

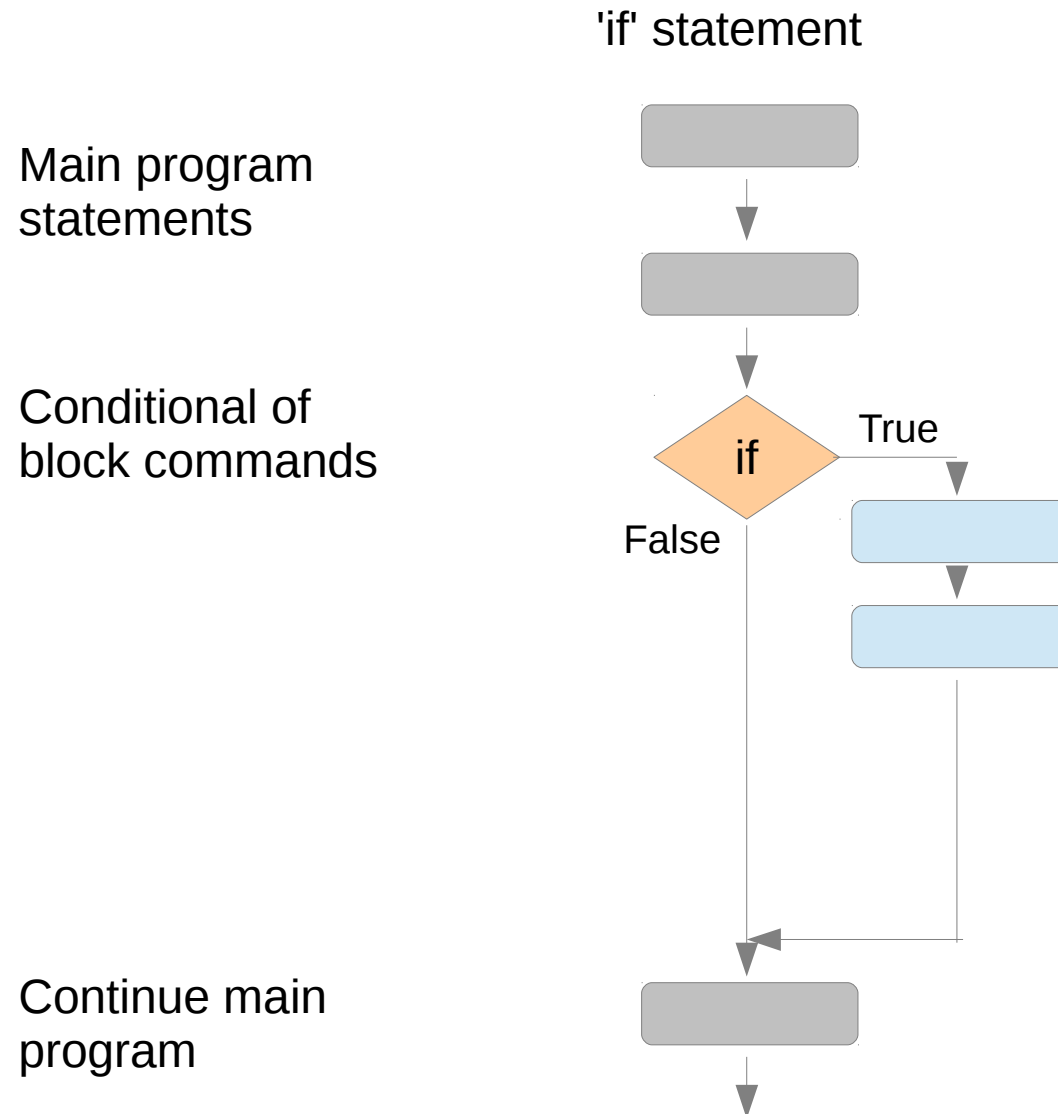
# *The R Book*

## **Chapter 2: Essentials of the R Language**

### **Session 10**

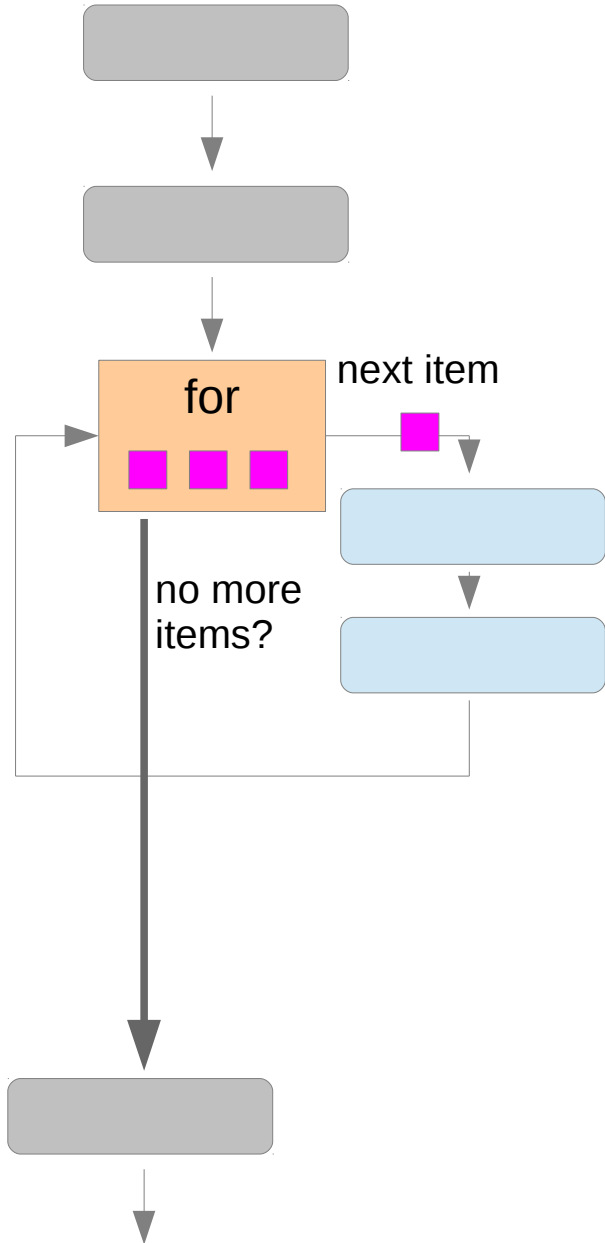
# Flow control: Decisions with the 'if' statement

