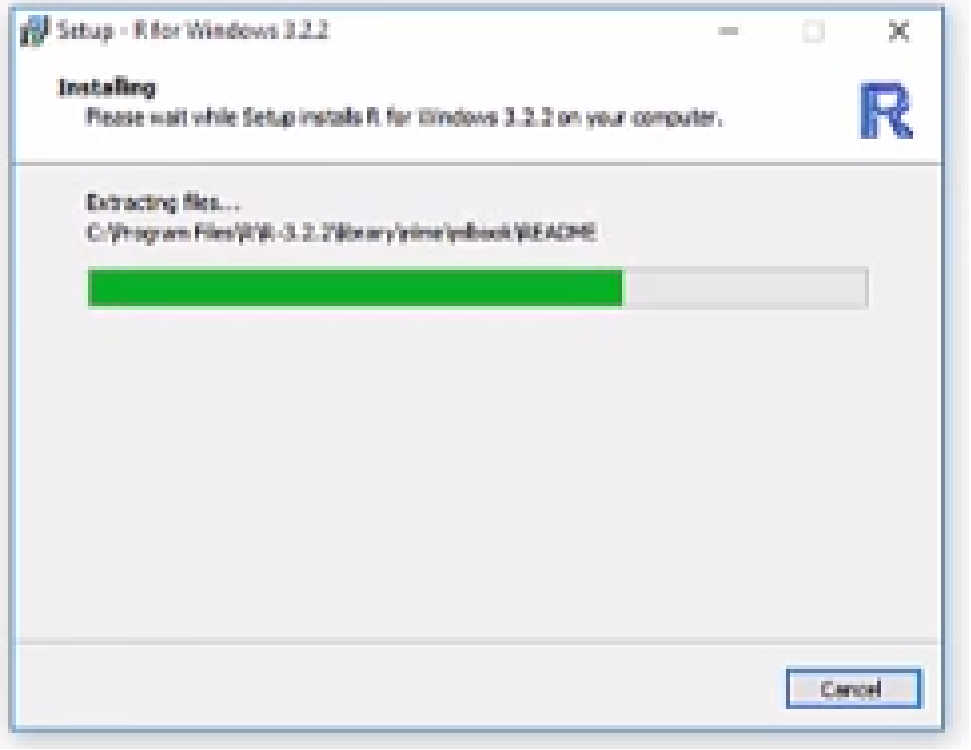
مقدمة مبسطة في لغة R مع تطبيقات إحصائية

