- **R Programming Language Statements**

The R programming language, like any other programming language, has specific programming statements used to solve a problem and simulate the computer to obtain results. Below are the programming statements used in the R language, along with the syntax and outline for each statement:

1. If statements

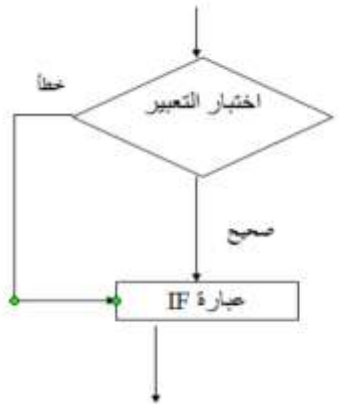
Decision-making is an important part of programming, and this can be achieved in programming using the R language. The following is an explanation of if statements:

- **if statement if**

An if statement is a logical expression used to check a condition and execute certai n commands when the condition is met. The if statement is structured as follows:

```
if (شرط) {
 # أوامر تُنفّذ إذا كان الشرط      TRUE
}
```

If the condition is true, the statement is executed, but if it is false, nothing happens. The condition can be a logical or numerical vector, but only the first element is t aken into consideration. The following is a diagram illustrating the structure of that statement:



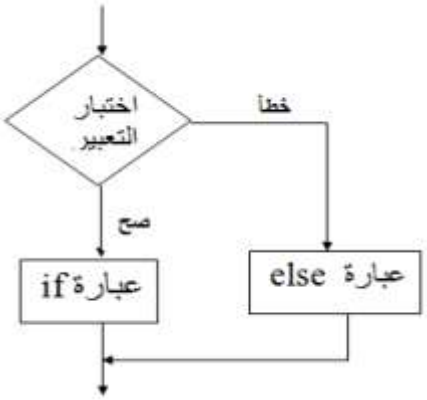
مخطط لعبارة if

Example :

```
x=5
if(x>0)
+ if(x>0) {print("positive number")}
[1] "positive number"
if(1==0){
+ print(1)
+ }
```

- **if … else**

An if statement followed by an else statement is used to execute conditional statements, i.e., to execute a specific code if a certain condition is met, and to e xecute another code if the condition is not met. The general syntax for an if…else s tatement is as follows:



Example :

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

Example :

```
> x=10
> if(x>5){print("5 من اكبر العدد")}
[1] "5 من اكبر العدد"
> if(x>5){print("5 من اكبر العدد")}else{print("العدد صغر او يس
5 اوين")}
[1] "5 من اكبر العدد"
```

- **Second-nested if...else statement**

A number of If…else statements can be nested, and the general form is as follows:

if (test expression1 اختبار التعبير) { statement 1 عبارة
}else if (test expression2){ statement 2
} else if (test expression3){ statement 3
}else
statement 4

Example :

```
x=85 # لشرطين متداخلين
> if(x>=90){print("امتياز")}else if(x>=80){print("جيد جدا")}e
lse if(x>=70){print("جيد")}else if(x>=60){print("متوسط")}else
if(x>=50){print("متوسط")}else{print("راسب")}
[1] "جيد جدا"
```

Example :

```
x=0
> if(x<0){print("Negative number")}
if(x>0){print("positive Number")}
> if(x<0){print("Negative number")}else if(x>0){print("
positive Number")}else print("zero")
[1] "zero"
```

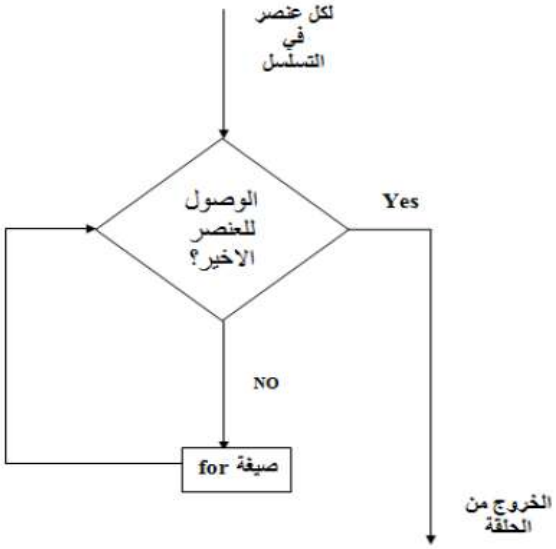
- **Third: Loops**

Loops in RStudio are used to repeat the execution of a set of managers multiple times rather than manually writing them each time. The most common loops in R are the for and while statements.

