Exercise Sessions

Session 2: vector, matrix, and data frame

Get the best from your sequences

May, 25–29 2015, FMB, UNESP, Botucatu, Brasil

Luigi Cerulo

Working with Vectors

1. Create a vector called `vec1` containing the numbers 2,5,8,12 and 16
2. Use `x:y` notation to make a second vector called `vec2` containing the numbers 5 to 9
3. Subtract `vec2` from `vec1` and look at the result
4. Use `seq()` function to make a vector of 100 values starting at 2 and increasing by 3 each time
5. Extract the values at positions 5,10,15 and 20 in the vector you made previously
6. Write a program that reads and integer N and generates a vector composed by the first N even numbers
7. Write a program that reads and integer N and generates a vector composed by the first N even numbers

Matrices and Data Frames

1. Enter the following into a vector with the name `mouse.colour`: purple red yellow brown
2. Display the 2nd element in the vector (red) in the console.
3. Enter the following into a vector with the name `mouse.weight`: 23 21 18 26
4. Join the 2 vectors together to make a data frame named `mouse.info` with 2 columns and 4 rows.
5. Display the data frame in the console.
6. Display just row 3 in the console
7. Display just column 1 in the console
8. Display the item of data in row 4, column 1.
9. Set your working directory to where the data files are stored. Make sure that the folder of data files has been unzipped.
10. Read the file `Child_Variants.csv` into a new data structure. This is a comma separated file so you should use `read.csv()` or the import feature of Rstudio
11. View the data set to check that it has imported correctly (`view()` function).
12. Display column 11.
13. Calculate the mean of the column named `MutantReadPercent`

Filtering

1. Create a filtered version of the child variants dataset which only includes rows where the `MutantReadPercent` is greater or equal 70

2. From the filtered list calculate the number of lines for mutations from C to T, C to G, and C to A and see if there is a preference for one of these mutations. You will need to use an exact match filter (using 2 equals signs!) against the text in the MUTATION column. To get the count you can simply sum() the boolean array values (true counts as 1, false counts as 0). Create a vector of the counts for the different C mutation frequencies.

3. Read the `movies.csv` file in a data structure. Select the most expensive drama movies (movies with a budget more than the upper quartile, i.e. 75th percentile) and compute their budget average. From computation remove movies where the budget is not available (NA symbol). The following code shows hot to remove those movies and how to compute the upper quartile threshold:

```
# reads the movies.csv file into a data frame
d = read.csv("movies.csv")

# creates a new data structure d1
# with movies having a budget different than NA
d1 = d[!is.na(d$budget),]

# computes the upper quartile threshold z
z = quantile(d1$budget,0.75)
```

4. Compute the average budget of the movies of the last 10 years in all different categories (Action, Animation, Comedy, Drama, Documentary, Romance). Also in this case remove movies with no budget information. Wich category is the most expensive?