HW 1

Due: Thursday January 18, 2pm

Instructions

- Produce your assignment as a RMarkdown document rendered to pdf (knit to pdf).
- Also submit your Rmd file (it will not be graded but we want it for reference purposes).
- Show all the code (use `echo=TRUE` as option in R chunks) as well as the results.
- 10 points per exercise.
- See Syllabus for HW policies.

Data for exercises:

```
data <- c(23, 89,  1, 34,
          26, 85, 24, 43,
          23, 93, 29, 45,
          32, 42, 43, NA,
          21, 54, 37, 76)
```

Positive Integral Quantities Index Vector

- Values in the index vector must lie in the set {1, 2, . . . , length(x)}.
- Values not in this set will produce NAs.
- Corresponding elements of the vector are selected and concatenated, in that order, in the result.
- The index vector can be of any length and the result is of the same length as the index vector.

Exercise 1:

Extract the elements of `data` found in positions 8, 3, 7, and 5.

Answer

```
data[c(8,3,7,5)]
```

```
## [1] 43  1 24 26
```

Exercise 2:

Random sample and simple random sample:

First search on the Internet which is the difference between:

- simple random sample
- random sample (keep in mind that sometimes people confuse both namings)

(Note: the difference should be related to sampling with or without replacement).

Generate both a simple random sample and a random sample of size 10 from `data`.

Hint: look the help for of function `sample`. Extracting directly from `data`, a simple random sample is:

```
sample(data, 10)
```

```
##  [1] 23  1 37 42 89 43 45 23 54 34
```

```r
sample(data, 10, TRUE)
```

```
##  [1] 85 34 37 43 26 89 24 NA 29 89
```

However, for didactical purposes on index vectors, I ask you to first generate a random index vector, and use it to extract the chosen elements from `data`.

Answer

```r
data[sample(length(data), 10, TRUE)]
```

```
##  [1] 26 23 NA 34 76 26 93 32 24 54
```

Exercise 3:

Systematic sample: Get a systematic sample of size `n = 5` from `data` by extracting each value that lies every `K = N/n` elements (where `N` is the total number of elements in `data`).

Your first element needs to be randomly determined as a number between 1 and K.

Note: Your result needs to be a vector containing the 5 elements that are part of your systematic sample.

Answer

```r
a <- sample(1:(length(data)/5),1)
data[seq(from = a, to = length(data), by = (length(data)/5))]
```

```
## [1] 34 43 45 NA 76
```

Negative Integral Quantities Index Vector

- Negative values in the index vector specify the values to be excluded.

Exercise 4:

Using negative indexes, obtain a sub-vector by removing from `data` elements in positions 3, 7, and at the end (the position at the end must be removed in a general way, even for vectors with different sizes).

Answer

```r
data[c(-3, -7, -length(data))]
```

```
##  [1] 23 89 34 26 85 43 23 93 29 45 32 42 43 NA 21 54 37
```

Exercise 5:

`data` has an even number of elements. Devise a general strategy to remove only the 2 elements found in the middle.

Answer

```r
data[c(-length(data)/2,-(length(data)/2+1))]
```

```
##  [1] 23 89  1 34 26 85 24 43 23 45 32 42 43 NA 21 54 37 76
```

Exercise 6:

Obtain a vector of the differences (second - first, third - second, . . .) using negative indexes.

Note: function `diff` does exactly this. you can use it to verify that the result of your solution (has to be a *one liner*) is OK:

```
diff(data)
```

```
##  [1]  66 -88  33  -8  59 -61  19 -20  70 -64  16 -13  10   1  NA  NA  33
## [18] -17  39
```

Answer

```
data1 <- data[c(-1)]
data2 <- data[c(-length(data))]
c(data1-data2)
```

```
##  [1]  66 -88  33  -8  59 -61  19 -20  70 -64  16 -13  10   1  NA  NA  33
## [18] -17  39
```

Logical Index Vectors

- Index vector must be of the *same length* as the vector from which elements are to be selected.
- If the length of the index vector is less than the vector, the index vector will be recicled with perhaps unwanted results.
- Values corresponding to TRUE in the index vector are selected.
- Values corresponding to FALSE are omitted.

Exercise 7:

Write an R expression that will return the sum value of 820 for the elements of `data`.

Note: using function `sum` directly yields

```
sum(data)
```

```
## [1] NA
```

as `NA`s are contagious. `sum` has the parameter `na.rm` to remove `NA`s before performing the sum.

```
sum(data, na.rm = TRUE)
```

```
## [1] 820
```

Your solution cannot use `na.rm`, but instead use logical indexes, and be general (not only for this specific example).

Anwer

```
sum(data[!is.na(data)])
```

```
## [1] 820
```

Exercise 8:

Write an R expression that will return the positions of values 23 and 43 in `data`.

Answer

```
which(data == 23 | data == 43)
```

```
## [1]  1  8  9 15
```

Exercise 9:

Verify the "Empirical rule", that states that when data is drawn from a Normal distribution,

- about 68% of the data is in the interval `(mean(x) - sd(x), mean(x) + sd(x))`
- about 95% of the data is in the interval `(mean(x) - 2 * sd(x), mean(x) + 2 * sd(x))`
- about 99.7% of the data is in the interval `(mean(x) - 3 * sd(x), mean(x) + 3 * sd(x))`

Hint: use "x <- rnorm(n)" to randomly generate a sample of size n from a standard normal distribution.

Use n = 1000

Answer

```
x <- rnorm(1000)
length(which(x <= sd(x) & x >= -sd(x)))/length(x)
```

```
## [1] 0.679
```

```
length(which(x <= 2 * sd(x) & x >= -2 * sd(x)))/length(x)
```

```
## [1] 0.954
```

```
length(which(x <= 3 * sd(x) & x >= -3 * sd(x)))/length(x)
```

```
## [1] 0.997
```