------------------
---------------------------------------------
>mode(x[2])
[1]     "numeric"
---------------------------------------------
>y<-c("abc",NA,"de")
>mode(y[2])
[1]     "character"
---------------------------------------------
```

(1-27) العامل ">" والعامل "<" & operator "<"

```
>">"(2,1)                    هل ٢ هي أكبر من ١؟
[1]     TRUE
---------------------------------------------
>"<"(7,2)                    هل ٧ هي أصغر من ٢ ؟
[1]     FALSE
---------------------------------------------
>z<-c(5,2,-3,8)
>j<-z*z>8          8  قارن قيم المتحول السابق والتي مربعها أكبرمن
>j
[1]     TRUE        FALSE           TRUE        TRUE
---------------------------------------------
>z[j]     ماهي القيم التي تحقق المتراجحة السابق ؟
[1]     5       -3      8
---------------------------------------------
```

(1-9) مثال:

لدينا البيانات التالية والمطلوب ضع كل القيم التي هي أكبر من ثلاثة تساوي الصفر.

```
---------------------------------------------
>x<-c(1,3,6,2,25)
>x[x>3]<-0
```

```
>x
[1]     1       3       0       2       0
```

(28-1) التصفية مع وجود تابع
Filtering With Function Subset

```
>x<-c(6,1,3,NA,12)
>x[x>5]
[1]     6       NA      12
```

لاحظ القيمة NA أكبر من أية قيمة عددية.
بينما لو كتبنا:

```
>subset(x,x>5)
[1]     6       12
```

لاحظ باستخدام subset يقوم الحاسب بإهمال القيمة NA ولا تدخل في عملية التصفية.

(29-1)تابع الاختيار which

يستخدم بشكل مشابه للتعليمات السابقة

(10-1)مثال:

```
>z<-c(5,2,-3,8)
>y<-which(z*z>8)
>y
[1]     1       3       4
```

أي أن القيم الأولى والثالثة والرابعة من الشعاع z المتراجحة السابقة.
ملاحظة: أن التابع which يتجاهل القيم NA & NULL

```
>z<-c(2,3,NA,1)
>y<-which(z*z>8)
>y
[1]     2
```

```
>z<-c(2,3,NULL,1)
>which(z*z>8)
[1]     2
```

(30-1) التوابع IF, THEN, ELSE,IFELSE

الشكل العام للتابع ifelse:

Ifelse(b,u,v)

b- متحول منطقي

v & U – متحولات عادية.

إذا كانت المقارنة صحيحة تطبع قيمة u والعكس تطبع قيمة v

(11-1)مثال:

```
-----------------------------------------------
>x<-1:10
>x
[1]  1  2  3  4  5  6  7  8  9
     10
-----------------------------------------------
>y<-ifelse(x%%2==0,5,12)
>y
[1]  12  5  12  5  12  5  12  5  12
     5
-----------------------------------------------
```

والأمر السابق يعني إذا كان حاصل قسمة x على 2 يساوي الصفر أعطي للمتحول y القيمة ٥ والعكس أعطيه القيمة 12 .

(12-1)مثال:

```
-----------------------------------------------
>x<-c(3,2,5)
>y<-ifelse(x>4,2*x,4*x)
>y
[1]  12  8  10
-----------------------------------------------
```

وهذا يعني قارن قيم المتحول X إذا كانت أكبر من 4 أعطي للمتحول y ضعفها والعكس أعطي للمتحول y أربعة أضعافها.

(31-1)تابع ifelse المتداخل

إذا كانت لدينا البيانات التالية التي تصف الذكور m والاناث f وغير المحدد i أعطي قيمة 10 للرجل والقيمة 15 للأنثى والقيمة 20 لغير المحدد وخزن النتائج في متحول y.

```
-----------------------------------------------
>g<-c("m","f","f","i","m","m","f")
>y<-ifelse(g=="m",10,ifelse(g=="f",15,20))
[1]  10  15  15  20  10  10  15
-----------------------------------------------
>m<-which(g=="m")
```

```
>f<-which(g=="f")
>i<-which(g=="I")
>m
[1]     1       5       6
-----------------------------------------------
>f
[1]     2       3       7
-----------------------------------------------
>i
[1]     4
-----------------------------------------------
```

(32-1) المساواة ==

تعمل اشارة المساواة ليس كما نظن بل تقوم باختبار تساوي قيم متحول الى قيم أخرى وتكون النتائج صح أو خطأ.

(13-1)مثال:

```
-----------------------------------------------
>x<-1:3
>x
[1]     1       2       3
-----------------------------------------------
>y<-c(1,3,4)
>x==y
[1]     TRUE            FALSE                   TRUE
-----------------------------------------------
```

هما تمت مقارنة كل قيم من المتحول x مع قيم y الموافقة لها بالترتيب الأولى مع الأولى و الثانية مع الثانية، وهكذا....

أما إذا أردنا مقارنة قيمتين فقط نكتب:

```
-----------------------------------------------
>"=="(3,2)
[1] FALSE
-----------------------------------------------
>i<-2
>"=="(I,2)
[1] TRUE
-----------------------------------------------
```

يمكن الاستعاضة عن == بـ identical

مثال:

