Additional assignments – Additional Practice
Danny Arends and Shijie Lyu

0) Open the file in your editor, and give a header, inspired on:

```
# Analysis of Hardy-Weinberg equilibrium
#
# copyright (c) 2016 - HU Berlin
# written by: Danny Arends
#
# last modified Apr, 2016
# first written Apr, 2016
setwd("<your location>/Assignments/")          # Instruct R to go to the assignments directory
```

R as a calculator

Use R to calculate the following:

1a) Euclidean division remainder of 20 and 3
1b) Euclidean division of 20 and 3
1c) the square root of -64
1d) the cube root of 4 * pi

Vectors

2a) Generate and round a vector of 20 random numbers between 1 and 3, and store/name it: "r1to3"
2b) Generate a vector containing 3 names of animals and store/name it: "animals"
2c) Use the r1to3 vector as an index into the animals vector, the resulting vector (containing 20 random animal names) should be stored/named: "rAnimals"
2d) Generate a vector containing 4 colors and store/name it: "myfavcolors"
2e) We can use the ***sample*** function to do the same thing, read the help ***?sample*** (look especially at the examples at the bottom) and randomly sample 20 times from the myfavcolors vector.
2f) Use the ***paste*** function to generate a vector containing 20 animals with random colors

Matrices

Hint: *At any time you can see a part of your matrix by using the* ***head()*** *function, head(animalMatrix)*

3a) Create an empty matrix (filled with NA) containing 100 rows (Individuals), and 2 colums (Species and Color) and name it: "animalMatrix"
3b) Add the two column names to your matrix using the ***colnames()*** function
3c) Add rownames (animal 1, animal2, .. , animal 100) to your matrix using the ***paste()*** and ***seq()*** function
3d) Fill the first column (Species) with 100 randomly selected species (using the sample function and the animals vector)
3e) Fill the second column (Color) with 100 random colors (using the sample function and the my favcolors vector)

3f) Use the ***paste()*** function, and combine the color and species together, then use the ***table()*** function (?table) on the resulting vector, to automatically count how many of each element there is

Loops

4a) Use a ***for*** loop and print out the rows of the animalMatrix, first print the row number, followed by the two row elements separate by a – and end the printing of a row by using a newline (\n). The output must look like this:

1 Black-Cow
….
100 White-Sheep

4b) Add another column (Weight) to the matrix using the ***cbind*** function, initially fill the column with NAs
4c) For each animal estimate it's average weight (e.g. Cow = 600, Sheep = 50), create a vector containing these weights, and name it animalWeights.
4d) Use the ***names***() function to set the correct name to the elements of the animalWeights vector
4e) Use a for loop to go through the data row by row, at each row do the following:

1) Use the species name as the index into the animalWeights vector to get the average weight of that species, and store it in a variable avgWeight
2) Take the square root of the avgWeight, and store it in a variable called speciesSD
3) Use the rnorm function to generate a random weight value in range of avgWeight +/- speciesSD
4) Store the weight in the weigth column of the matrix

4f) Use the ***head*** function to print the first couple of rows of your matrix, check if the weights you generated are in line with what you expect.

Basic functions

5) Create a function that returns the result of a dice, for which you can specify the type such as:
D20 = 20 sided dice
D6 = normal 6 sided dice
6D8 = sum of 6 rolls of an 8 sided dice
3D6+4 = sum of 3 rolls of a 6 sided dice + 4

```
rollDice <- function(D, N = 1, A = 0) {

}
```