Exercise Series 2

Here is the recommended way to start your R session. It allows you to store the workspace and save your commands.

- Start Tinn-R. Open the file in which you store the S commands. If you open a new file, save it immediately.
- Start R by selecting '`Start preferred Rgui`' from Tinn-R's menu '`R`'.

If you do it in this sequence, you will be able to save the workspace upon leaving R by `q()`, and you can write files from within R.

1. **Vectors.** What is the output of the following commands? Try to predict the solutions before you type in the commands.

   (*) The first 4 items are identical to exercise 4 of series 1.

   We define: `x <- c(5,2,1,4); y <- rep(1,5)`

   a) (*) `sum(x)`
   `range(x)`
   `length(y)`
   `sum(y)`

   b) (*) `c(x,y,13)`

   c) (*) `x[4] * y[2]`
   `x[2:4] + x[1:3]`

   d) (*) `x <= 2`
   `x[x <= 2]`

   e) Read `?rep` and type
   `rep(1:4, 2); rep(1:4, each = 2); rep(1:4, c(2,2,2,2)); rep(1:4, c(2,1,2,1));`
   `rep(1:4, each = 2, len = 4); rep(1:4, each = 2, len = 10)`

   f) `2:3 ^ 2`
   `seq(2,-3)*c(-1,1)`

   g) Let `a <- LETTERS[1:3]`. Evaluate
   `paste('V',a, sep=':')`
   `nchar(paste(a, collapse='<'))`

   h) Let `u <- c(TRUE,FALSE,TRUE,TRUE)`. Evaluate
   `!u ; u | !u ; any(u)`

2. **Selecting elements,** From the data.frame `d.sport`, select

   a) the values of all athletes in the discipline `kugel`

   b) the performances of athletes `SMITH` and `MUELLER` for the first 3 disciplines, and their total (`punkte`).

   c) the data for all athletes who achieved at least 8600 `punkte`.

3. Matrices. Generate the following matrices.

a)
```
     [,1] [,2]
[1,]    1  101
[2,]    2  102
[3,]    3  103
[4,]    4  104
```
and
```
     [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]  101  102  103  104
```
Use `matrix` or `cbind` and `rbind` to do this. Store them as `t.m1` and `t.m2`. How can you obtain `t.m2` from `t.m1`?

b)
```
     [,1] [,2] [,3] [,4]
[1,]    5    0    0    0
[2,]    0    5    0    0
[3,]    0    0    5    0
[4,]    0    0    0    5
```
and
```
     [,1] [,2] [,3] [,4]
[1,]    3    0    0    0
[2,]    0    1    0    0
[3,]    0    0   -2    0
[4,]    0    0    0    0
```
Use `diag` to do this. Store them as `t.d1` and `t.d2`.

c) Divide `t.d1` by `t.d2` elementwise. Try the same for `t.m1` and `t.d2`.

d) Multiply the two matrices `t.m2` and `t.d2` by matrix multiplication. Do the same for `t.m1` and `t.d2`.

e) Matrix-multiply `t.d2` by the vector `5:8`, from the left as well as from the right.

4. Tables. Use the function `table` to

a) count the number of athletes that achieve less than and more than 8600 `punkte` (according to `d.sport`). Hint: You can (but need not) store a logical vector corresponding to the said condition first, and then use `table` on it.

b) cross tabulate this with the distinction between those that achieve less or more than 7.8 in `weit`.

c) Use `hist` with the argument `plot=FALSE` to obtain a classification of `punkte`. The first two items that you will see on the screen define the classification and the respective counts.

d) Read `?cut` (see "`## Default S3 method:`") and try to obtain the same result with `cut` and `table` (up to the names of the classes).

5. Estimation, Testing, Confidence Interval. The `data.frame sleep` is available from R directly (as are many other example datasets). Type `sleep` and `?sleep` to see what it contains.

a) Obtain the mean and the median of the `extra` hours of sleep for `group` 2.

b) Test the hypothesis that the 2 groups sleep essentially equally long by a Wilcoxon rank sum test (`wilcox.test`) and by a `t.test`.
Hint: Here, you can profitably use a formula as the first argument, and the data.frame as the second.

c) Get the confidence interval from the output of `t.test`. What is the parameter for which this confidence interval applies?