

# Introduction to R Software

## Swayam Prabha

### Lecture 17

## Repeat Loop and Sequences of Numbers

Shalabh

Department of Mathematics and Statistics

Indian Institute of Technology Kanpur

Slides can be downloaded from  
<http://home.iitk.ac.in/~shalab/sp>



# Control structures in R :

## Loops

Repetitive commands are executed by loops

- **for loop**
- **while loop**
- **repeat loop**

## 1. The `for` loop

If the number of repetitions is known in advance (, a `for()` loop can be used.

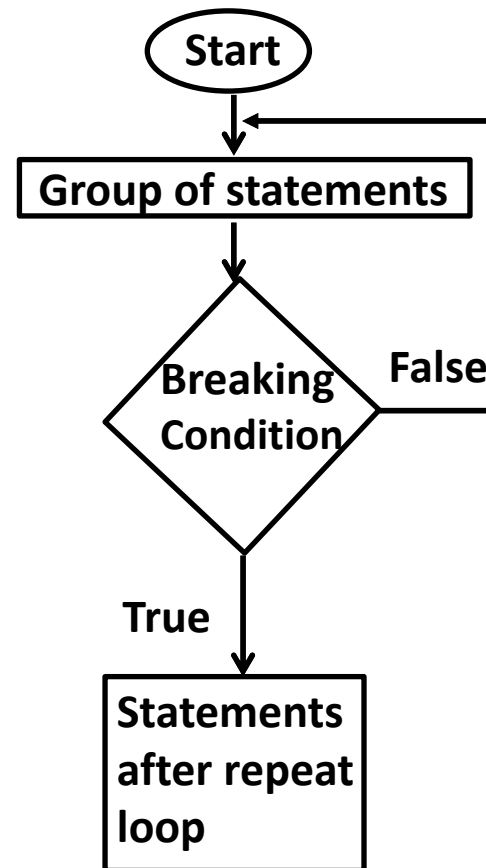
## 2. The `while` loop

If the number of loops is not known in before, one can use a `while()` loop.

The programmer itself has to be careful that the counting variable `i` within the loop is incremented. Otherwise an infinite loop occurs.

### 3. The repeat loop

The repeat loop doesn't test any condition — in contrast to the `while()` loop — *before entering* the loop and also not during the execution of the loop.



### 3. The `repeat` loop

Again, the programmer is responsible that the loop terminates after the appropriate number of iterations. For this the `break` command can be used.

#### Syntax

```
repeat{ commands to be executed }
```

### 3. The repeat loop

#### Example 1

```
> i <- 1
```

```
> repeat{  
  + print( i^2 )  
  + i <- i+2  
  + if ( i > 12 ) break  
  + }
```

```
[1] 1
```

```
[1] 9
```

```
[1] 25
```

```
[1] 49
```

```
[1] 81
```

```
[1] 121
```

### 3. The repeat loop

#### Example 1

```
R Console  
  
> i <- 1  
> repeat{  
+ print( i^2 )  
+ i <- i+2  
+ if ( i > 12 ) break  
+ }  
[1] 1  
[1] 9  
[1] 25  
[1] 49  
[1] 81  
[1] 121  
,
```

### 3. The repeat loop

#### Example 2

Additionally, the command `next` is available, to return to the beginning of the loop (to return to the first command in the loop).

```
> i <- 1
> repeat{
+ i <- i+1
+ if (i < 10) next
+ print(i^2)
+ if (i >= 14) break
+ }
[1] 100
[1] 121
[1] 144
[1] 169
[1] 196
```

```
R Console
> i <- 1
> repeat{
+ i <- i+1
+ if (i < 10) next
+ print(i^2)
+ if (i >= 14) break
+ }
[1] 100
[1] 121
[1] 144
[1] 169
[1] 196
```



## Sequences

A sequence is a set of related numbers, events, movements, or items that follow each other in a particular order.

The regular sequences can be generated in R.

Syntax

`seq()`

```
seq(from = 1, to = 1, by = ((to -  
from)/(length.out - 1)), length.out = NULL,  
along.with = NULL, ...)
```

Help for `seq`

```
> help("seq")
```

# Sequences

127.0.0.1:13069/library/base/html/seq.html

R Documenta

seq {base}

## Sequence Generation

### Description

Generate regular sequences. `seq` is a standard generic with a default method. `seq.int` is a primitive which can be much faster but has a few restrictions. `seq_along` and `seq_len` are very fast primitives for two common cases.

