# Workshop Activity 1 Answers

Fabio Setti

2024-02-19

## Basic Functions and Calculations

**1.** Create an object (1D numeric vector) that contains all the *even numbers* from 1 to 10 (10 included). Name the object **even_10**.

```
even_10 <- c(2, 4, 6, 8, 10)
```

**2.** Create an object (1D numeric vector) that contains all the *odd numbers* from 1 to 10. Name the object **odd_10**.

```
odd_10 <- c(1, 3, 5, 7, 9)
```

**3.** Create an object (1D numeric vector) that contains the multiplication between the respective elements of **even_10** and **odd_10**. Name the object **mult_10**. Additionally, calculate the sum of the elements of the **mult_10** object.

**HINT:** You can apply mathematical operations to vectors of the same length (Why same length?). Mathematical operations will be performed between the respective elements of each vector.

```
mult_10 <- even_10*odd_10
```

**4.** Calculate the *mean* of all of the numbers contained in the **even_10**, **odd_10**, and **mult_10** (so only 1 mean, not 3). Use the `mean()` function for this.

**HINT:** the `mean()` function only takes in one object at a time, maybe you can get creative with the `c()` function?

```
x <- c(even_10,
       odd_10,
       mult_10)

mean(x)
```

```
## [1] 16.33333
```

**4.1.** Calculate the same mean, but do so without using the `mean()` function! The mathematical formula for the mean is $Mean = \frac{\sum x_i}{n}$, where the numerator is the sum of all of your values, and the denominator is how many values you have.

**HINT:** there is a function that you can use to count how many elements there are in an object.

```
# x is defined above, so the mean is

sum(x)/length(x)
```

```
## [1] 16.33333
```

**5.** Calculate the *standard deviation* of all of the numbers contained in the **even_10**, **odd_10**, and **mult_10** (so only 1 standard deviation, not 3). The same hint from above applies, but you will also need to find the function that calculates the standard deviation!

```
sd(x)
```

```
## [1] 24.88162
```

**5.1.** Calculate the same standard deviation without using the standard deviation function! The mathematical formula for the mean is $SD = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n-1}}$, where $x_i$ represents every single values, $\bar{x}$ represent the mean. You will also need to find the function that calculates the square root.

**HINT:** here you need to place parenthesis "()" to tell R the correct order of operations and functions.

```
# You don't need to create this many object, but it makes the formula clearer

# the numerator

num <- sum((x - mean(x))^2)

# the denominator

den <- length(x) - 1

# calculate the SD

sqrt(num/den)
```

```
## [1] 24.88162
```

## Importing Data and Subsetting

**6.** Import the **mammal_sleep.csv** data set and name it **dat**. you can find the description of the variables in the data set here (https://www.openintro.org/data/index.php?data=mammals). Additionally, there is an extra variable, *primate*, that specifies whether the mammal is a primate or not. Explore the data either visually or with the `str()` function to get a better sense of what you are looking at!

```
library(rio)

dat <- import("Mammal_Sleep.csv")
```

**7.** The `summary()` function has MANY uses in R (the output is different depending on what object you use as input). When applied to a `data.frame` object, `summary()` calculates some descriptive statistics for numeric variables. Run the following code:

```
sum_tab <- summary(dat)
```

Now, extract *only* the means of the `BrainWt` and `TotalSleep` variables from the **sum_tab** object.

**HINT:** You can investigate what and how information is stored in the **sum_tab** object by just running `sum_tab`, which will print all of the store information. Additionally, note that this is a subsetting problem, so try to identify what the dimensions of the **sum_tab** object are (looking at the environment may help!).

```
sum_tab <- summary(dat)

sum_tab[4,c(3,6)]
```

```
##             BrainWt          TotalSleep
## "Mean   : 283.13  "   "Mean   :10.53  "
```

**8.** How many animals in the data are primates?

**HINT:** I would look for a function that counts unique elements in a vector.

```
table(dat$Primate)
```

```
##
##  N  Y
## 54  8
```

**8.1.** can you find a way to output *only* the number for primates?

**HINT:** This question has to do with dimensions and subsetting.

```r
# tables for only one variable only have 1 Dimensions

table(dat$Primate)[2]
```

```
## Y
## 8
```

**9.** Can you create a new variable in the "dat" data set that is the proportion of body weight that brain weight takes up? That is. if body weight is 2 and brain weight is 0.2, then brain weight takes up .1 (0.2/2 = .1, or 10%) of the total body weight. Name the new variable **br_to_bd_weight**.

**HINT 1:** you should be able to calculate the proportion in a really short line of code, a hint for one of the previous questions may help you out!

**HINT 2:** you can create a new variable in a data.frame as follows:

```r
# this is not runnable code, just a conceptual example

name_of_data$new_variable <- the variable that you want to add to the data
```

```r
# Just divide the two variables in question.

dat$br_to_bd_weight <- dat$BrainWt/dat$BodyWt
```