Stat 291 - Recitation 7

Orçun Oltulu

10 / 12 / 2021

1. Loops:

Exercise 1.1:

Create an empty Weight_change variable, then fill it with after - before for each person, using a **for loop**

```
set.seed(291)
id <- 1:10
before <- sample(50:90, size = 10)
after <- sample(55:89, size = 10)
data <- data.frame(id, before, after)
```

```
Weight_change <- numeric()

for(i in 1:10){
  Weight_change[i] <- data$after[i] - data$before[i]
}

Weight_change
```

```
##  [1]  7  0 36 27  8  1 23 20 12 -4
```

Exercise 1.2:

Using while() loop print integers from 1 to 9.

```
i <- 1
while(i < 10){
  print(i)
  i = i + 1
}
```

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

```
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
```

Exercise 1.3:

Using while() loop print integers that can be divisible by 3 from 1 to 50.

```
i <- 1

while(i < 50){

  if(i %% 3 == 0){
    print(i)
  }

  i <- i + 1

}
```

```
## [1] 3
## [1] 6
## [1] 9
## [1] 12
## [1] 15
## [1] 18
## [1] 21
## [1] 24
## [1] 27
## [1] 30
## [1] 33
## [1] 36
## [1] 39
## [1] 42
## [1] 45
## [1] 48
```

Exercise 1.4:

Assume you have list of your friends;

```
friends_names <- c("Frodo","Bilbo","Sam","Gandalf","Sauron",
                   "Aragorn","Legolas", "Galadriel", "Balrog")
```

Part A.

Print all the names using while loop.

```
i <- 1
while(i <= length(friends_names)){
  print(friends_names[i])
  i <- i + 1
}
```

```
## [1] "Frodo"
## [1] "Bilbo"
## [1] "Sam"
## [1] "Gandalf"
## [1] "Sauron"
## [1] "Aragorn"
## [1] "Legolas"
## [1] "Galadriel"
## [1] "Balrog"
```

Part B.

Print the names till "Aragorn".

```
i <- 1
while(i <= length(friends_names)){
  print(friends_names[i])
  i <- i + 1
  if(friends_names[i] == "Aragorn"){
    break
  }
}
```

```
## [1] "Frodo"
## [1] "Bilbo"
## [1] "Sam"
## [1] "Gandalf"
## [1] "Sauron"
```

Part C.

Print the names without "Sam", "Sauron", "Balrog".

```r
i <- 0

while(i < length(friends_names)){
  i <- i + 1

  if(friends_names[i] %in% c("Sam", "Sauron", "Balrog")){
    next
  }
  print(friends_names[i])

}
```

```
## [1] "Frodo"
## [1] "Bilbo"
## [1] "Gandalf"
## [1] "Aragorn"
## [1] "Legolas"
## [1] "Galadriel"
```

Exercise 1.5:

Part A.

Using **while** loop find the minimum value of x such that $\sum_{i=1}^{x} \geq 1234$

```r
total <- 0
max_val <- 1234
x <- 1

while(TRUE){

  total <- total + x

  if(total >= max_val){
    cat("sum of integers from 1 to", x, "is", total)
    break
  }

  x <- x + 1

}
```

```
## sum of integers from 1 to 50 is 1275
```

Part B.

Using **repeat** loop find the minimum value of x such that $\sum_{i=1}^{x} \geq 1234$

```
total <- 0
max_val <- 1234
x <- 1

repeat{

  total <- total + x
  if(total >= max_val){
    cat("sum of integers from 1 to", x, "is", total)
    break
  }
  x <- x + 1

}
```

```
## sum of integers from 1 to 50 is 1275
```

Exercise 1.6:

Write a while loop to calculate the sum of given integers (take inputs using readline function) and if user gives "exit" or "0", make sure your loop stops and prints the sum of given integers.

```
total <- 0
while(TRUE){
  x <- readline(prompt = "Enter an integer: ")

  if(x %in% c("exit","Exit",0)){
    cat("total is", total)
    break
  }

  total <- total + as.numeric(x)
}
```

2. Cleaning Data:

Exercise 2.1:

```
friends_names2 <- c("Frodo","Bilbo","Sam","Gandalf","Sauron",
                    "Aragorn","Legolas", "Galadriel", "Balrog",
```

```
                    "Balrog","Aragorn","Bilbo","Sam")
```

Part A.