## Conditional decision making

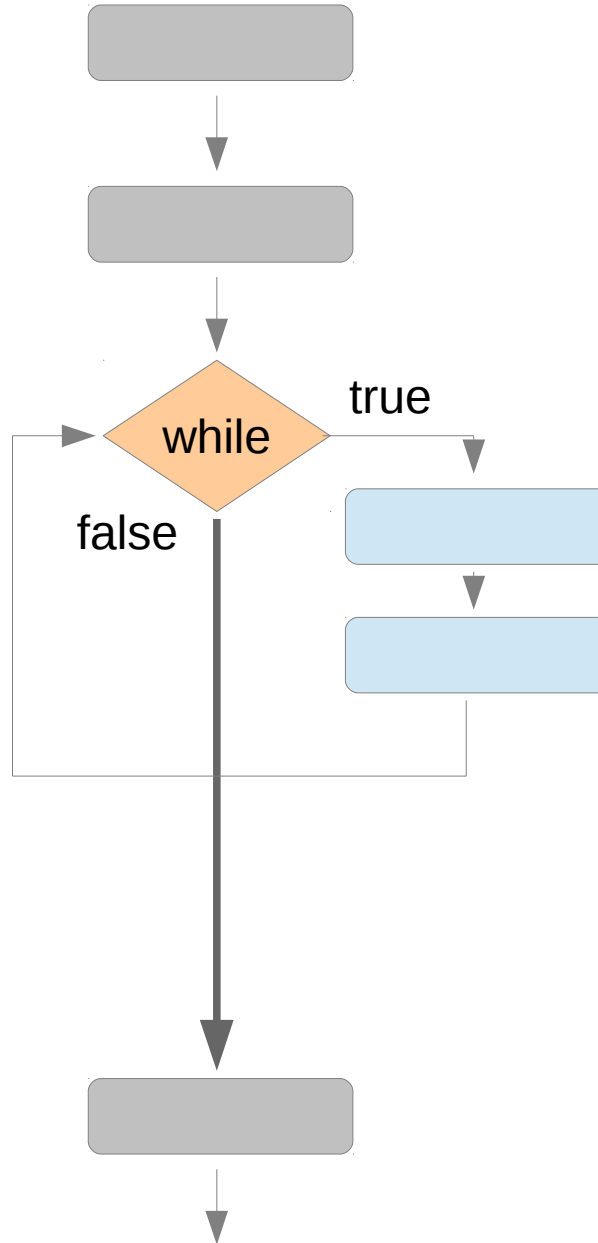


# Flow control: Looping with 'for' and 'while'

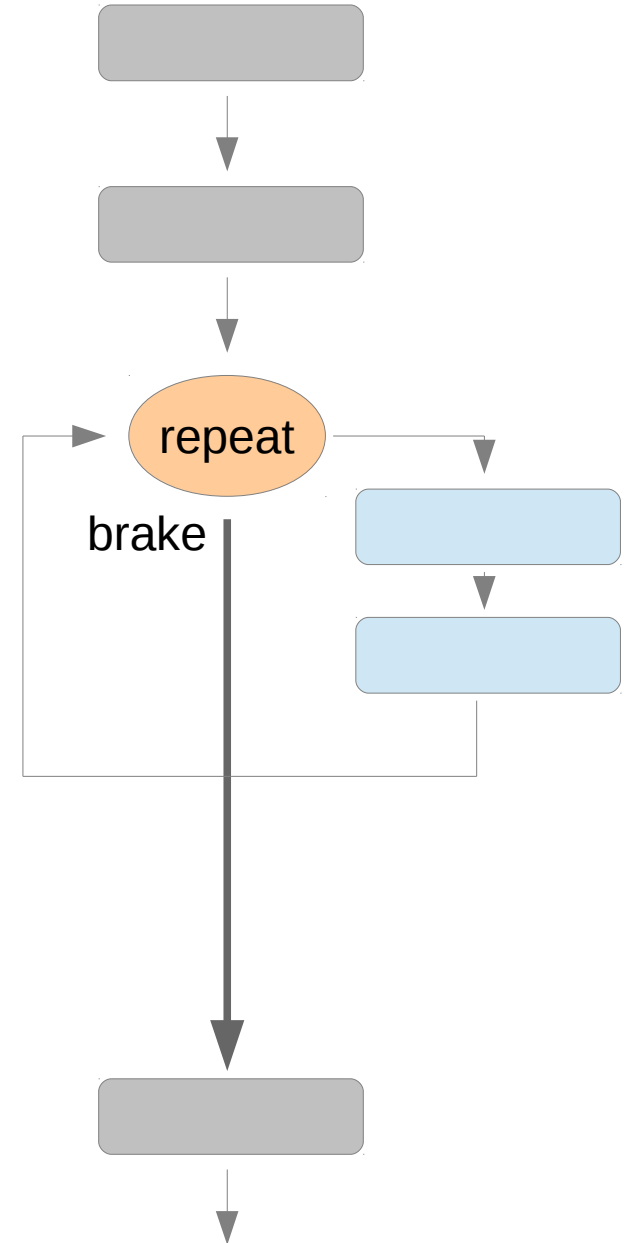
'for' loop



'while' loop



'repeat' loop



# Loops and Repeats

## For-loop

```
> for (i in 1:5) print(i^2)
```

```
[1] 1  
[1] 4  
[1] 9  
[1] 16  
[1] 25
```

```
> phrase<- "the quick brown fox jumps over the lazy dog"
```