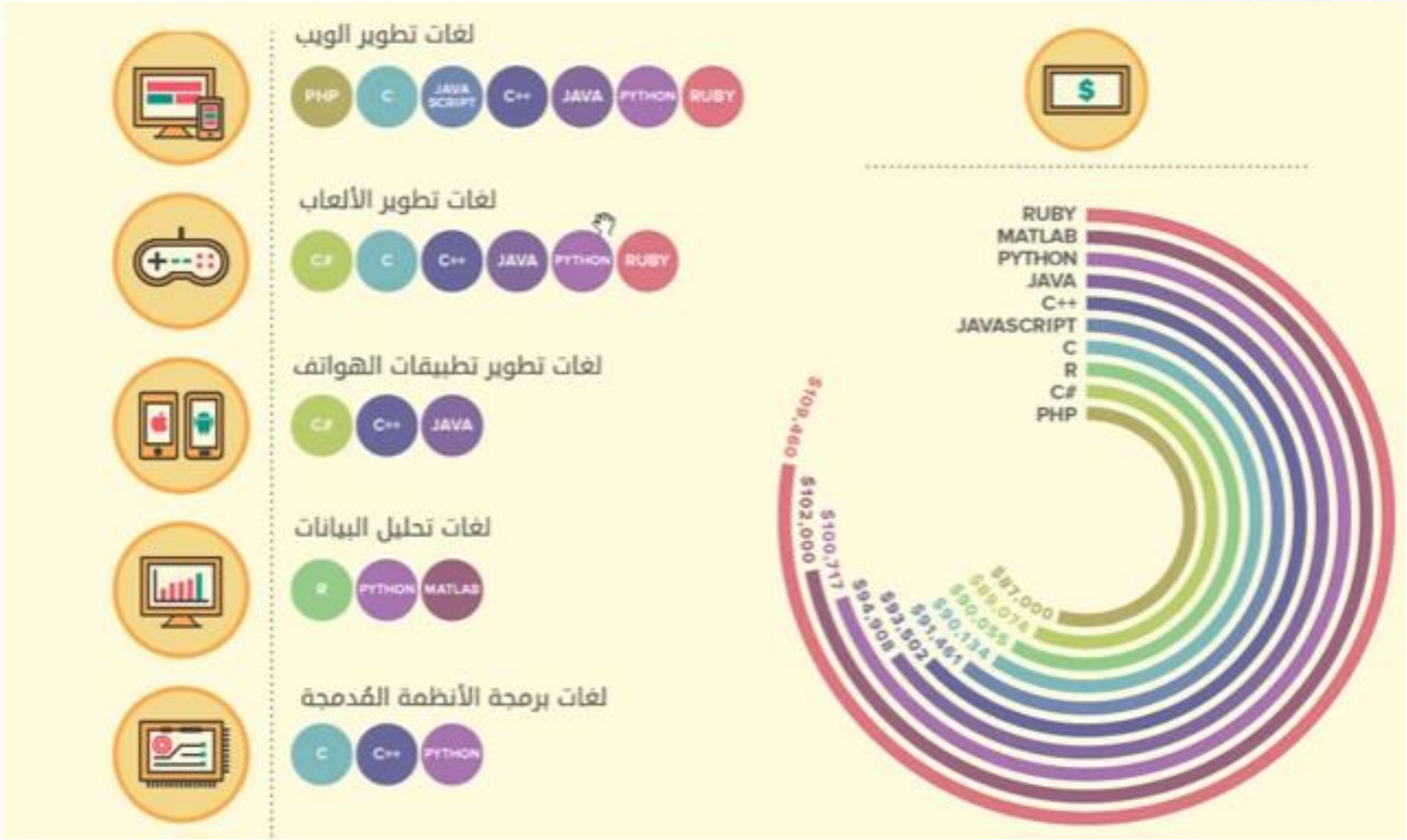
Introduction to R Language with Statistical Applications