Find the proportion of the duplicated values.

```
mean(duplicated(friends_names2))
```

```
## [1] 0.3076923
```

Part B.

Remove the duplicated values.

```
friends_names2[!duplicated(friends_names2)]
```

```
## [1] "Frodo"     "Bilbo"     "Sam"       "Gandalf"   "Sauron"    "Aragorn"
## [7] "Legolas"   "Galadriel" "Balrog"
```

```
# alternatively,
unique(friends_names2)
```

```
## [1] "Frodo"     "Bilbo"     "Sam"       "Gandalf"   "Sauron"    "Aragorn"
## [7] "Legolas"   "Galadriel" "Balrog"
```

Exercise 2.2:

Part A.

Read "iris_new" dataset into R.

```
data <- read.csv("iris_new.csv")
```

Part B.

Check if there are any duplicated lines and remove them from your data set.

```
data_new <- data[!duplicated(data),]
```

Part C.

Check if there are any NA values, remove them from your data set.

```
data_new_NAfree <- na.omit(data_new)
```

Part D.

Sort your data frame by Species.

```
data_new_NAfree[order(data_new_NAfree$Species),]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 2           5.1         3.5          1.4         0.3     setosa
## 3           5.0         3.6          1.4         0.2     setosa
## 6           4.8         3.1          1.6         0.2     setosa
## 9           5.4         3.7          1.5         0.2     setosa
## 13          4.9         3.1          1.5         0.2     setosa
## 23          5.1         3.8          1.9         0.4     setosa
## 4           6.4         2.9          4.3         1.3 versicolor
## 8           4.9         2.4          3.3         1.0 versicolor
## 10          5.5         2.3          4.0         1.3 versicolor
## 20          5.5         2.4          3.8         1.1 versicolor
## 22          6.2         2.2          4.5         1.5 versicolor
## 1           4.9         2.5          4.5         1.7  virginica
## 5           6.2         2.8          4.8         1.8  virginica
## 15          7.7         3.0          6.1         2.3  virginica
## 16          5.8         2.8          5.1         2.4  virginica
## 18          6.5         3.2          5.1         2.0  virginica
## 19          5.6         2.8          4.9         2.0  virginica
```

Part E.

Sort your data frame by their Sepal features.

```
data_new_NAfree[order(data_new_NAfree$Sepal.Length,
                      data_new_NAfree$Sepal.Width),]
```

```
##    Sepal.Length Sepal.Width Petal.Length Petal.Width    Species
## 6           4.8         3.1          1.6         0.2     setosa
## 8           4.9         2.4          3.3         1.0 versicolor
## 1           4.9         2.5          4.5         1.7  virginica
## 13          4.9         3.1          1.5         0.2     setosa
## 3           5.0         3.6          1.4         0.2     setosa
## 2           5.1         3.5          1.4         0.3     setosa
## 23          5.1         3.8          1.9         0.4     setosa
## 9           5.4         3.7          1.5         0.2     setosa
## 10          5.5         2.3          4.0         1.3 versicolor
## 20          5.5         2.4          3.8         1.1 versicolor
## 19          5.6         2.8          4.9         2.0  virginica
## 16          5.8         2.8          5.1         2.4  virginica
## 22          6.2         2.2          4.5         1.5 versicolor
## 5           6.2         2.8          4.8         1.8  virginica
## 4           6.4         2.9          4.3         1.3 versicolor
## 18          6.5         3.2          5.1         2.0  virginica
```

```
## 15          7.7         3.0          6.1         2.3  virginica
```