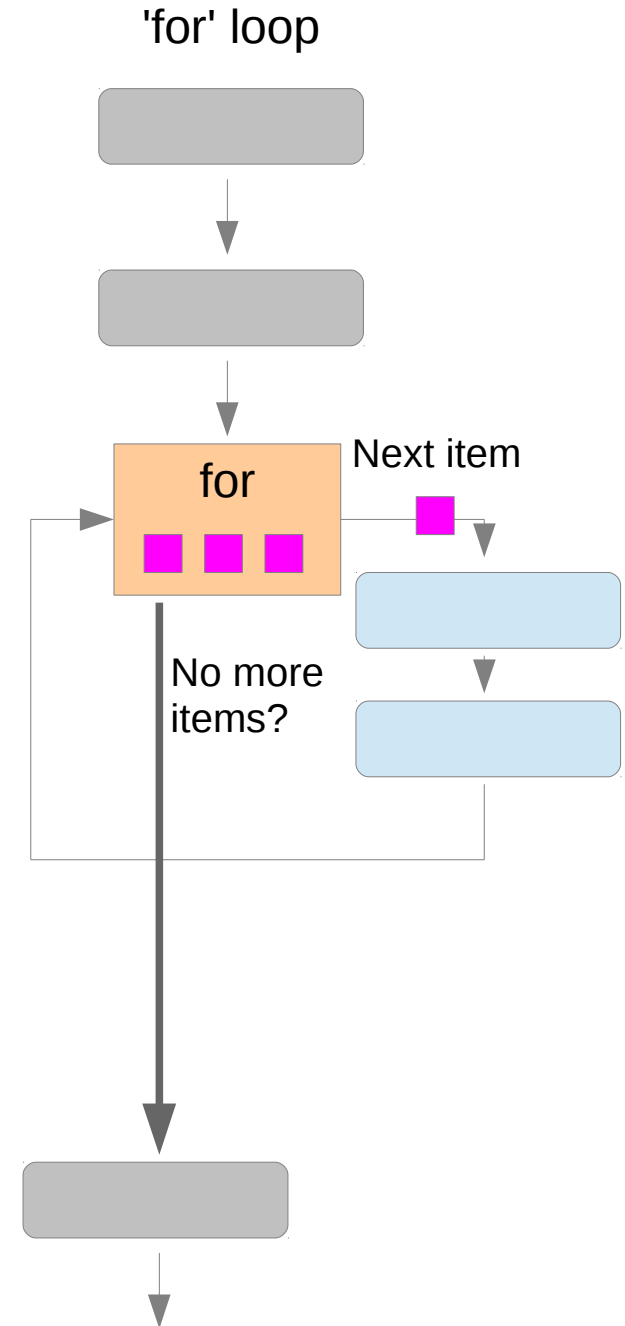
```
> q<-character(20) vector with 20 "blank" entries
```

```
> for (i in 1:20) q[i]<- substr(phrase, 1, i)
```

```
> q
```

```
...
```

**Puts Substrings into a character vector**



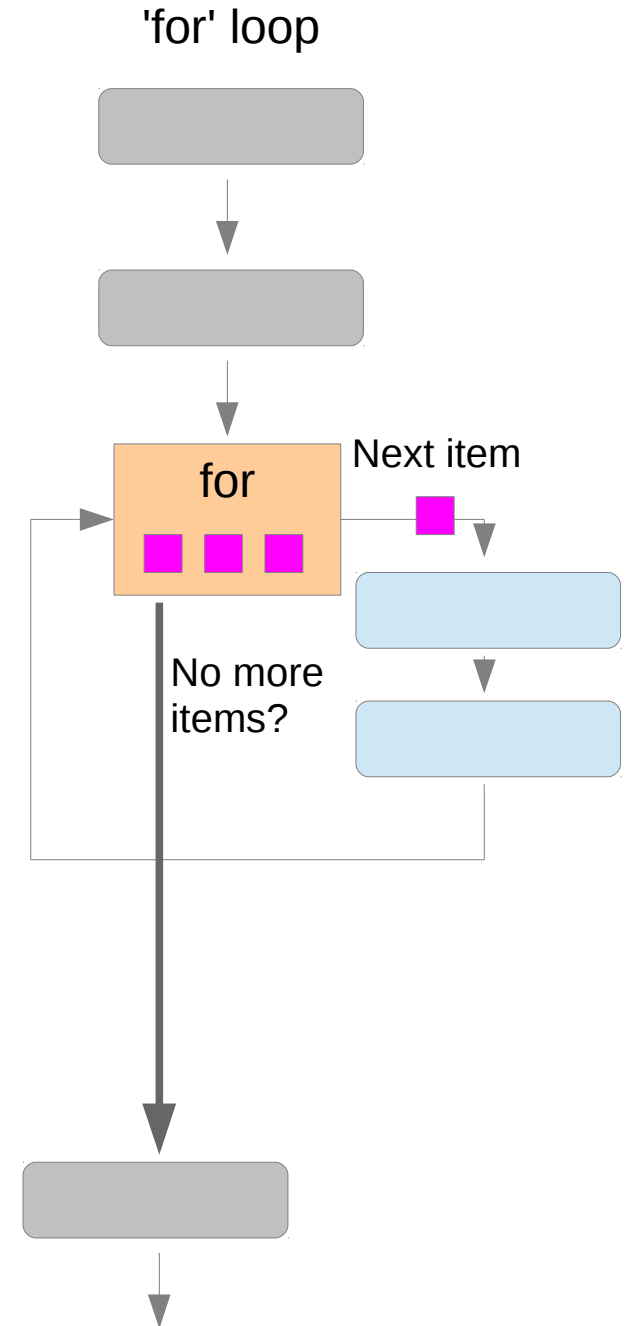
# Loops and Repeats

## For-loop

```
> j<-k<-0  
> for (i in 1:5) {  
+ j<-j+1  
+ k<-k+i*j  
+ print(i+j+k) }  
[1] 3  
[1] 9  
[1] 20  
[1] 38  
[1] 65
```



```
> j<-k<-0; for (i in 1:5) {j<-j+1; k<-k+i*j; print(i+j+k)}
```