### Usage

```
seq(...)
```

```
## Default S3 method:
```

```
seq(from = 1, to = 1, by = ((to - from)/(length.out - 1)),  
    length.out = NULL, along.with = NULL, ...)
```

```
seq.int(from, to, by, length.out, along.with, ...)
```

```
seq_along(along.with)
```

```
seq_len(length.out)
```

### Arguments

`...` arguments passed to or from methods.

`from`, `to` the starting and (maximal) end values of the sequence. Of length 1 unless just `from` is supplied as an unnamed argument.



# Sequences

□ The default increment is +1 or -1

```
> seq(from=2, to=6)
```

```
[1] 2 3 4 5 6
```

```
> seq(from=4, to=6)
```

```
[1] 4 5 6
```

```
> seq(from=-6, to=6)
```

```
[1] -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
```

# Sequences

R Console

```
> seq(from=2, to=6)
[1] 2 3 4 5 6
>
> seq(from=4, to=6)
[1] 4 5 6
>
> seq(from=-6, to=6)
[1] -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6
>
```

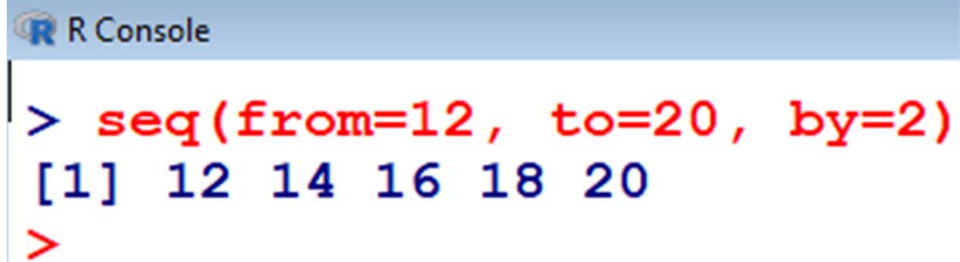
## Sequences

□ Sequence with constant increment:

Generate a sequence from 12 to 20 with an increment of 2 units

```
> seq(from=12, to=20, by=2)
```

```
[1] 12 14 16 18 20
```



```
R Console  
> seq(from=12, to=20, by=2)  
[1] 12 14 16 18 20  
>
```

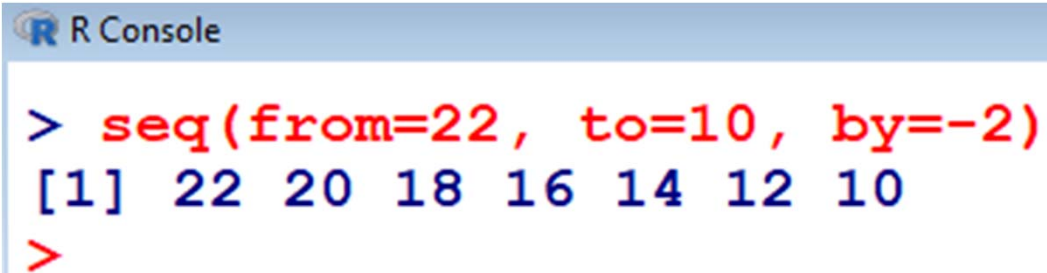
# Sequences

□ Sequence with constant increment:

Generate a sequence from 22 to 10 with a decrement of 2 units

```
> seq(from=22, to=10, by=-2)
```

```
[1] 22 20 18 16 14 12 10
```



```
R Console  
> seq(from=22, to=10, by=-2)  
[1] 22 20 18 16 14 12 10  
>
```

## Sequences

□ Downstream sequence with constant increment:

Generate a sequence from 2 to -1 with a decrement of 0.5 units

```
> seq(from=2, to=-1, by=-0.5)
```

```
[1] 2.0 1.5 1.0 0.5 0.0 -0.5 -1.0
```

R Console

```
> seq(from=2, to=-1, by=-0.5)
```

```
[1] 2.0 1.5 1.0 0.5 0.0 -0.5 -1.0
```

```
>
```

