C00-1 Vector basics

Creation of a vector

A vector is created using the c function. Here are some examples:

```
my_vector_1 <- c(1,2,3,4,5)
print(my_vector_1)
```

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

```
my_vector_2 <- c(1:10)
print(my_vector_2)
```

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

```
my_vector_3 <- c(10:5)
print(my_vector_3)
```

```
## [1] 10  9  8  7  6  5
```

```
my_vector_4 <- seq(from=0, to=30, by=10)
print(my_vector_4)
```

```
## [1]  0 10 20 30
```

You can skip the `print` function but just type the variable name if your standard out is the console. We will assume that from now on.

Length of a vector

To get the lenght of a vector, use the `length` function:

```
my_vector <- c(1:10)
length(my_vector)
```

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

Displaying and accessing the content of a vector

In order access the value(s) of a vector, you have to supply the position of the element in the vector within square brackets. Please note that indexing starts with 1:

```
# get the value of the element(s) at the specified position(s)
my_vector[1]
```

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

```
my_vector[1:3]
```

```
## [1] 1 2 3
```

```
my_vector[c(1,3)]
```

```
## [1] 1 3
```

Changing, adding or deleting an element of a vector

To overwrite an element, you have to access it following the logic above. To add an element, you have to cut the existing vector at the specified position and insert it. The result must be stored in a new variable (it will be new, even if you name it like the existing one). The same structure applies for deleting an element which is the same as combining the part of the vector before and after the value which should be deleted:

```
# modify an element at position 3
my_vector[3] <- 30

# add an element at position 4
my_added_vector <- c(my_vector[1:3], 20, my_vector[4:length(my_vector)])
my_added_vector
```

```
##  [1]  1  2 30 20  4  5  6  7  8  9 10
```

```
# delete an element at position 4
my_deleted_vector <- c(my_vector[1:3], my_vector[5:length(my_vector)])
my_deleted_vector
```

```
## [1]  1  2 30  5  6  7  8  9 10
```

Recycling of vectors

If one combines a shorter with a longer vector in e.g. an arithmetic operation, the shorter vector is recycled until the length of the longer vector is reached (i.e. the values are repeated over and over again)

```
my_short_vector <- c(1,2,3)
my_long_vector <- c(10,20,30,40,50,60)
my_sum_vector <- my_short_vector + my_long_vector
my_sum_vector
```

```
## [1] 11 22 33 41 52 63
```

For more information have a look at e.g. the respective data type site at Quick R. There you will also find an overview on how to get information about an object. Of course, looking into the package

documentation or search the web is always a good idea, too.