سبأ محمد علوان

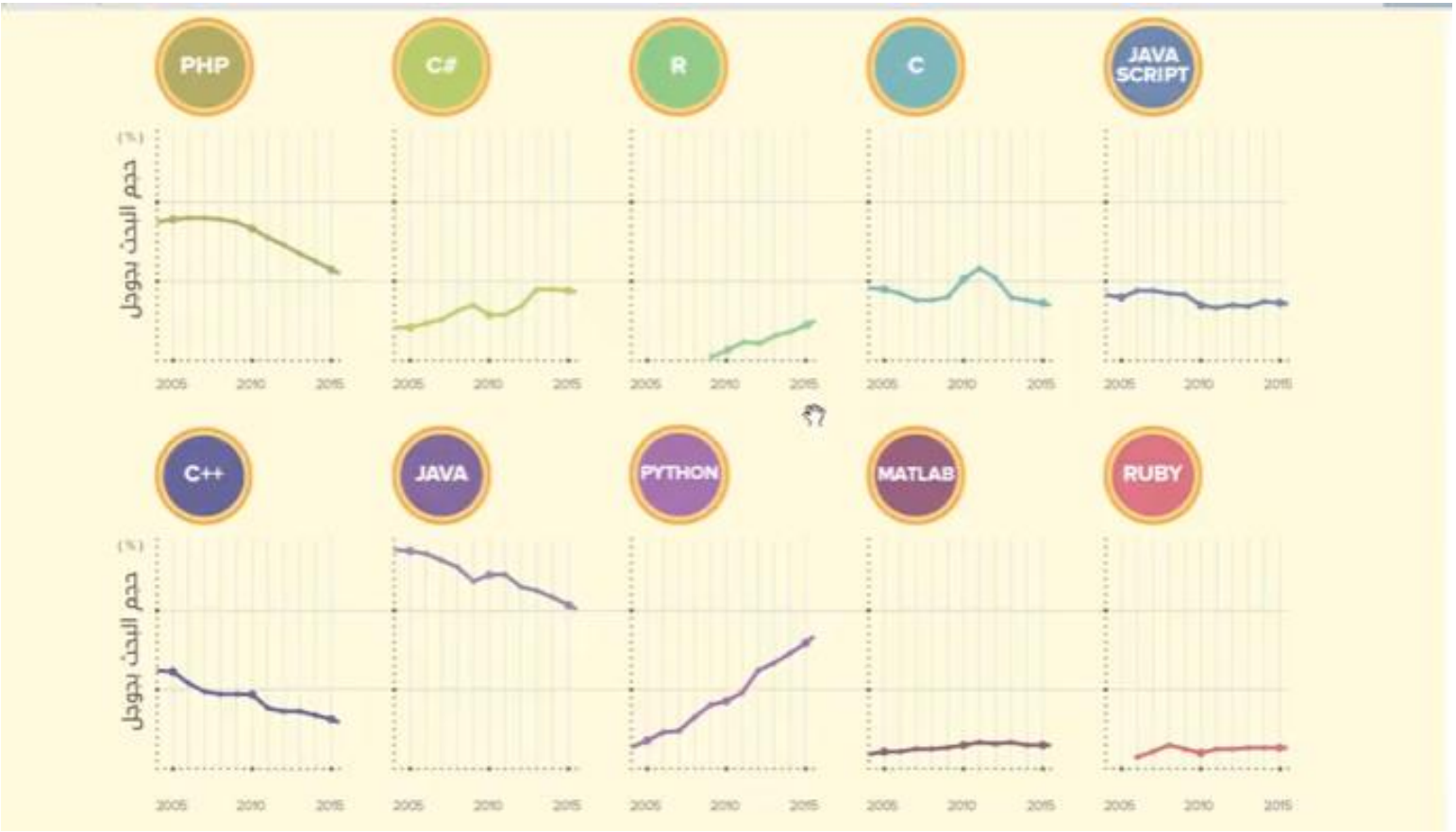
طريقة تثبيت اللغة
أنظر الفيديو في الرابط أدناه

