Another type of data structure: a list.

```
>l <- list("first",3,T,10:5)
```

>1

>1

List elements may be of different mode. They may be even lists ("recursive" object)

```
>l<-list(a = 1:3,
    b = c("tre","de","ge"),
    thirdElement = list(1:3, (1:3)>2, 1:15)
    )
```

```
>Lst<-list(Name="Charles",
```

```
Wife="Emma",
```

```
no.children=5,
```

```
children.age=c(2,5,7,8,12),
```

```
favor.food=list(fruit="apple",drink="wine"))
```

Have a look on resulting object

Check what is the lenght of list Lst?

What are the names of list elements?

To access a list element, one uses [[]]. So first Lst element is Lst[[1]] and so on. One may use also element name:

```
>Lst[[2]]
>Lst[["Wife"]]
```

What is fifth element of the Lst? Second element of of fifth element? Get age of third kid. Get favorite drink.

Another way of accesing an element : listname\$elementname

>Lst\$Wife

```
>Lst$favor.food$drink
```

What are:

```
>Lst[1]  #(compare with Lst[[1]])
>Lst["Name"]  #(compare with Lst[["Name"]]
>Lst[c(4,2,1)]
>Lst[c("Name","Wife")]
```

Applying a function to each list element:

```
lapply (list, function)
unlist(list)
```

Prepare a list with three elements:

A: vector with integers from 1 to 10

- B: 25 numbers from uniform distribution between 1 and 10
- C: 7 numbers from normal distribution, with mean 10 and stand. dev. 2

What is the maximal element of A, B, C? Average? Variance? Maximum/mean/variance for all elements of the list? Make a list with food items: vegetables (onion, cucumber), fruits (apples, pears...), dairy (cheese).

Add "bread" element, containing "ryebread" and "croissant".

Add milk and cream to dairy.

Get length of each type of food (fruits vegetables etc.)

Construct a list of dataframes - each one contains results of one experiment birds<-lapply(1:3, function(x) {

Name list elements "smallBird", "mediumBrid" and "Bigbird". What can you say about offspring number and weight for the list elements?

Compute mean number of offspring and mean body weight for each bird and mean of both body weight and offspring number (use lapply and apply) Apply/tapply/lapply with your own function.

Write a function, which given a vector of values, compares first three values with next three values with a ttest.

Use it to compare, gene by gene, data from expression.txt file.

$$T = \frac{\overline{x}_a - \overline{x}_b}{\sqrt{\frac{s_a}{n_a} + \frac{s_b}{n_b}}}$$

```
source (file)
```

```
source (AniaFunc.R)
```

```
ifelse(condition, yes, no)
switch(expression, var1=whatToDo,var2=whatToDo)
break
next
```

HOME

Write a function which, given 2 vectors, writes all possible pairwise combinations of elements from two vectors, but a)excludes pairs wich sum is equal 21 b)for a given element from first vector, writes pairs only until it finds first pair which sum is equal 21.

Set operations		
A=c(1,4,7,19)		B=c(7,2,31,19)
intersect(A,B)	А∩В	7,19
union(A,B)	АUВ	1,4,2,7,19,31
setdiff(A,B)	A - B	1,4

%in%

a logical vector indicating if there is a match or not for its left operand match() a vector giving the position in table of the first match if there is a match, otherwise nomatch

```
>1:10 %in% c(1,3,5,9)
```

```
> match(1:10, c(1,3,5,9))
```

> which(1:10 %in% c(1,3,5,9))

> fruits<c("apple","pear","plum","grape","tomatoe","cucumber")</pre>

> vegetables<-c("potato","tomatoe","onion","cucumber")</pre>

Check with intersect common elements of both vectors

Add sets.

What elements are only in fruits?

What elements are only in vegetables?

Prepare a vector with 10 random letters of alphabet (use sample() and letters)

Prepare a vector with another 10 random letters of alphabet

Check how many letters appear in one/both sets.

Repeat, using sampling with replacement.

Check which letters reappeared.

Load the data from a file exercices6.Rdata

Check, how many genes have genes1 and genes2 in common.

How many genes do they contain in total?

How many genes are present on genes1 and not on genes2 and vice versa?

For common genes, prepare a dataframe with expression values from both lists.