# Loops and Repeats

## Example : Factorial function

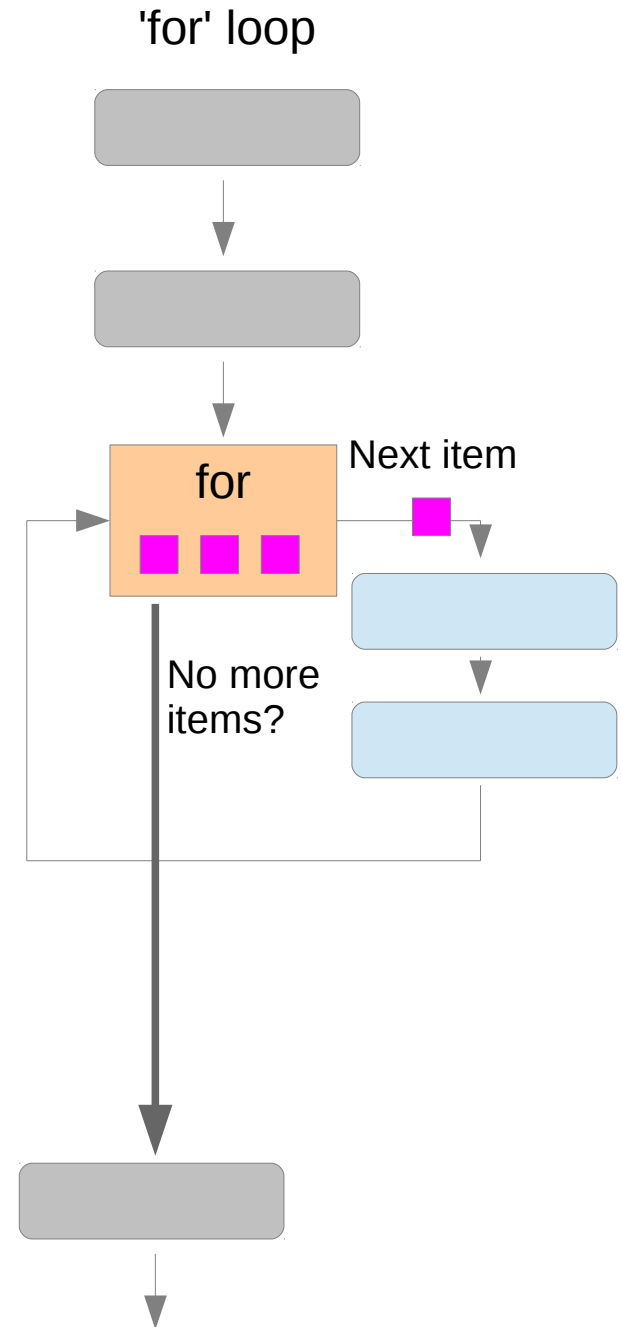
$$x! = x \times (x - 1) \times (x - 2) \times (x - 3) \dots \times 2 \times 1$$

### For:

```
> fac1 <- function(x) {  
+ f <- 1  
+ if (x<2) return (1)  
+ for (i in 2:x) {  
+ f <- f*i }  
+ f }
```

```
> fac1(5)  
[1] 120
```

```
> sapply(0:5, fac1)  
[1] 1 1 2 6 24 120
```



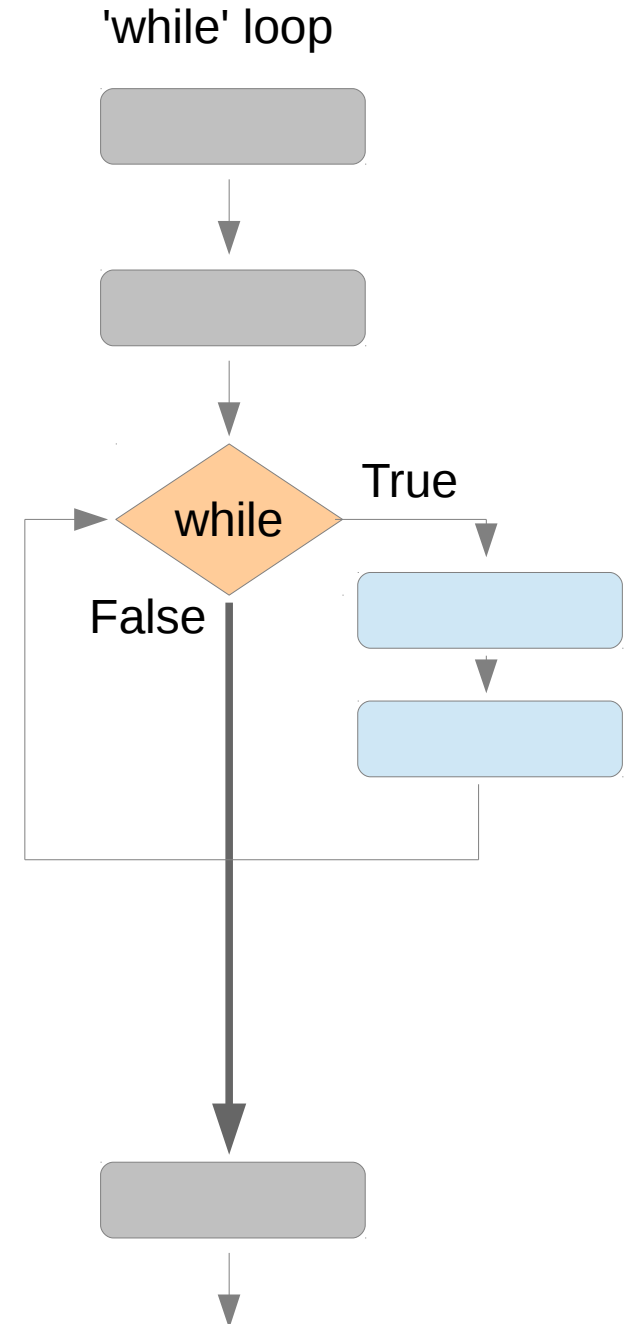
# Loops and Repeats

$$x! = x \times (x - 1) \times (x - 2) \times (x - 3) \dots \times 2 \times 1$$

while:

```
> fac2<-function(x) {  
+ f <- 1  
+ t <- x  
+ while(t>1) {  
+ f <- f*t  
+ t <- t-1 }  
+ return(f) }
```

```
> sapply(0:5, fac2)  
[1] 1 1 2 6 24 120
```



