Introduction to R programming for data science – day 4

Dr. Francesca Finotello
Medical University of Innsbruck, Austria

Loops and control structures

Loops

Loops are used to repeat the same code multiple times. They are constructed using *reserved* words.

In R, there are three loop structures:

```
repeat {
  # Chunk of code to be repeated...
}
```

```
while (condition) {
  # Chunk of code to be repeated...
}
```

```
for (variable in sequence) {
  # Chunk of code to be repeated...
}
```


The *for* loop

```
mysum <- 0
for (i in seq(1,10)) {
  mysum <- mysum + i
}
mysum
```

```
## [1] 55
```

```
days <- c("Monday", "Tuesday", "Wednesday")
for (i in 1:length(days)) {
  cat("Day ", i, " of the week: ", days[i], "\n", sep="")
}
```

```
## Day 1 of the week: Monday
## Day 2 of the week: Tuesday
## Day 3 of the week: Wednesday
```

The *while* loop

```
mysum <- 0
while (mysum < 10) {
  mysum <- mysum + 1

}
mysum
```

```
## [1] 10
```

Beware of infinite loops!

```
while (mysum < 100) {
  mysum <- mysum - 1

}
```


The *repeat* loop

```
x <- 1
repeat {

  print(x)
  x <- x + 1
  if (x == 6) {
    break
  }

}
```

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

The *if-else* statement

The if/else statement executes a block of code if a specified condition is TRUE. If the condition is FALSE, another block of code is executed.

```
temperature <- 39 #°C
fever_thresh <- 37.2

if (temperature > fever_thresh) {
  cat("Ouch, you have a fever. Stay in bed!\n")

}  else {
  cat("Don't panic! Your body temperature is normal.\n")

}
```

```
## Ouch, you have a fever. Stay in bed!
```


The *else if* statement

The if/else statement can be used to consider more than two conditions.

```
glycemia <- 100 # mg/dL
hypo_thresh <- 70
hyper_thresh <- 130

if (glycemia < hypo_thresh) {
  cat("Hypoglycemia\n")
}  else  if (glycemia > hyper_thresh) {
  cat("Hyperglycemia\n")
} else {
  cat("Normoglycemia\n")
}
```

```
## Normoglycemia
```

Other control statements: *break*

break can be used to interrupt a loop

```
x <- c(0, 1, 22, 100, 5, 8, 90, 6, 100)
wantedNum <- 100 # Number we are looking for
for (i in 1:length(x)) {
  if (x[i] == wantedNum) {
    cat(wantedNum, " found after ", i, " iterations.\n", sep="")
    break
  }
}
```

```
## 100 found after 4 iterations.
```

```
i
```

```
## [1] 4
```


Other control statements: *next*

next can be used to skip a loop iteration

```
x <- c(0, 1, 22, -9, 5, 8)
sumPosX <- 0 # Sum of all positive numbers in x
for (i in 1:length(x)) {

  if (x[i] > 0) {
    sumPosX <- sumPosX + x[i]

  } else {
    next
  }
}
sumPosX
```

```
## [1] 36
```