https://www.youtube.com/watch?v=aUxVtzj5z18



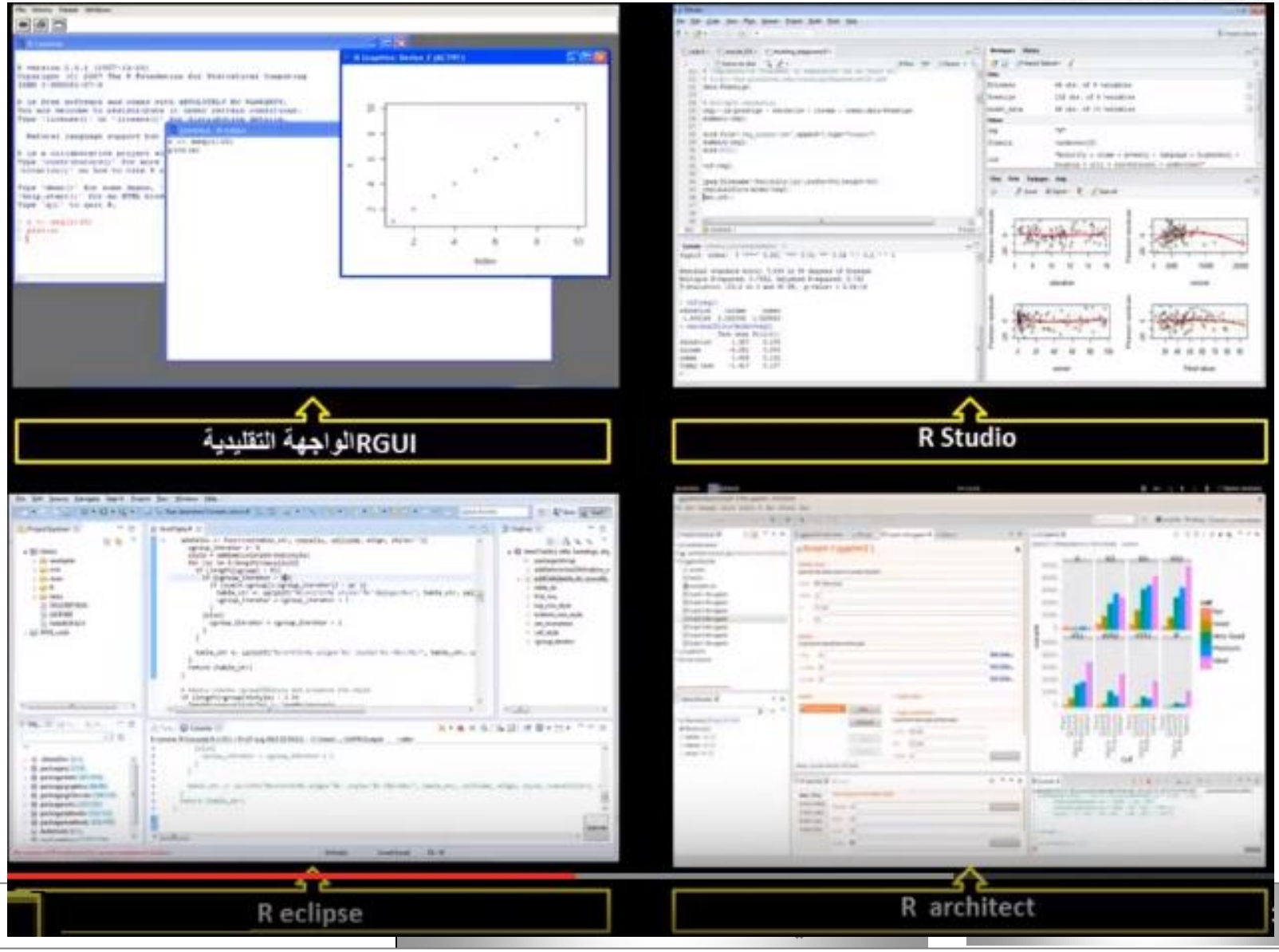


لغة R ، PYTHON و MATLAB الأكثر صيتا في تحليل البيانات



تطور الطلب على لغة R مقارنة بغيرها من اللغات ، حيث يصعد صيتها مع PYTHON

واجهات مختلفة للغة R ، تطور و مميزات



المساعدة فى R

للحصول على المساعدة في R فقط أكتب علامة استفهام ? قبل الامر المراد الاستفسار عن طريقة عمله أو من خلال كتابة الامر help () ثم نضع بين القوسين الامر المراد الاستفهام عنه ، حيث سيتم نقلنا الى R Documentation وبها شرح و يعض الامثلة ، يمكن عمل نسخ لها لمعرفة النتائج

```
> help(rep)
> ?rep
```

المساعدة فى R

R Documentation

rep {base}

Replicate Elements of Vectors and Lists

Description

rep replicates the values in x. It is a generic function, and the (internal) default method is described here.

rep.int and rep_len are faster simplified versions for two common cases. They are not generic.