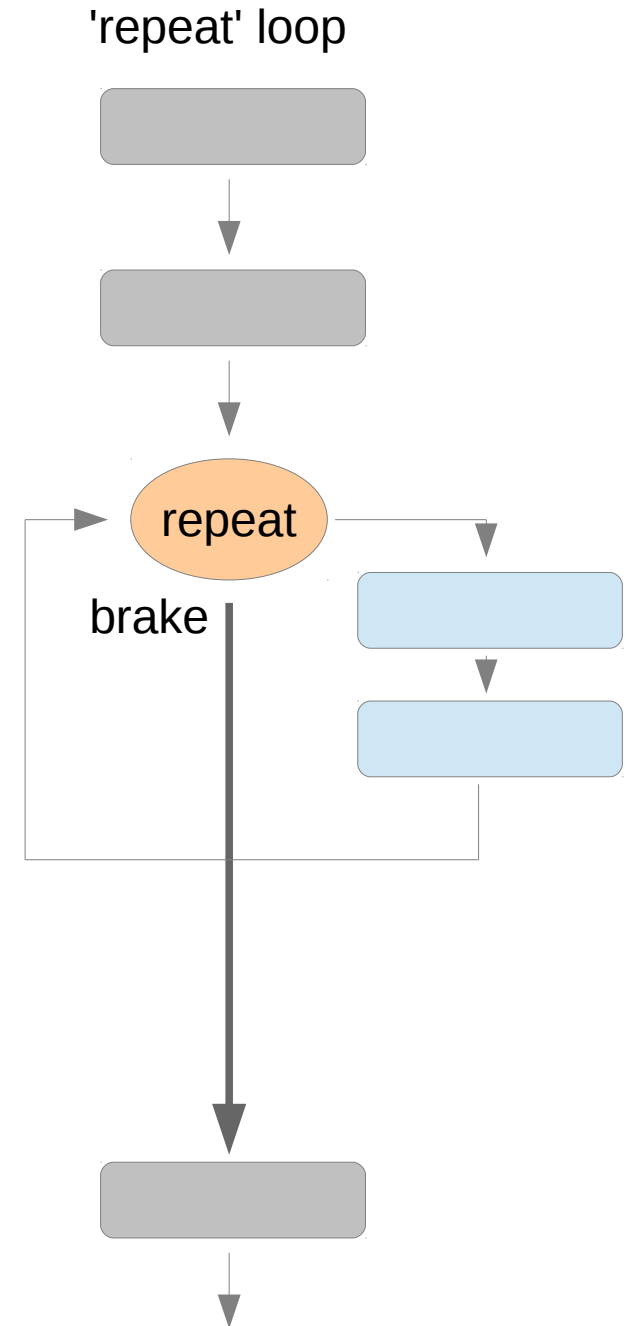
# Loops and Repeats

$$x! = x \times (x - 1) \times (x - 2) \times (x - 3) \dots \times 2 \times 1$$

repeat:

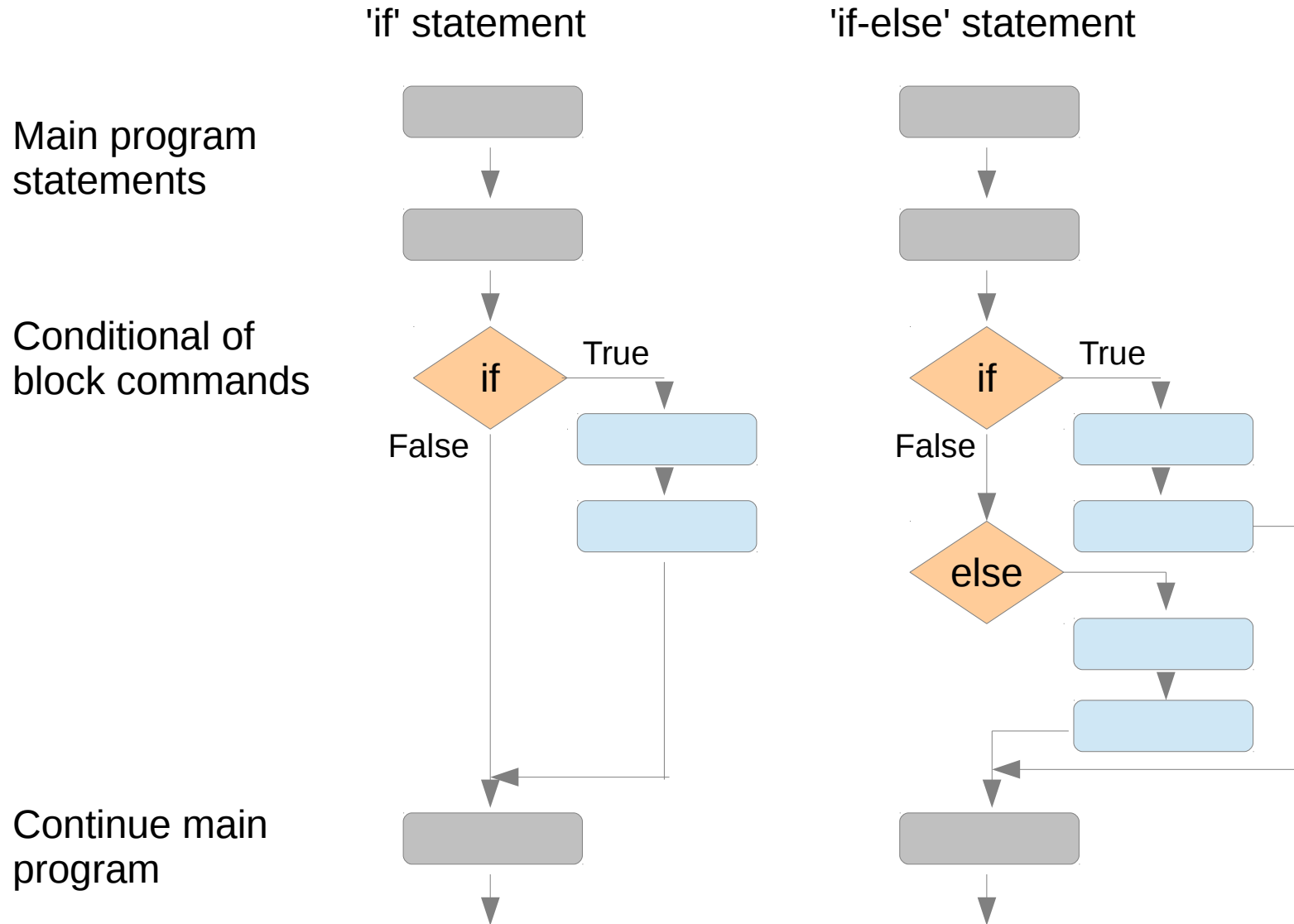
```
> fac3<-function(x) {  
+ f <- 1  
+ t <- x  
+ repeat {  
+ if (t<2) break  
+ f <- f*t  
+ t <- t-1 }  
+ return(f) }
```

```
> sapply(0:5, fac3)  
[1] 1 1 2 6 24 120
```