# Sequences

- ❑ Sequences with a predefined length with default increment +1

```
> seq(to=15, length=10)
```

```
[1] 6 7 8 9 10 11 12 13 14 15
```

```
R Console  
> seq(to=15, length=10)  
[1] 6 7 8 9 10 11 12 13 14 15  
>
```



# Sequences

- Sequences with a predefined length with default increment +1

```
> seq(from=15, length=10)
```

```
[1] 15 16 17 18 19 20 21 22 23 24
```

R Console

```
> seq(from=15, length=10)
```

```
[1] 15 16 17 18 19 20 21 22 23 24
```

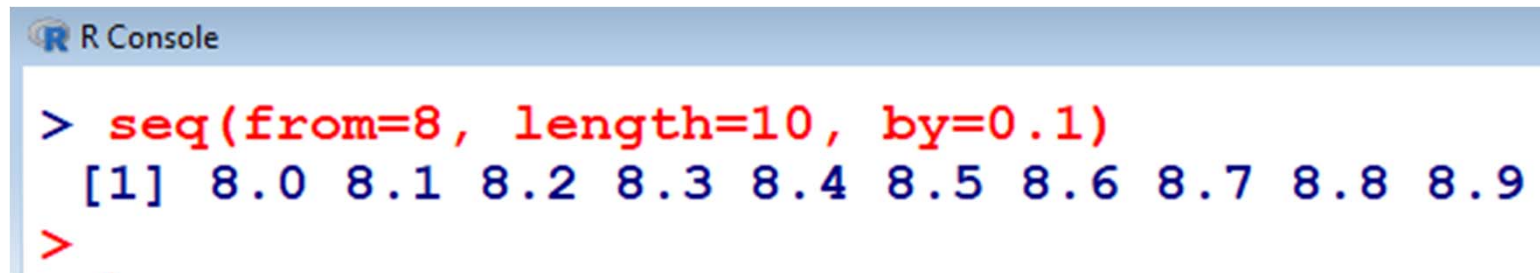
```
>
```

## Sequences

- ❑ Sequences with a predefined length with constant fractional increment

```
> seq(from=8, length=10, by=0.1)
```

```
[1] 8.0 8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9
```



```
R Console  
> seq(from=8, length=10, by=0.1)  
[1] 8.0 8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8 8.9  
>
```

## Sequences

- Sequences with a predefined length with constant decrement

```
> seq(from=15, length=10, by=-2)
[1] 15 13 11  9  7  5  3  1 -1 -3
```

```
R Console
> seq(from=15, length=10, by=-2)
[1] 15 13 11  9  7  5  3  1 -1 -3
>
```

## Sequences

- Sequences with a predefined length with constant fractional decrement

```
> seq(from=15, length=5, by=-.2)
```

```
[1] 15.0 14.8 14.6 14.4 14.2
```

R Console

```
> seq(from=15, length=5, by=-.2)
```

```
[1] 15.0 14.8 14.6 14.4 14.2
```

```
>
```

# Sequences

## Sequences with a predefined variable and constant increment

```
> x<-3
```

```
> seq(1, x, x/10)
```

```
[1] 1.0 1.3 1.6 1.9 2.2 2.5 2.8
```

```
> x<-60
```

```
> seq(0, x, x/10)
```

```
[1] 0 6 12 18 24 30 36 42 48 54 60
```

# Sequences

R Console

```
> x<-3
> seq(1, x, x/10)
[1] 1.0 1.3 1.6 1.9 2.2 2.5 2.8
>
> x<-60
> seq(0, x, x/10)
[1] 0 6 12 18 24 30 36 42 48 54 60
>
```

## Sequences


```
> seq(12)
```

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

is the same as

```
> seq(1:12)
```

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

 R Console

```
> seq(12)
```

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

```
>
```

```
> seq(1:12)
```

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

```
>
```

## Sequences

### □ Assignment of an index-vector

```
> x <- c(9,8,7,6,5)
> ind <- seq(along=x)
> ind
[1] 1 2 3 4 5
```

```
R Console
> x <- c(9,8,7,6,5)
> ind <- seq(along=x)
> ind
[1] 1 2 3 4 5
>
```

### Accessing a value in the vector through index vector

### □ Accessing an element of an index-vector

```
> x[ ind[3] ]
[1] 7
```

```
R Console
> x[ ind[3] ]
[1] 7
>
```