

Homework Assignment 9

Armen Mkrtumyan

16/04/23

Write your solutions and comments in this markdown file and submit it into moodle.

Part 1: Basic Operations with R (50 points)

1.1. Create a vector *numbers* that contains numbers from 1 to 20. (5 points)

```
numbers <- c(1:20)
print(numbers)
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
```

1.2. Create a vector of natural numbers from 21 to 27 (both ends inclusive) and letters “A”, “B”, “C”. After check the data type of the vector and explain. (5 points)

```
new_vector <- c(21:27, "A", "B", "C")
print(new_vector)
```

```
## [1] "21" "22" "23" "24" "25" "26" "27" "A" "B" "C"
```

```
print(class(new_vector))
```

```
## [1] "character"
```

```
#We can see that the output is `character`
#since when different data types are given to vector,
#R coerces the types into one common.
```

1.3. Create a vector “numbers_2”, containing integers from 1 to 100 and 200 to 300. Print the length. (5 points)

```
numbers_2 <- c(1:100, 200:300)
print(length(numbers_2))
```

```
## [1] 201
```

1.4. Create another vector *odd_numbers* that contains only odd numbers from 1 to 20 using function *seq()*. (5 points)

Hint: You can see the documentation of the function by typing *?seq*, pay attention to the argument *by*.

```
odd_numbers = seq(1, 20, 2)
print(odd_numbers)
```

```
## [1] 1 3 5 7 9 11 13 15 17 19
```

1.5. Subset the vector *odd_numbers* so that only numbers greater than 10 remain in the new vector. Name the new vector as you wish. The resulting vector should have 5 elements, check the length. (5 points)

```
sub_odd = odd_numbers[odd_numbers > 10]
print(length(sub_odd))
```

```
## [1] 5
```

1.6. Consider the vector given below. Merge this vector with the vector obtained in the previous point. Calculate the sum, mean, standard deviation for the obtained vector. (5 points)

```
v1 <- c(12, 3, 11, 8, 4, 7, 5)
```

```
merged_vector = c(v1, sub_odd)
```

```
paste(c("sum is: ", sum(merged_vector)), collapse = " ")
```

```
## [1] "sum is: 125"
```

```
paste(c("mean is: ", mean(merged_vector)), collapse = " ")
```

```
## [1] "mean is: 10.4166666666667"
```

```
paste(c("mean is: ", sd(merged_vector)), collapse = " ")
```

```
## [1] "mean is: 5.14266173269929"
```

1.7. Build a 4x3 matrix and populate it with the vector generated above, where matrix elements are filled by row. (5 points)

```
my_matrix = matrix(data = merged_vector, nrow = 4, ncol = 3, byrow = T)
print(my_matrix)
```

```
##      [,1] [,2] [,3]
## [1,]  12   3  11
## [2,]   8   4   7
## [3,]   5  11  13
## [4,]  15  17  19
```

1.8. Choose element from forth row and second column assigning it to a new variable *num_1*. (5 points)

```
num_1 = my_matrix[4, 2]
print(num_1)
```

```
## [1] 17
```

1.9. Make the rownames of matrix the followings respectively: first_row, second_row, third_row, forth_row. (5 points)

```
rownames(my_matrix) <- c("first_row", "second_row", "third_row", "forth_row")
print(my_matrix)
```

```
##      [,1] [,2] [,3]
## first_row  12   3  11
## second_row   8   4   7
## third_row   5  11  13
## forth_row  15  17  19
```

1.10. For each row calculate the sum of elements in that row. (5 points)

Hint: During the lecture you used *colSums* function, look at *?rowSums*.

```
print(my_matrix)
```

```
##      [,1] [,2] [,3]
## first_row  12   3  11
## second_row   8   4   7
## third_row   5  11  13
## forth_row  15  17  19
```

```
print(rowSums(my_matrix))
```

```
## first_row second_row third_row forth_row
##          26          19          29          51
```

Part 2: Working with data (50 points)

2.1. Run the previous lines to have the dataframe. State the number of observations and features. Use *str* function. (5 points)

```
df <- data.frame("YEARS_ACTIVE" = c(2001:2010),
                 "SCORED_GOALS" = c(20, 14, 31, 11, 21, 22, 19, 19, 15, 11),
                 "PLAYED_GAMES" = c(6, 7, 10, 3, 4, 7, 7, 7, 6, 6))
print(str(df))
```

```
## 'data.frame':    10 obs. of  3 variables:
## $ YEARS_ACTIVE: int  2001 2002 2003 2004 2005 2006 2007 2008 2009 2010
## $ SCORED_GOALS: num  20 14 31 11 21 22 19 19 15 11
## $ PLAYED_GAMES: num   6  7 10  3  4  7  7  7  6  6
## NULL
```

2.2. The column *SCORED_GOALS* shows the goals. Find the maximum, minimum and mean values for scored goals. (10 points)

```
paste(c("max is: ", max(df["SCORED_GOALS"])), collapse = " ")
```

```
## [1] "max is:  31"
```

```
paste(c("min is: ", min(df["SCORED_GOALS"])), collapse = " ")
```

```
## [1] "min is:  11"
```

```
paste(c("mean is: ", mean(df$SCORED_GOALS)), collapse = " ")
```

```
## [1] "mean is:  18.3"
```

2.3. Subset the dataframe to have the columns *YEARS_ACTIVE* and *PLAYED_GAMES*. Store it in variable *df_subset*. (8 points)

```
df_subset = df[c("YEARS_ACTIVE", "PLAYED_GAMES")]
print(df_subset)
```

```
##   YEARS_ACTIVE PLAYED_GAMES
## 1          2001           6
## 2          2002           7
## 3          2003          10
## 4          2004           3
## 5          2005           4
## 6          2006           7
## 7          2007           7
## 8          2008           7
## 9          2009           6
## 10         2010           6
```

2.4. Subset the dataframe to have only the rows when *YEARS_ACTIVE* equals 2004, 2007 and 2010, and exclude columns *SCORED_GOALS* and *PLAYED_GAMES*. (12 points)

```
df[df$YEARS_ACTIVE %in% c(2002, 2004, 2007, 2010), !(names(df) %in% c("SCORED_GOALS", "PLAYED_GAMES"))]
```

```
## [1] 2002 2004 2007 2010
```

2.5. Write a for statement that will compute the sum of even numbers from 1 to 11. Note that modulo operator in R that checks remainder after a division is “%%” rather than “%”. (15 points)

```
sum = 0
for (num in 1:11)
{
  if(num %% 2 == 0){
    sum = sum + num
  }
}
print(sum)
```

```
## [1] 30
```