# The ifelse function



# The ifelse function

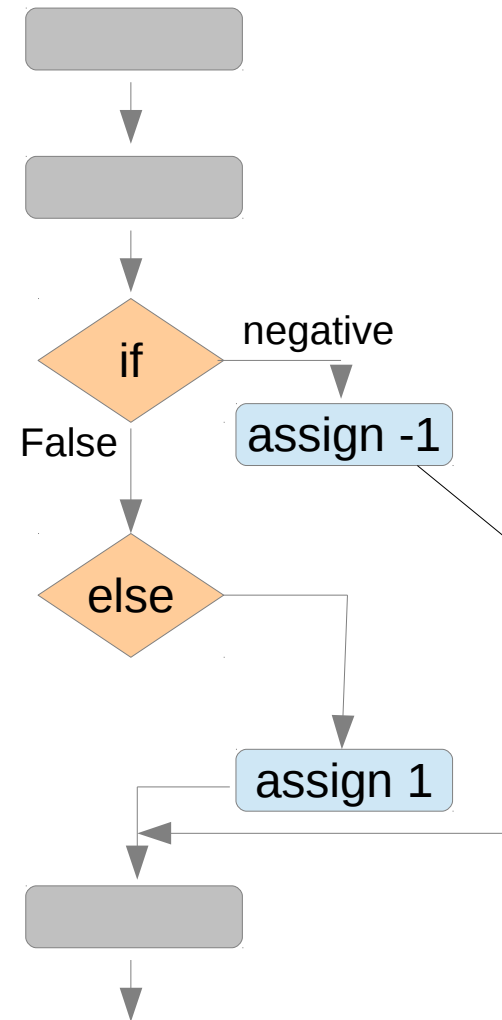
Example :

replace negative values of  $y$  by  $-1$  and positive values and zero by  $+1$ :

```
>y <- rnorm(10,0,20)
```

```
> z <- ifelse (y < 0, -1, 1)
```

if do else

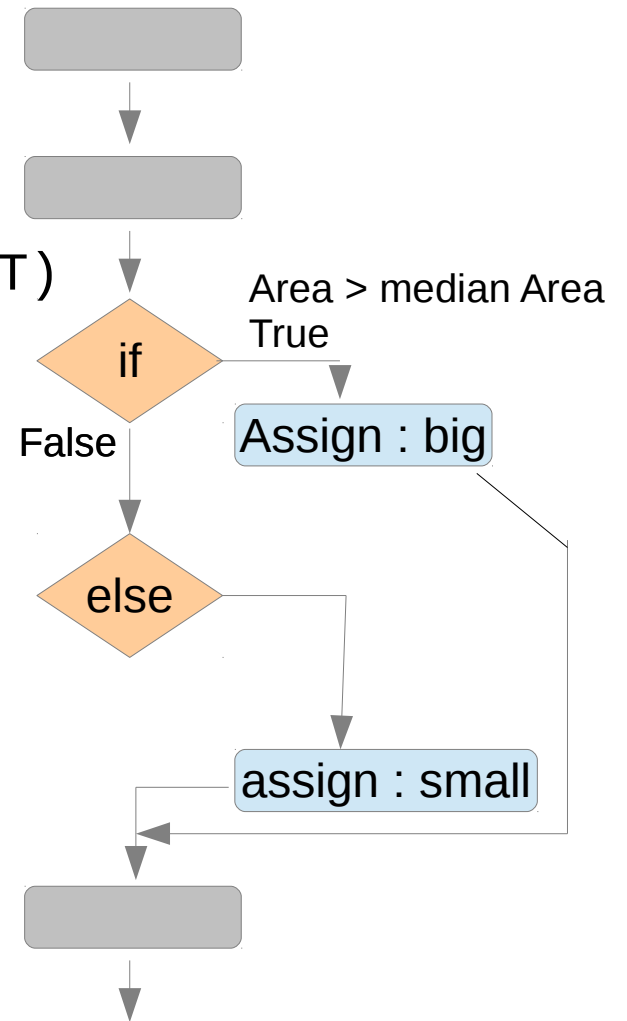


# The ifelse function

Example :

```
> data<-read.table("//worms.txt",header=T)  
> attach(data)  
> data  
...
```

```
> ifelse(Area>median(Area), "big", "small")  
[1] "big" "big" "small" "small" "big" "big" "big" "small" "small"  
[10] "small" "small" "big" "big" "small" "big" "big" "small" "big"  
[19] "small" "small"
```