```
-----------------------------------------------
>x<-1:3
>y<-c(1,3,4)
```

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

```
>identical(x,y)
[1]    FALSE
```

ملاحظة: يجب توخي الحذر عند استخدام identical

(14-1)مثال:

```
>x<-1:2
>x
[1]      1        2
```

```
>y<-c(1,2)
>y
[1]      1        2
```

```
>identical(x,y)
[1]      FALSE
```

السبب هو:

```
>type of(x)
[1] "integer"
```

```
>typeof(y)
[1] "double"
```

(33-1) تابع الفرق diff

الشكل العام لهذا التابع هو:

Diff(u, lag, deference)

حيث أن:

U – هو المتحول أو الشعاع المراد حساب الفروق له

Lag –تشير الى مقدار الخطوة.

Deference – تشير الى درجة الفروق.

ملاحظة: إن أهملنا قيم الوسطاء الثاني والثالث أي diff(u) ستحسب الفروق الأولى بخطوة مقدارها ١.

مثال:

أحسب الفروق الأولى للمتحول التالي:

```
>u<-c(1,6,7,2,3,5)
>diff(u)
[1]     5       1       -5      1       2
```

والفرق هو أن نطرح كل رقم من الذي يسبقه

$\Delta u=u_i-u_{i-1}$

الفروق الثانية هي فرق أول للفروق الأولى.

$\Delta^2 u=\Delta(\Delta u)= \Delta(5,1,-5,1,2)$

$=-4 \quad -6 \quad 6 \quad 1$

```
>diff(u,,2)
[1]     -4      -6      6       1
```

إذا كانت الخطوة 2 بدل 1 يتم طرح الرقم الثالث من الأول والخامس من الثالث وهكذا

$\Delta_2 u=(7-1),(2-6),(3-7),(5-2)$

$=6, -4, -4, 3$

الدليل السفلي يشير الى الخطوة.

```
>diff(u,2,1)      استخرج الفروق الأولى بخطوة مقدارها اثنان.
[1]     6       -4      -4      3
```

إذا أردنا معرفة اشارات الأرقام بعد الفروق نكتب:

```
>sign(diff(u))
[1]     1       1       -1      1       1
```

(34-1) استعراض البيانات بطرق مختلفة

اذا كانت لدينا البيانات التالية مسجلة تحت اسم اطار البيانات db

---

```
>db
 X        Y
 25       1
 12       2
 30       3
 45       4
```

إن الأوامر التالية تعطي نفس النتيجة.

```
>db
```

```
>head(db)
>tail(db)
>edit(db)
```

بينما:

```
>head(db,2)
 X        Y
 25       1
 12       2
```
تعطي أول سطرين من قيم متحولات اطار البيانات db

```
>tail(db,1)
 X        Y
 45       4
```
تعطي أخر سطر من بيانات اطار البيانات db

أما لو كتبنا db<-edit(db)> سوف يفتح اطار البيانات واي تعديلات تجريها سوف يتم حفظها تلقائيا تحت نفس اسم اطار البيانات.

(35-1)التعامل مع أرقام محددة من الشعاع

اذا كان لدينا الشعاع:

```
>b<-c(2,4,7,8,11,15,22)
```
يعطي العنصر الرابع من الشعاع
```
>b[4]
[1]  8
```

يعطي العناصر الثلاث الأولى من الشعاع
```
>b[1:3]
[1]  2   4   7
```

تعطي العناصر من الثاني الى الثالث
```
>b[2:3]
[1]  4   7
```

يعطي كل العناصر التي هي أصغر تماما من 8
```
>b[b<8]
[1]  2   4   7
```

يعطي كل العناصر التي هي أكبر من ثمانية وأقل من 22
```
>b[b>8 & b<22]
[1]  11  15
```

العناصر التي هي أقل من ثمانية أو أكبر من 15
```
>b[b<8 | b>15]
[1]  2   4   7   22
```

ملاحظة: مختلف تطبيقات R موجودة على العنوان
:http://CRAN.R-Project.org

كما يمكن تحميل اللغة من الموقع المذكور.

(1-36) تمارين غير محلولة.

١- لتكن لدينا البيانات التالية:

جدول(1-1) بيانات عشوائية عن ثلاث متحولات

X1	X2	X3
1	10	11
2	9	15
3	8	22
4	7	33
5	11	17
6	12	12
7	13	21

المصدر: بيانات افتراضية.

المطلوب.

- أدخل البيانات الى محرر لغة R
- شكل اطار بيانات اسمه db
- سجل اطار البيانات في المكان الافتراضي لتسجيل البيانات
- احسب متوسط المتحول x1 من خلال اطار البيانات
- احسب الربيعات لكل المتحولات
- أحسب المتوسطات لكل المتحولات
- أحسب الانحرافات المعيارية لكل المتحولات
- أضف لاطار البيانات متحولاً جديداً بالقيم التالية:

X4:2 3 5 7 2 8 7

- أحذف من اطار البيانات المتحول x2
- سجل بيانات الاطار ضمن المسار التالي:f:/rcourse