PP4RS | R Module

Slot 1 - Additional Exercises

Dora Simon

05.09.2018

Slot 1: Data Types

Exercises: Vector Operations

Define the vector `v<-c(1,2,3)`. Multiply the vector with 2. What happens?

```
v<-c(1,2,3)
v*2
```

```
## [1] 2 4 6
```

Define the vector `w<-c(1,2)`. Multiply the v with w. What happens?

```
w<-c(1,2)
v*w
```

```
## [1] 1 4 3
```

```
# "w" gets recycled!
# It is as if w was c(1,2,1)
# It gets copied until it has the same length as v.
```

Exercises: Subsetting Vectors

Define a vector `a<-c("blue","red","yellow","green")`. Print only the first value. (hint: use square brackets)

```
a<-c("blue","red","yellow","green")
a[1]
```

```
## [1] "blue"
```

Print the first and the last values of `a`. (hint: think about the first and the last values being a vector)

```
a[c(1, 4)]
```

```
## [1] "blue"  "green"
```

Exercises: Subsetting Vectors

Print all values of `a` except the first. (hint: use the minus sign)

```
a[c(-1)]
```

```
## [1] "red"    "yellow" "green"
```

Use `a` to print a vector of four elements, all called "red".

```
a[c(2, 2, 2, 2)]
```

```
## [1] "red" "red" "red" "red"
```

Exercises: Subsetting Vectors

Define `b<-c(1,2,3,4,5,6,7,8,9,10)`. Print all elements which are smaller than 5.

```
b<-c(1,2,3,4,5,6,7,8,9,10)
b[b<5]
```

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

Exercises: Subsetting Lists

Create the following list: `fancy_list <- list(a = 1:12, b = 'Pancakes are lovely, dear!', c = TRUE, d = list(-99, 1))`

Return a sublist of the last item in our fancy_list and check its structure.

```
fancy_list = list(a = 1:12, b = 'Pancakes are lovely, dear!',
                  c = TRUE, d = list(-99, 1))
fancy_list[4]
```

```
## $d
## $d[[1]]
## [1] -99
##
## $d[[2]]
## [1] 1
```

```
str(fancy_list[4])
```

```
## List of 1
##  $ d:List of 2
##   ..$ : num -99
##   ..$ : num 1
```

Exercises: Subsetting Lists

Return the last item in our fancy_list. Check what structure the returned element has.

```
fancy_list[[4]]
```

```
## [[1]]
## [1] -99
##
## [[2]]
## [1] 1
```

```
str(fancy_list[[4]])
```

```
## List of 2
##  $ : num -99
##  $ : num 1
```

```
#Now we obtain the element of the fancy list.
#So we obtain a list "d".
```

Exercises: Subsetting Lists

Return the element named `b` from our fancy_list. (hint: use the `$` sign to combine `fancy_list` and `b`)

```
fancy_list$b
```

```
## [1] "Pancakes are lovely, dear!"
```

```
str(fancy_list$b)
```

```
##  chr "Pancakes are lovely, dear!"
```

```
#Now we obtain the element of the fancy list.
#So we obtain all characters behind variable "b".
```

Exercises: Subsetting Lists

Return the first element of the last element.

```
fancy_list[[4]][[1]]
```

```
## [1] -99
```

```
str(fancy_list[[4]][[1]])
```

```
##  num -99
```

```
#Now we obtain -99, the last element of the list d.
```

Exercises: Subsetting Lists

Return the first element of the last element as a list.

```
fancy_list[[4]][1]
```

```
## [[1]]
## [1] -99
```

```
str(fancy_list[[4]][1])
```

```
## List of 1
##  $ : num -99
```

```
#Now we obtain a list that contains the element -99.
```