Usage

```
rep(x, ...)

rep.int(x, times)

rep_len(x, length.out)
```

```
> ?seq # استفهام حول الامر seq
> seq(1, 9, by = 2)# من الأمثلة في R Documentation
[1] 1 3 5 7 9
> ?rep
```

English Introduction

R is a software language for carrying out complicated (and simple) statistical analyses. It includes routines for data summary and exploration, graphical presentation and data modelling. The aim of this document is to provide you with a basic fluency in the language. It is suggested that you work through this document at the computer, having started an R session. Type in all of the commands that are printed, and check that you understand how they operate. Then try the simple exercises at the end of each section.

When you work in R you create objects that are stored in the current workspace(sometimes called image). Each object created remains in the image unless you explicitly delete it. At the end of the session the workspace will be lost unless you save it. You can save the workspace at any time by clicking on the disc icon at the top of the control panel.

مقدمة

تعد لغة Rمن اللغات التي صعد نجمها حديثا وبشكل سريع بمجال البرمجة العلمية في قطاعي الإحصاء والمعلوماتية الحيوية ((bioinformaticsحيث باتت معتمدة على نطاق واسع في كثير من الجامعات ومراكز البحث العلمية، وأصبحنا نرى استخدامها والإشارة إليها في المقالات المنشورة بالمجلات العلمية المحكّمة يزداد بشكل طردي ومتسارع، هذا عدى عن حقيقة كونها لغة حرة مفتوحة المصدر يخضع توزيعها لترخيص GPLالشهير. كل ذلك أدى إلى تزايد ما هو متوفر ومتاح على الشابكة (الإنترنت) من مصادر لها على توزع طيف تلك المصادر، فهناك الكتب الإلكترونية والدروس التعليمية وحتى المناهج الأكاديمية والدورات التدريبية إضافة إلى البرامج الجاهزة والمكتوبة بلغة Rلتنفيذ هذه المهمة أو تلك، حتى أنها باتت تحظى ببعض الامتياز مقارنة بالعديد من العمالقة في قطاع البرمجة الرياضياتية العلمية والإحصائية مثل SASو SPSSخصوصا في مجال توافر الجديد من الطرق والخوارزميات الحديثة، حيث يقاد هذا التوجه في معظمه من طرف الجامعات ممثلة بطلاب الدراسات العليا يحفّزهم على ذلك سهولة بناء الإضافات لهذه اللغة، ويعتبر هذا الأسلوب رغم ما قد يشوبه من نقاط ضعف تتعلق بموثوقية وجودة وغزارة تلك الإضافات الجديدة، والتي تتبع خبرة ومهارة مطوريها وناشريها، لكنها تبقى في القطاع العلمي والأكاديمي أفضل كثيرا من البدائل التجارية التي يعيبها ارتفاع ثمنها من جهة، ومن جهة أخرى بطئ إضافة التحديثات التي تعكس تطور القطاعات العلمية المختلفة، حيث أنها عادة ما تتبع دورة تجارية تتحكم بها الشركات المنتجة.

و تستطيع أن تقوم بتحميل لغة Rمن الموقع الرسمي لها على الرابط
http://www.r-project.org

وعملية تنصيب هذه اللغة تخلو من التعقيدات وبانتهاءها يمكنك تشغيل سطر الأوامر الخاص بها من خلال النقر على أيقونة اللغة على سطح المكتبة، وكل مانكتبه تاليا يكون داخل سطر الأوامر هذا، علما أننا نستخدم في كل أمثلتنا إطار البيانات المدعو mtcarsوالذي يأتي محزوما مع اللغة بشكل إفتراضي، وللحصول على معلومات إضافية عن طبيعة محتوى هذه البيانات يمكنك كتابة الأمر التالي في سطر الأوامر ?mtcarsولاختصار طريقة الوصول إلى المعلومات ضمن إطار البيانات ننفذ الأمر (attach(mtcarsفنصبح قادرين على استخدام التسمية mpgبدلا من استخدام الطريقة المفصلة mtcars$mpgللدلالة على عدد الأميال المقطوعة بغالون البنزين الواحد.

