Scientific programming in R – pandemic edition

Dr. Henry Löffler-Wirth
Maria Schmidt

Interdisciplinary Center for Bioinformatics - IZBI,
University Leipzig
**Definition of own functions**

**Aim: Learn how to write your own functions.**

> `max( c( 1, 6, 18 ) )`

[1] 18

> `increment <- function( x )`

  ```
  inc <- x + 1
  return( inc )
  ```

> `increment( x=7 )`

[1] 8

> `increment( 7 )`

[1] 8

Functions always look like `functionname( parameter, ... )` They (usually) receive parameters, calculate something and return some value(s) or other object(s).

Virtually all R code calls functions, here `max()` and `c()`

Definition of a new custom function:

`functionname <- function( parameter, ... ){ code }`

Parameter list can also be empty.

`inc` is a local variable hidden in the function.

`return()` provides value of `inc` outside the function.

Our function is called as all functions.

Parameters can be addressed with their name, otherwise the values are mapped according to order (1st->1st etc)
Definition of own functions

Aim: Learn how to write your own functions.

```r
> myAdd <- function( x, y=1 )
  {
    return( x + y )
  }

> myAdd( x=7, y=10 )
[1] 17
> myAdd( 7, 10 )
[1] 17
> myAdd( 7 )
[1] 8
```

Function definition also accepts multiple parameters and default values.

Example for calling parameters by name and by mapping (ordering is important!!)

```
y=... is missing in function call. So default value is used as set in function definition.
```

Code blocks are recommended also for single line functions, but this is not mandatory.
Matrices

Aim: Set out to a second dimension, from vectors to matrices.

Build a matrix using \texttt{matrix(values, \#rows, \#columns)}.

\( \rightarrow \) 12 values into 3 rows \( \rightarrow \) \#columns=4 can be omitted.

Values are placed in column-wise fashion (see \texttt{?matrix}).

\begin{verbatim}
> num <- c(1:12)
> mat <- matrix(num, nrow=3)
> mat

[1,]  1  4  7 10
[2,]  2  5  8 11
[3,]  3  6  9 12

> is.vector(num)
[1] TRUE
> is.vector(mat)
[1] FALSE
> is.matrix(mat)
[1] TRUE
\end{verbatim}

Check modes of the variables:

\textit{num} is a vector (i.e. one-dimensional)

\textit{mat} is a matrix (i.e. two-dimensional)
Matrices

Aim: Set out to a second dimension, from vectors to matrices.

```r
> dim( mat )
[1] 3 4
```

Size of the matrix are queried by `dim()` for both dimensions (!convention: first rows, then columns!)

```r
> c( nrow( mat ), ncol( mat ) )
[1] 3 4
```

... or by using `nrow()` and `ncol()` functions for row and columns numbers, respectively.

```r
> str( mat )
int [1:3, 1:4] 1 2 3 4 5 6 7 8 9 10 ...
```

`str(objectname)` is one of the most useful functions in R. It gives details of the structure of any object. Here: a) it contains integer numbers, b) it has 1:3 rows and 1:4 columns, and c) the first values are shown
Matrices

Example data: Bitcoin prize over time (somewhat outdated, but nwm.)
Matrices

Aim: Learn how to work with matrices.

Loading data using `load()` from webspace via `url()`..... more on data I/O next time.

```r
> load( url("http://izbifs.izbi.uni-leipzig.de/~wirth/bitcoin.RData") )

> mat
     Offen    Hoch    Tief Geschlossen    Volumen
21.Nov18 4465.54 4675.73 4343.98   4602.17 6120120000
20.Nov18 4863.93 4951.61 4272.11   4451.87 8428290000
19.Nov18 5620.78 5620.78 4842.91   4871.49 7039560000
18.Nov18 5559.74 5653.61 5559.74   5623.54 4159680000
17.Nov18 5578.58 5578.58 5519.56   5554.33 4303150000
16.Nov18 5645.32 5657.02 5498.94   5575.55 5279320000
15.Nov18 5736.15 5774.82 5358.38   5648.03 7032140000
14.Nov18 6351.24 6371.55 5544.09   5738.35 7398940000
13.Nov18 6373.19 6395.27 6342.67   6359.49 4503800000
12.Nov18 6411.76 6434.21 6360.47   6371.27 4295770000
```

```r
> str(mat)
num [1:325, 1:5] 4466 4864 5621 5560 5579 ...
- attr(*, "dimnames")=List of 2
  ..$: chr [1:5] "Offen" "Hoch" "Tief" "Geschlossen" ...
```

*str() also shows row and column names.*
Matrices

Aim: Learn how to work with matrices.

Accessing matrix elements using
\[ \text{matrix[ row indices, col indices ]} \]

\[ \text{matrix[, col indices ]} \text{ is short for } \text{matrix[ 1:nrow(matrix), col indices ]} \]
\[ \rightarrow \text{ give me all rows} \]
\[ \text{matrix[ row indices, ]} \rightarrow \text{ all columns} \]
Matrices

Aim: Learn how to work with matrices.

Rows/columns can also be addressed by names instead of indices. This is highly recommended! 😊

```r
> mat[ "17.Jan18", ]
  Offen    Hoch   Tief Geschlossen   Volumen
 1.14311e+04 1.16780e+04 9.40229e+03 1.11886e+04 1.88306e+10

> mat[ , "offen" ]
   4465.54  4863.93  5620.78  5559.74  5578.58  5645.32  5736.15  6351.24  6373.19

> mat[ "20.Nov18", c("Offen","Hoch") ]
  Offen    Hoch
  4863.93  4951.61

> mat[ 1:3, c("Offen","Hoch") ]
  Offen    Hoch
  21.Nov18  4465.54  4675.73
  20.Nov18  4863.93  4951.61
  19.Nov18  5620.78  5620.78
```
Matrices

Aim: Learn how to work with matrices.

Filtering of data according to some criteria is a common use case in R:

```r
> buy.thresh <- 6000

> which( mat[, "Offen"] < buy.thresh )
[1] 1 2 3 4 5 6 7 146

> filter.mat <- mat[ which( mat[, "Offen"] < buy.thresh ) , , ]

> filter.mat
   Offen  Hoch  Tief Geschlossen  Volumen
21.Nov18 4465.54 4675.73 4343.98 4602.17 6120120000
20.Nov18 4863.93 4951.61 4272.11 4451.87 8428290000
19.Nov18 5620.78 5620.78 4842.91 4871.49 7039560000
18.Nov18 5559.74 5653.61 5559.74 5623.54 4159680000
17.Nov18 5578.58 5578.58 5519.56 5554.33 4303150000
16.Nov18 5645.32 5657.02 5498.94 5575.55 5279320000
15.Nov18 5736.15 5774.82 5358.38 5648.03 7032140000
29.Jun18 5898.13 6261.66 5835.75 6218.30 3966230016
```

I want to buy if price is below 6000$. Which is corresponding sub-matrix?
Programming tasks III

Load the matrix containing the bitcoin prizes and answer the following questions:

• What was the maximum bitcoin prize in 2018?

• What is the largest prize range within one day? (range: maximum – minimum)

• Write a function and use it in a loop to calculate a potential money flow for each day. Money flow means here: you buy 1 bitcoin at lowest prize of the day and sell it end of the day. How much money would you earn/lose over the year this way?

How to get rich with your new knowledge?