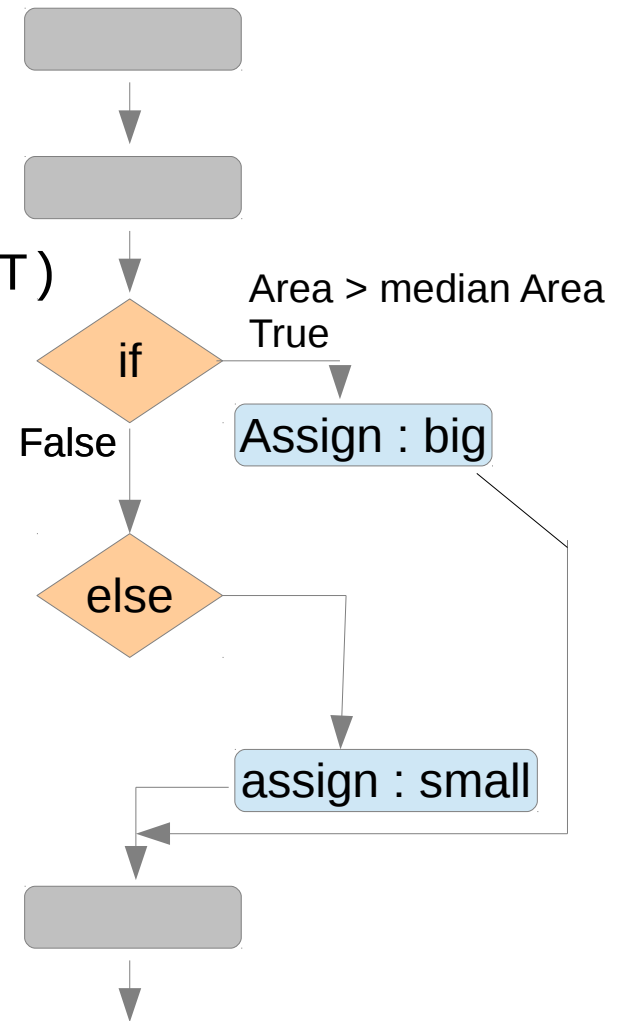
# The ifelse function

Example :

```
> data<-read.table("//worms.txt",header=T)  
> attach(data)  
> data  
...
```

```
> ifelse(Area>median(Area), "big", "small")  
[1] "big" "big" "small" "small" "big" "big" "big" "small" "small"  
[10] "small" "small" "big" "big" "small" "big" "big" "small" "big"  
[19] "small" "small"
```

Question : How do you add this data to the table ???



# The ifelse function

Example :

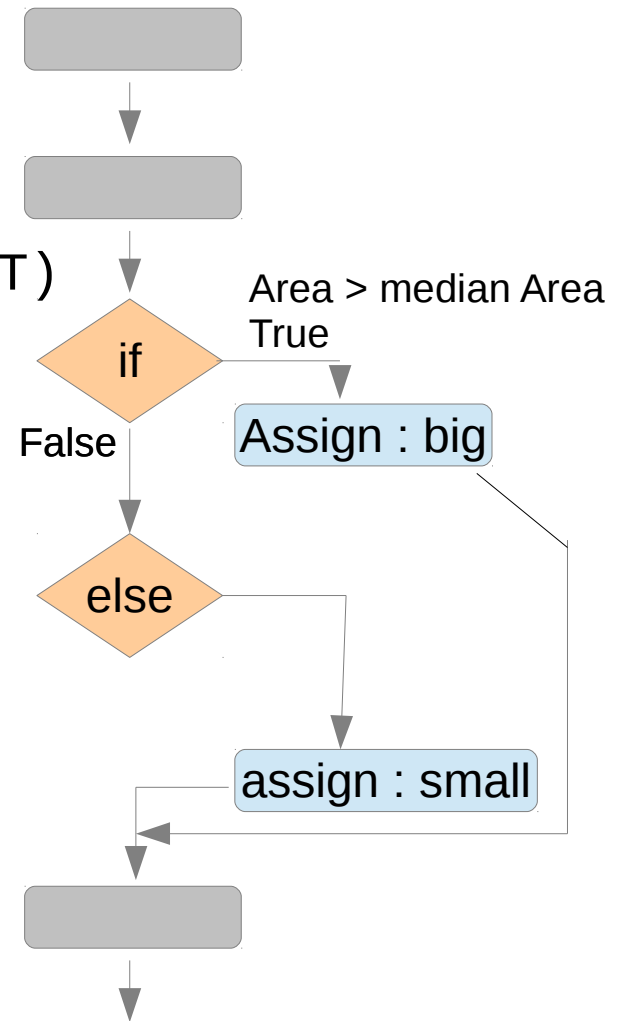
```
> data<-read.table("//worms.txt",header=T)
> attach(data)
> data
...
```

```
> ifelse(Area>median(Area), "big", "small")
[1] "big" "big" "small" "small" "big" "big" "big" "small" "small"
[10] "small" "small" "big" "big" "small" "big" "big" "small" "big"
[19] "small" "small"
```

Question : How do you add this data to the table ???

```
> data2<-cbind(data, size=ifelse(Area>median(Area), "big", "small"))
> data2
```

...



