

# R:If, else and loops

Presenter: Georgiana Onicescu

January 19, 2012

# Contents

- Vectors
- Matrices
- If else statements
- For loops
- Leaving the loop: stop, break, next commands
- Other loops: while and repeat
- Avoiding the loops: apply function

- A one column list of elements (a scalar is also a vector) - a one dimensional array.
- Methods of creating vectors:

```
vec.a = c(2,3,4,5,6,7,8,9,10,11,12,13,14,15)
```

```
vec.a = c(2:15)
```

```
vec.a = as.numeric(c("-.01"," 2.5 ","C"))
```

```
vec.a = seq(2,15,by=1)
```

```
vec.a = c(a, b, c, d)
```

- A combination of vectors.
- Can combine by column or by row.
- Methods of creating matrices:

```
vec1 = c(2,3,4,5,6,7,8,9,10,11,12,13,14,15)
vec2 = c(as.numeric(22:35))
vec3 = seq(42,55,by=1)
vecb = c("aa", "bb", "cc", "dd", "ee", "ff", "gg",
"hh","ii", "jj", "kk", "ll","mm","nn")
matr1 = cbind(vec1, vec2, vec3)
matr2 = rbind(vec1, vec2, vec3)
matr3 = cbind(vec1, vec2, vec3, vecb)
matr4 = matrix(vecb,7,2)
matr5=matrix(vec1,ncol=2,byrow=TRUE)
```

# If else statements

- **if ( statement1 )  
statement2  
else  
statement3**
- *If statements can be nested:*  
**if ( statement1 )  
statement2  
else if ( statement3 )  
statement4  
else if ( statement5 )  
statement6  
else  
statement8**

## If else statements. Example

- Example: Calculating median of a single sample

```
x=c(2,7,1,8,9)
```

```
med=function(x) { odd=length(x)%%2
if (odd==0) {
  med=(sort(x)[length(x)/2]+sort(x)[1+length(x)/2])/2
}
else med=sort(x)[ceiling(length(x)/2)]
}
m=med(x)
```

```
m
```

```
7
```

*Note: ceiling takes a single numeric argument and returns the smallest integer not less than the corresponding elements of the argument.*

# Ifelse statements

- Syntax: `ifelse(test, truevalue, falsevalue)`
- `ifelse` operates on vectors therefore avoiding loops
- Example:

```
y=log(c(3,0.5,2,4))  
ifelse(y<0,NA,y)
```

# For loop

```
for(i in 1:n){  
  execute commands  
}
```

- Example: Calculating factorials for an integer x:

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

```
fac=function(x){  
  f=1  
  if (x<2) {f=1}  
  else {for (i in 2:x) f=f*i  
  }  
  return (f)  
}
```

# For loop

- The for loop can iterate any sequence. Examples:

```
for (i in 1:2) print(i)
```

1

2

```
for (i in (1:4)-2) print(i)
```

-1

0

1

2

```
library(MASS)
```

```
data(phones)
```

```
for(var in names(phones)) print(var)
```

"year"

"calls"

```
for(func in c(sin,cos)) print(func(pi))
```

1.224606e-16

-1

# Leaving the loop:stop

- stop on condition and print error message:

**a = 1:10**

**b = NULL**

**for (i in seq(along=a))**

**{ if (a[i]<5 || a[i]>8) { b = c(b,a[i]) } else {**

**stop("values need to be <5") } }**

*Error: values of a need to be <5*

**b**

**1 2 3 4**

## Leaving the loop:break

- break:stop on condition without printing the error message

```
rm(b)
```

```
a = 1:10
```

```
b = NULL
```

```
for (i in seq(along=a))  
{ if (a[i]<5 || a[i]>8) { b = c(b,a[i])} else  
{break("values of a need to be <5")}}
```

```
b
```

```
1 2 3 4
```

## Leaving the loop:`next`

- `next`

```
rm(b)
```

```
a = 1:10
```

```
b = NULL
```

```
for (i in seq(along=a)) { if (a[i]<5 || a[i]>8) { b =  
c(b,a[i])} else {next}}
```

```
b
```

```
1 2 3 4 9 10
```

## Other loops: While

- When using while, we need to set up an indicator variable and change its value within each iteration.
- Example: Calculation of the factorial

x=4

f=1

t=x

while (t > 1){

f=f\*t

t=t-1 }

f

24

# Other loops:Repeat

- Repeat: We must have a logical escape clause that leads to a break command

x=4

f=1

t=x

```
repeat {
  if (t < 2) break
  f=f*t
  t=t-1 }
f
24
```

# Avoiding the loops:apply

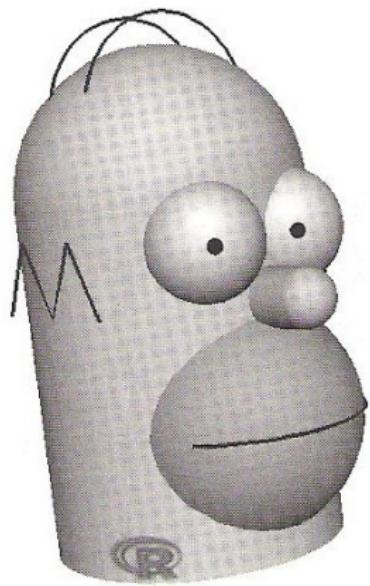
- Syntax: `apply(X, MARGIN, FUN, ARGs)`
- `X` is an array, matrix or `data.frame`; `MARGIN`: 1 for rows, 2 for columns, `c(1,2)` for both; `FUN`: one or more functions; `ARGs`: possible arguments for function
- Examples:

```
matrix = matrix(c(1:10, 11:20), nrow = 10, ncol = 2)
apply(matrix, 1, mean)
apply(matrix, 2, mean)
apply(matrix, 1:2, function(x) x*2)
```

# References

- Michael J. Crawley. The R book. Wiley, 2007
- Johns Hopkins BST 140.776 Statistical Computing:  
<http://www.biostat.jhsph.edu/bcaffo/statcomp/files/ingo.R.2.notes.pdf>
- <http://manuals.bioinformatics.ucr.edu/home/programming-in-r>

# QUESTIONS?



Source:Paul Murrell.R graphics, Second edition.