Saved Files

Commands written in R are saved in memory throughout the session. You can scroll back to previous commands typed by using the `up' arrow key (and `down' to scroll back again). You can also `copy' and `paste' using standard windows editor techniques (for example, using the `copy' and `paste' dialog buttons). If at any point you want to save the transcript of your session, click on `File' and then `Save History', which will enable you to save a copy of the commands you have used for later use. As an alternative you might copy and paste commands manually into a notepad editor or something similar.
You finish an R session by typing

```
> q( )
```

Objects and Arithmetic

R stores information and operates on *objects*. The simplest objects are *scalars*, *vectors* and *matrices*. But there are many others: *lists* and *dataframes* for example. In advanced use of R it can also be useful to define new types of object, specifiic for particular application.

Some Operators : + , -, / , *

Examples

```
> x<-3 # تعريف لمتغير واعطائه قيمة
> y<-6 # تعريف لمتغير واعطائه قيمة

> x
[1] 3
> y
[1] 6
> x*y
[1] 18
> x/y
[1] 0.5
```

تعريف المتغير يستخدم الرمز « <-»
الاوارم باللون الاحمر ، والنتائج باللون الازرق

Examples

```
x<-6
y<-4
z<-x+y
z
```

[١] 10

At any time we can list the objects which we have created:

لاستعراض كافة المتغيرات التي تم تعريفها في الملف نستخدم الامر التالي

```
> ls()
```

[1] "x" "y" "z“

Objects can be removed from the current workspace with the rm

```
> rm(x,y)
```

Matrices “function c”

```
> x<-c(5,9)
> y<-c(1,0)
> z<-c(x,y)
```

```
> x<-c(1,2,3)
> y<-c(3,-4,5)
> x+y
[1]  4 -2  8
> x-y
[1] -2  6 -2
> x*y
[1]  3 -8 15
> x%*%y
     [,1]
[1,]   10
>
```

```
> x+4
[1] 5 6 7
> 2*y
[1]  6 -8 10
```

Matrices cbind ( ) and rbind ( ) , matrix ( ) functions

```
> x<-c(5,7,9)
> y<-c(6,3,4)
> z<-cbind(x,y)
> z
     x y
[1,] 5 6
[2,] 7 3
[3,] 9 4
```

```
> z<-rbind(x,y)
> z
  [,1] [,2] [,3]
x    5    7    9
y    6    3    4
```

```
> z<-matrix(c(5,7,9,6,3,4),nrow=3)
> z
     [,1] [,2]
[1,]    5    6
[2,]    7    3
[3,]    9    4
```

```
> z<-matrix(c(5,7,9,6,3,4),nrow=2)
> z
     [,1] [,2] [,3]
[1,]    5    9    3
[2,]    7    6    4
```

```
> z<-matrix(c(5,7,9,6,3,4),nc=3)
> z
     [,1] [,2] [,3]
[1,]    5    9    3
[2,]    7    6    4
> z<-matrix(c(5,7,9,6,3,4),nr=3)
> z
     [,1] [,2]
[1,]    5    6
[2,]    7    3
[3,]    9    4
```

```
> z+z
     [,1] [,2]
[1,]   10   12
[2,]   14    6
[3,]   18    8
```

Example

```
> x<-c(1,22,32,44,12,22,15,12,3,22)
> y<-c(1,22,32,44,12,22,15,12,3,22)
> z<-c(11,28,23,44,26,26,17,17,30,12)
> set<-cbind(x,y,z)
> set
      x  y  z
 [1,]  1  1 11
 [2,] 22 22 28
 [3,] 32 32 23
 [4,] 44 44 44
 [5,] 12 12 26
 [6,] 22 22 26
 [7,] 15 15 17
 [8,] 12 12 17
 [9,]  3  3 30
[10,] 22 22 12
```

