

Tidyverse dan Importing Data

Elisia Dwi Rahayu

2022-10-29

```
library(dslabs)
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(purrr)
library(dplyr)
data("murders")
```

Soal Nomor 1

Gunakan `as_tibble` untuk mengkonversi tabel dataset “US murders” dalam bentuk tibble dan simpan dalam objek baru bernama ‘murders_tibble’.

```
murders_tibble <- as_tibble(murders)
class(murders_tibble)

## [1] "tbl_df"      "tbl"        "data.frame"
```

Soal Nomor 2

Gunakan fungsi `group_by` untuk mengkonversi dataset “US murders” menjadi sebuah tibble yang dikelompokkan berdasarkan ‘region’.

```
murders %>% group_by(region)

## # A tibble: 51 x 5
## # Groups:   region [4]
##   state      abb region population total
##   <chr>      <chr> <fct>      <dbl> <dbl>
## 1 Alabama    AL    South      4779736 135
```

```
## 2 Alaska AK West 710231 19
## 3 Arizona AZ West 6392017 232
## 4 Arkansas AR South 2915918 93
## 5 California CA West 37253956 1257
## 6 Colorado CO West 5029196 65
## 7 Connecticut CT Northeast 3574097 97
## 8 Delaware DE South 897934 38
## 9 District of Columbia DC South 601723 99
## 10 Florida FL South 19687653 669
## # ... with 41 more rows
```

Soal Nomor 3

Tulis script tidyverse yang menghasilkan output yang sama dengan perintah berikut:

```
exp(mean(log(murders$population)))
```

```
## [1] 3675209
```

Gunakan operator pipe sehingga setiap fungsi dapat dipanggil tanpa menambahkan argumen.

```
murders$population %>%
  log %>%
  mean %>%
  exp
```

```
## [1] 3675209
```

Soal Nomor 4

Gunakan map_df untuk membuat data frame yang terdiri dari tiga kolom: 'n', 's_n', dan 's_n_2'. Kolom pertama harus berisi angka 1 hingga 100. Kolom kedua dan ketiga masing-masing harus berisi penjumlahan 1 hingga n, dimana n menyatakan jumlah baris.

```
x <- 100

compute_s_n <- function(x) {
  angka <- 1:x
  tibble (
    n = length(angka),
    s_n = sum(angka),
    s_n_2 = sum(angka)
  )
}

n <- 100

hasil <- map_df(n, compute_s_n)

hasil
```

```
## # A tibble: 1 x 3
##       n     s_n s_n_2
##   <int> <int> <int>
## 1    100  5050  5050
```