

# Modul 5

Ahmad Abdullah Azzam

10/27/2021

## D.Latihan

Pada latihan ini soal no 1 akan menggunakan *dataset*: AS *murders*.

```
library(dslabs)
data(murders)
```

1. Fungsi `nchar` dapat digunakan untuk menghitung jumlah karakter dari suatu vektor karakter. Buatlah satu baris kode yang akan menyimpan hasil komputasi pada variabel '`new_names`' dan berisi singkatan nama negara ketika jumlah karakternya lebih dari 8 karakter.

```
new_long = nchar(murders$state)
num = 1

new_df = data.frame(name = murders$state, panjang = new_long, new_name = 1:51)

while (num <= 51){
  if(new_long[num] > 8 ){
    new_df$new_name[num] = murders$abb[num]
  }
  else{
    new_df$new_name[num] = murders$state[num]
  }
  num = num + 1
}

print(new_df)
```

```
##           name panjang new_name
## 1      Alabama      7  Alabama
## 2       Alaska      6   Alaska
## 3      Arizona      7  Arizona
## 4    Arkansas      8 Arkansas
## 5   California     10      CA
## 6    Colorado      8 Colorado
## 7  Connecticut     11      CT
## 8    Delaware      8 Delaware
## 9 District of Columbia 20      DC
## 10   Florida      7   Florida
```

|       |                |    |          |
|-------|----------------|----|----------|
| ## 11 | Georgia        | 7  | Georgia  |
| ## 12 | Hawaii         | 6  | Hawaii   |
| ## 13 | Idaho          | 5  | Idaho    |
| ## 14 | Illinois       | 8  | Illinois |
| ## 15 | Indiana        | 7  | Indiana  |
| ## 16 | Iowa           | 4  | Iowa     |
| ## 17 | Kansas         | 6  | Kansas   |
| ## 18 | Kentucky       | 8  | Kentucky |
| ## 19 | Louisiana      | 9  | LA       |
| ## 20 | Maine          | 5  | Maine    |
| ## 21 | Maryland       | 8  | Maryland |
| ## 22 | Massachusetts  | 13 | MA       |
| ## 23 | Michigan       | 8  | Michigan |
| ## 24 | Minnesota      | 9  | MN       |
| ## 25 | Mississippi    | 11 | MS       |
| ## 26 | Missouri       | 8  | Missouri |
| ## 27 | Montana        | 7  | Montana  |
| ## 28 | Nebraska       | 8  | Nebraska |
| ## 29 | Nevada         | 6  | Nevada   |
| ## 30 | New Hampshire  | 13 | NH       |
| ## 31 | New Jersey     | 10 | NJ       |
| ## 32 | New Mexico     | 10 | NM       |
| ## 33 | New York       | 8  | New York |
| ## 34 | North Carolina | 14 | NC       |
| ## 35 | North Dakota   | 12 | ND       |
| ## 36 | Ohio           | 4  | Ohio     |
| ## 37 | Oklahoma       | 8  | Oklahoma |
| ## 38 | Oregon         | 6  | Oregon   |
| ## 39 | Pennsylvania   | 12 | PA       |
| ## 40 | Rhode Island   | 12 | RI       |
| ## 41 | South Carolina | 14 | SC       |
| ## 42 | South Dakota   | 12 | SD       |
| ## 43 | Tennessee      | 9  | TN       |
| ## 44 | Texas          | 5  | Texas    |
| ## 45 | Utah           | 4  | Utah     |
| ## 46 | Vermont        | 7  | Vermont  |
| ## 47 | Virginia       | 8  | Virginia |
| ## 48 | Washington     | 10 | WA       |
| ## 49 | West Virginia  | 13 | WV       |
| ## 50 | Wisconsin      | 9  | WI       |
| ## 51 | Wyoming        | 7  | Wyoming  |

2. Buat fungsi `sum_n` yang dapat digunakan untuk menghitung jumlah bilangan bulat dari 1 hingga  $n$ .  
Gunakan pula fungsi ini untuk menentukan jumlah bilangan bulat dari 1 hingga 5000.

```
sum_n <- function(n){
  jumlah = 0
  num = 1

  while (num <= n){
    jumlah = jumlah + num
    num = num + 1
  }
  return(jumlah)
}
```

```
}
sum_n(5000)
```

```
## [1] 12502500
```

3. Buat fungsi `compute_s_n` yang dapat digunakan untuk menghitung jumlah  $S_n = 1^2 + 2^2 + 3^2 + \dots + n^2$ . Tampilkan hasil penjumlahan  $n=10$ .

```
compute_s_n <- function(n){
  jumlah = 0
  num = 1

  while (num <= n){
    jumlah = jumlah + (num*num)
    num = num + 1
  }

  return(jumlah)
}

compute_s_n(10)
```

```
## [1] 385
```

4. Buat vektor numerik kosong dengan nama: `s_n <- vector ("numeric", 25)`. Simpan di hasil komputasi  $S_1, S_2, \dots, S_{25}$  menggunakan FOR-LOOP.

```
s_n <- vector ("numeric", 25) # create an empty vector

m <- 25

for(n in 1:m){
  s_n[n] <- compute_s_n(n)
}

s_n
```

```
## [1] 1 5 14 30 55 91 140 204 285 385 506 650 819 1015 1240
## [16] 1496 1785 2109 2470 2870 3311 3795 4324 4900 5525
```

5. Ulangi langkah pada soal no.4 dan gunakan fungsi `sapply`.

```
n <- 1:25
s_n_baru <- sapply(n, compute_s_n)

s_n_baru
```

```
## [1] 1 5 14 30 55 91 140 204 285 385 506 650 819 1015 1240
## [16] 1496 1785 2109 2470 2870 3311 3795 4324 4900 5525
```