1. for loop

It is used to repeat the vector in R, and to know the remaining number of times. T he general formula is as follows:

For (variable in sequence){
Statement
}



Example:

```
#  To calculate the number of even numbers in vector x
 x=c(2,5,3,9,8,11,6)
 count=0
 for (variable in x) if (variable%%2==0) count=count+1
   #   مقارنة منطقية صح و خطأ(%%)
    + }
 print(count)
[1] 3
```

For example, there are 7 numbers in the vector x. The counting is used to calculate t he remaining even number. Divide by 2. If the result is 0, then the number is an eve n number, and thus the result becomes 3 numbers.

Example :

```
# this code calculates square of the number 1 to 10 ,cre
ate a vector zero 10 time
> v1=rep(0,10)
> v1=rep(0,10) # repeating zero 10 times
> for (i in 1:10) {v1[i]=i^2
+
+ }
> v1
 [1]   1   4   9  16  25  36  49  64  81 100
```

Example :

```
# To calculate the square of each number in vector v1
> v1=rep(0,10)
> v1=rep(0,10) # repeating zero 10 times
> for (i in 1:10) {v1[i]=i^2
+
+ }
> v1
 [1]   1   4   9  16  25  36  49  64  81 100
```

Example:

```
# this code computes the sum and the mean of a vector
> v=c(2,3,7,12,9) # Input the vector data
> n=length(v) # compute the number of observation in v
> sum.v=0  # will be used to save the sum of v
> for (i in 1:n) {sum.v=sum.v+v[i] # beginning of the
for-loop and calculating the sum step by step
+
+ } # end the loop
> sum.v
```

```
[1] 33
> n
[1] 5
> mean.v=sum.v/n
> mean.v
[1] 6.6
```

Sometimes we need to use more than one **for-loop** to get the desired results .
For example, if we want to read the elements of a matrix ,then we have to have tw o index variable , and therefor ,two nested **for-loops** .

Example :

```
# Define the two matrix
m1=matrix(1:9,nrow = 3,ncol = 3)
> m2=matrix(27:35,nrow = 3,ncol = 3)
> m2
     [,1] [,2] [,3]
[1,]   27   30   33
[2,]   28   31   34
[3,]   29   32   35
 m2=matrix(27:35,nrow = 3,ncol = 3,byrow = T)
 m2
     [,1] [,2] [,3]
[1,]   27   28   29
[2,]   30   31   32
[3,]   33   34   35
# Defining the needed matrices to store the results
m3=matrix(0,nrow = 3,ncol = 3) # for m1+m2
m3
     [,1] [,2] [,3]
[1,]    0    0    0
[2,]    0    0    0
[3,]    0    0    0
m4=matrix(0,nrow = 3,ncol = 3) # for m2-m1
m4
     [,1] [,2] [,3]
[1,]    0    0    0
```

```
[2,]    0    0    0
[3,]    0    0    0
for (i in 1:nrow(3)) {    # i will be used for rows
+
+ }
+ for (j in 1:ncol(3)) {  # j will be used for columns
+ m3[i,j]=m1[i,j]+m2[i,j]

+ m4[i,j]=m2[i,j]-m1[i,j]
+ }
+ }
 m3
     [,1] [,2] [,3]
[1,]   28   32   36
[2,]   32   36   40
[3,]   36   40   44
 m4
     [,1] [,2] [,3]
[1,]   26   24   22
[2,]   28   26   24
[3,]   30   28   26
```

Sometimes we need to use some conditional statements to evaluate certain expressions inside a **for-loop**. As shown in the example below:

Example :

```
# find the square root
# randomly generate 10 numbers from -100 to 100
set.seed(1) # تثبيت البيانات
v1=sample(-100:100,size=0)# Setting random number seed (10 عينة عشوائية بحجم)
v1
n=length(v1)
sqr.v1=rep(0,times=n) # to store the sruare roots
for (i in 1:n) {
if (v1[i]>=0){
 sqr.v1[i] =sqrt(v1[i])
}else{
   sqr.v1[i]=NA # NA stands for not Applicable
```

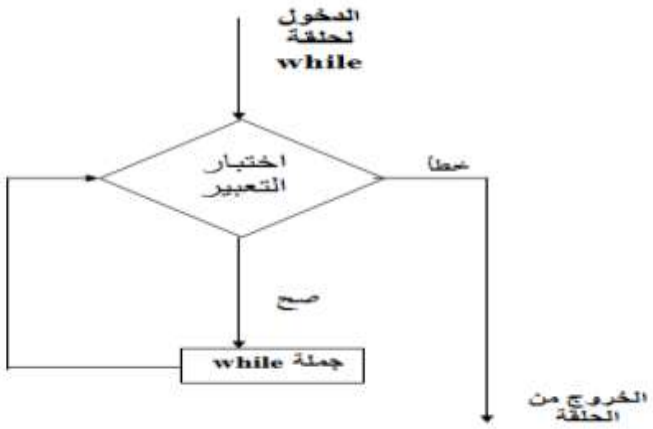
```
 }
}
# round numbers to 2 decimal place
sqr.v1=round(sqr.v1,digits = 2)
cbind(v1,sqr.v1)# display the results
```