تحويل البيانات إلى جدول فى R

```
> Name<-c("Ahmed","Belal","Camal","Dena")
> Age<-c(23,18,20,22)
> Gender<-c("M", "M", "M", "F")
> Table<-data.frame(Name, Age, Gender)
> Table
   Name Age Gender
1 Ahmed  23      M
2 Belal  18      M
3 Camal  20      M
4  Dena  22      F
```

استخدام الامر : ( ) data.frame

استخدام الامر : ( ) cbind

```
> Table2<-cbind(Name, Age, Gender)
> Table2
     Name     Age  Gender
[1,] "Ahmed" "23" "M"
[2,] "Belal" "18" "M"
[3,] "Camal" "20" "M"
[4,] "Dena"  "22" "F"
```

استخراج بعض البيانات من متجه في R

للمتجه Name السابق

```
> Name[2]
[1] "Belal"
> Name[2,4]
Error in Name[2, 4] : incorrect number of dimensions
> Name[c(2,4)]
[1] "Belal" "Dena"
```

لاحظ أنه للحصول على أكثر من عنصر فإننا نستخدم الرمز c للحصول على تراتيب معينة مراد اظهارها

```
> Name[c(-2,-4)]
[1] "Ahmed" "Camal"
```

اظهار محتوى المتجه Name ماعدا الثاني و الرابع

استيراد جدول من الحافظة باستخدام الامر :

```
<- read.delim('clipboard')
```

لو كان لدينا الجدول التالي في اكسل و من ثم قمنا بنسخه ليتم حفظه في الحافظة

	A	B	C
1	Name	Gender	Age
2	Ali	M	23
3	Ommar	M	25
4	Noor	F	22
5	Shahd	F	24
6			

باستخدام الامر read.delim('clipboard') نخبر R أن تستورد البيانات الموجودة في الحافظة كما يلي

```
> Table<-read.delim('clipboard')
> Table
   Name Gender Age
1   Ali      M  23
2 Ommar      M  25
3  Noor      F  22
4 Shahd      F  24
```

هنالك بعض المجموعات والتي تأتي محزومة مع اللغة بشكل إفتراضي، مثل trees, mtcars ، وللحصول على محتوى هذه المجموعات يمكن استخدام الاستدعاء attach(datasetname) أو لاستعراض محتواها ب كتلبة اسمها مباشرة

```
> mtcars
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C         17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE        16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL        17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC       15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
Fiat 128          32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic       30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Toyota Corolla    33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Toyota Corona     21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Dodge Challenger  15.5   8 318.0 150 2.76 3.520 16.87  0  0    3    2
AMC Javelin       15.2   8 304.0 150 3.15 3.435 17.30  0  0    3    2
Camaro Z28        13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Pontiac Firebird  19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2
Fiat X1-9         27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
Porsche 914-2     26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa      30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Ford Pantera L    15.8   8 351.0 264 4.22 3.170 14.50  0  1    5    4
Ferrari Dino      19.7   6 145.0 175 3.62 2.770 15.50  0  1    5    6
Maserati Bora     15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8
Volvo 142E        21.4   4 121.0 109 4.11 2.780 18.60  1  1    4    2
```

لمعرفة تركيب هذه البيانات فى mtcars نستخدم الامر str ()

```
> str(mtcars)
'data.frame':   32 obs. of  11 variables:
 $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
 $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
 $ disp: num  160 160 108 258 360 ...
 $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
 $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
 $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
 $ qsec: num  16.5 17 18.6 19.4 17 ...
 $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
 $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
 $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
 $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

استخراج بيانات معينة من مجموعة بيانات كبيرة

مثلا استخراج بيانات معينة من بيانات mtcars

```
> head(mtcars)#لاظهار أول ستة أسطر من البيانات
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
> tail(mtcars)#  لاظهار آخر ستة أرقام
                mpg cyl  disp  hp drat    wt qsec vs am gear carb
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.7  0  1    5    2
Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.9  1  1    5    2
Ford Pantera L 15.8   8 351.0 264 4.22 3.170 14.5  0  1    5    4
Ferrari Dino   19.7   6 145.0 175 3.62 2.770 15.5  0  1    5    6
Maserati Bora  15.0   8 301.0 335 3.54 3.570 14.6  0  1    5    8
Volvo 142E     21.4   4 121.0 109 4.11 2.780 18.6  1  1    4    2
>
```

"t " function to calculate a matrix transpose
"solve" to calculate the inverses