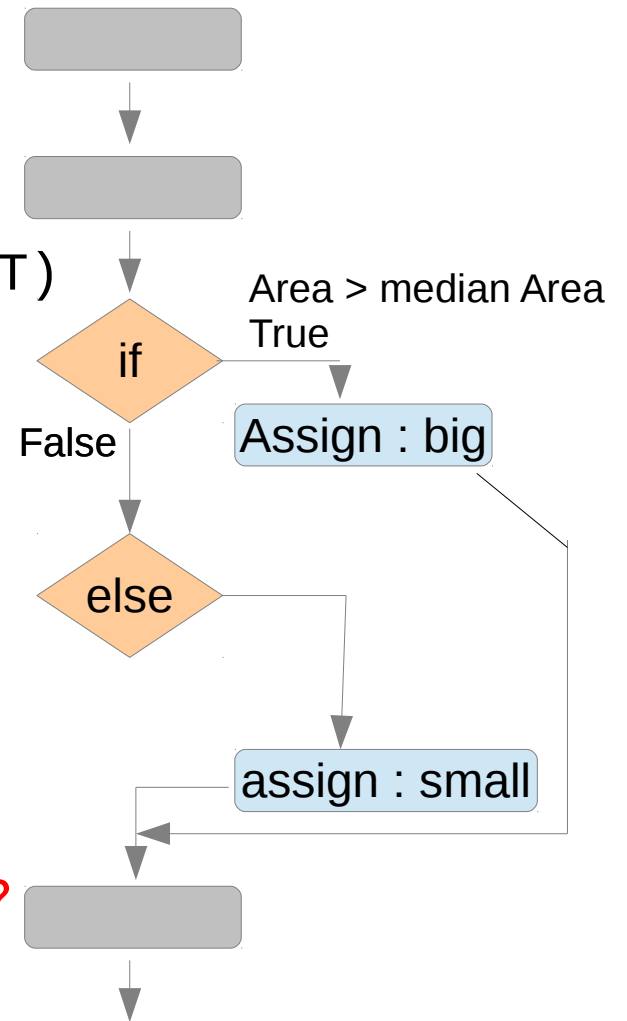
# The ifelse function

Example :

```
> data<-read.table("//worms.txt",header=T)  
> attach(data)  
> data  
...
```

```
> ifelse(Area>median(Area), "big", "small")  
[1] "big" "big" "small" "small" "big" "big" "big" "small" "small"  
[10] "small" "small" "big" "big" "small" "big" "big" "small" "big"  
[19] "small" "small"
```

**Question :**  
What is the mean soil pH in big vs. small areas ?

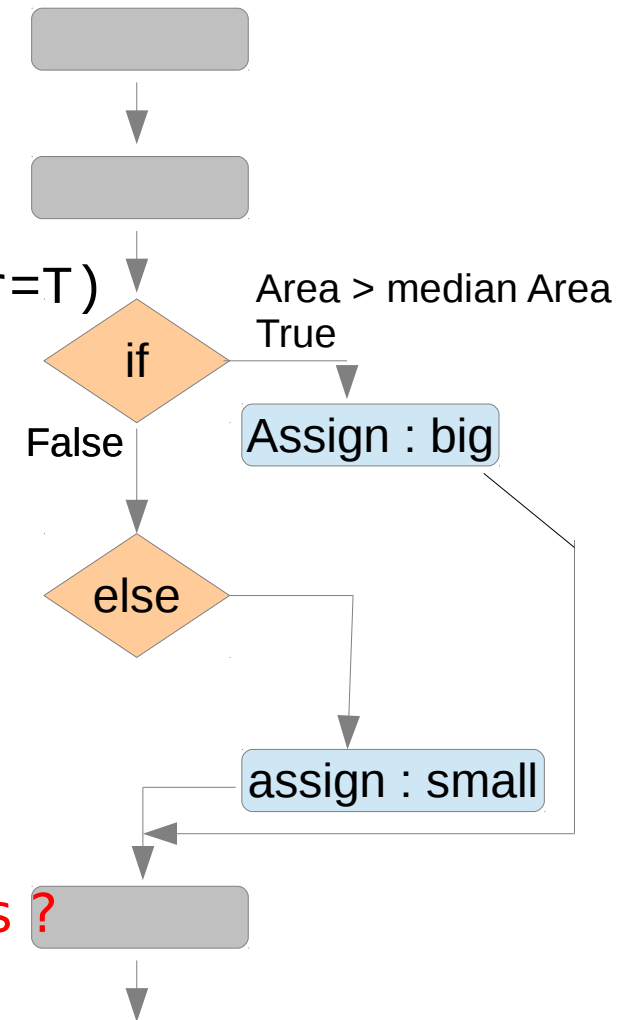


# The ifelse function

Example :

```
> data<-read.table("//worms.txt",header=T)
> attach(data)
> data
..
```

```
> ifelse(Area>median(Area), "big", "small")
[1] "big" "big" "small" "small" "big" "big" "big" "small" "small"
[10] "small" "small" "big" "big" "small" "big" "big" "small" "big"
[19] "small" "small"
```



**Question :**  
What is the mean soil pH in big vs. small areas ?

```
> tapply(data2$Soil.pH,data2$size,mean)
```

Or :

```
> attach(data2) #here you have a problem ...
> detach(data2)
> detach(data)
> attach(data2)
> tapply(Soil.pH,size,mean)
```

# The ifelse function

## Override entries in vectors

```
> y<-log(rpois(20,1.5))
```

Poisson distribution    n    lambda

```
> y
[1]      -Inf 0.0000000 0.0000000 0.0000000 0.6931472 0.0000000 0.0000000
[8] 0.0000000      -Inf 0.0000000 1.0986123 1.0986123 0.0000000 1.0986123
[15]      -Inf 0.0000000      -Inf      -Inf 0.6931472      -Inf
```

```
> rpois(20,1.5)
[1] 0 1 1 1 2 4 0 1 2 0 0 2 1 1 4 0 2 4 1 3
```

```
> y2<-ifelse(y<0,NA,y)
```

```
> y2
[1]      NA 0.0000000 0.0000000 0.0000000 0.6931472 0.0000000 0.0000000
[8] 0.0000000      NA 0.0000000 1.0986123 1.0986123 0.0000000 1.0986123
[15]      NA 0.0000000      NA      NA 0.6931472      NA
```



# The slowness of loops

```
>x<-runif(100000000) #random number from 0-1
```

```
> system.time(max(x)) #to see how long a process lasts
```

utilisateur	ystème	écoulé
0.040	0.001	0.040

CPU time charged for  
the execution of user  
instructions

CPU time charged  
for execution

Total time

CPU : central processing unit

```
> pc<-proc.time()  
> cmax<-x[1]  
> for (i in 2:100000000) {  
+ if(x[i]>cmax) cmax<-x[i] }  
> proc.time()-pc
```

utilisateur	ystème	écoulé
9.358	0.067	13.008

# The switch Function

... to do different things in different circumstances

```
> central<-function(y, measure) {  
+   switch(measure,  
+     Mean = mean(y),  
+     Geometric = exp(mean(log(y))),  
+     Harmonic = 1/mean(1/y),  
+     Median = median(y),  
+     stop("Measure not included")) }
```

```
> cen<-rnorm(100,10,2)  
> central(cen,"Median")  
> central(cen,4)  
> central(cen,1)  
> central(cen,2)  
> central(cen,5)
```

```
> print(switch(ceiling(runif(1,min=0,max=5))), "rain",  
"clouds", "storm", "fog", "sun"))
```