2.While

The While statement in R is used to execute a loop as long as the specified cond ition is true (TRUE), meaning that the commands inside the loop are executed repe atedly until the condition becomes false (FALSE). The syntax of the While stateme nt is as follows:

```
While (test-expression)(الشرط) {     # الأوامر التي سيتم تنفيذها طالما الشرط TRUE
Statement
}
```

يتم تقييم (test-expression) فاذا كانت النتيجة صحيحة يتم تنفيذ التعليمات داخل الحلقة ، ويستم تنفيذ العبارة الى حتى تصبح النتيجة خطأ ، وبالتالي يتم انهاء الحلقة . والمخطط لعبارة While تكون كالاتي :



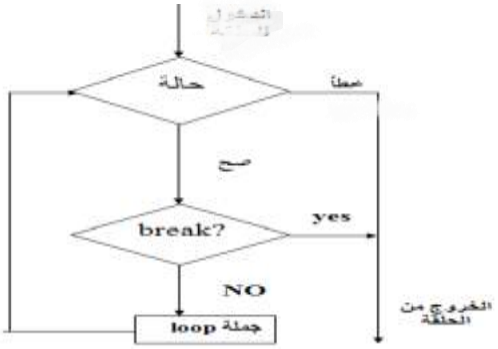
Another way to Control **for-loop** in R is by using break , Using break inside a loop allows use to stop the loop if certain condition is satisfied . For example , break the loop if find we read a negative **number** . Notice that break usually requires an if- st atement before breaking the loop .

Example :

```
> x<-1  #   x=1 تبدأ الحلقة بالقيمة
> while (x<=5) {
+   print(x)
+   x=x+1
+ }
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

3. Break Statement

The Break statement in RStudio is used to stop a loop immediately before it completes the remaining iterations, even if the original stopping condition for the loop is not met. In other words, the Break statement exits the loop completely and moves on to the next code, as shown in the diagram below:



Example:

```
> x=1:5
> for (v1 in x) {
+   if(v1==3){
+     break
+   }
+   print(v1)
+   }
[1] 1
[1] 2
```

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

Example :

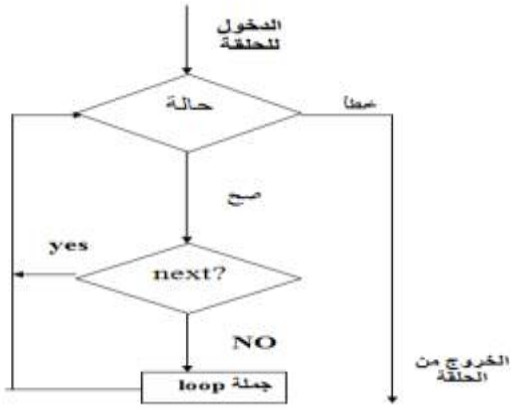
```
> x<-1
> while (x<=10) {
+   print
+   if(x==5){
+     break  #  x5 الحلقة عندما تصل يخرج من
+   }
+   x=x+1
+   print(x)
+ }
[1] 2
[1] 3
[1] 4
[1] 5
```

Example 9 :

```
> # Example with for loop
> for (i in 1:10) {
+   if(i==4){
+     break   #  4 توقف عند الرقم
+   }
+  print(i)
+ }
[1] 1
[1] 2
[1] 3
```

4. The next statement

The next statement is used inside loops (for, while) to skip the current iteration of t he loop without ending it, i.e. to move directly to the next iteration of the loop. This is the opposite of the break statement, which ends the loop completely, as shown in the diagram below:



Example:

```
> x=1:5
> for (v1 in x) {
+   if(v1==3){
+     next
+   }
+   print(v1)
+ }
[1] 1
[1] 2
[1] 4
[1] 5
```

Example:

```
> for (i in 1:10) {
+   if (i%%2==0){
+     next  # تخط الاعداد الزوجية
+   }
+   print(i)
+ }
[1] 1
[1] 3
[1] 5
[1] 7
[1] 9
```

In the above example when (i) is an even number, next is executed thus skipping printing and going directly to the next number.