```
> t(z)
     [,1] [,2] [,3]
[1,]    5    7    9
[2,]    6    3    4
> solve(z)
Error in solve.default(z) : 'a' (3 x 2) must be square
> z<-matrix(c(5,7,9,6,3,4,3,4,-1),nr=3)
> z
     [,1] [,2] [,3]
[1,]    5    6    3
[2,]    7    3    4
[3,]    9    4   -1
> solve(z)
             [,1]       [,2]         [,3]
[1,] -0.114457831  0.1084337  0.090361446
[2,]  0.259036145 -0.1927711  0.006024096
[3,]  0.006024096  0.2048193 -0.162650602
```

Examples

```
> z
     [,1] [,2] [,3]
[1,]    5    6    3
[2,]    7    3    4
[3,]    9    4   -1
```

```
> z[1,1]
[1] 5
> z[2,2]
[1] 3
> z[3,2]
[1] 4
> z[c(2,3),2]
[1] 3 4
> z[,2]
[1] 6 3 4
> z[2,]
[1] 7 3 4
```

Exercises

1. Create in **R** the matrices

$$x = \begin{bmatrix} 3 & 2 \\ -1 & 1 \end{bmatrix}$$

and

$$y = \begin{bmatrix} 1 & 4 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

Calculate the following and check your answers in R:

(a) `2*x`
(b) `x*x`
(c) `x%*%x`
(d) `x%*%y`
(e) `t(y)`
(f) `solve(x)`

2. With `x` and `y` as above, calculate the effect of the following subscript operations and check your answers in R.

(a) `x[1,]`
(b) `x[2,]`
(c) `x[,2]`
(d) `y[1,2]`
(e) `y[,2:3]`

Generating Sequences (**seq()**) function

- `> x<-1:10`

```
> seq(1,9,by=2)
[1] 1 3 5 7 9

> seq(8,20,length=6)
[1]  8.0 10.4 12.8 15.2 17.6 20.0

> x<-seq(1,10)
```

Repeating "rep()" function"

```
> rep("328 stat", 5)
[1] "328 stat" "328 stat" "328 stat" "328 stat" "328 stat"
```

```
> rep("328 stat", 3)
[1] "328 stat" "328 stat" "328 stat"
> rep(1:3,6)
 [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
> > rep(1:3,rep(6,3))
Error: unexpected '>' in ">"
> rep(1:3,c(6,6,6))
 [1] 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3
>
```

Exercises

1. Define

```
> x<-c(4,2,6)
> y<-c(1,0,-1)
```

Decide what the result will be of the following:

(a) `length(x)`

(b) `sum(x)`

(c) `sum(x^2)`

(d) `x+y`

(e) `x*y`

(f) `x-2`

(g) `x^2`

Use **R** to check your answers.

2. Decide what the following sequences are and use **R** to check your answers:

   (a) `7:11`

   (b) `seq(2,9)`

   (c) `seq(4,10,by=2)`

   (d) `seq(3,30,length=10)`

   (e) `seq(6,-4,by=-2)`

3. Determine what the result will be of the following **R** expressions, and then use **R** to check you are right:

   (a) `rep(2,4)`

   (b) `rep(c(1,2),4)`

   (c) `rep(c(1,2),c(4,4))`

   (d) `rep(1:4,4)`

   (e) `rep(1:4,rep(3,4))`

4. Use the `rep` function to define simply the following vectors in R.

   (a) 6,6,6,6,6,6

   (b) 5,8,5,8,5,8,5,8

   (c) 5,5,5,5,8,8,8,8