

Tugas7

DBRT

12/22/2020

```
library(tibble)
library(dslabs)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(purrr)
data(murders)
```

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

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

## tibble [51 x 5] (S3: tbl_df/tbl/data.frame)
## $ state      : chr [1:51] "Alabama" "Alaska" "Arizona" "Arkansas" ...
## $ abb        : chr [1:51] "AL" "AK" "AZ" "AR" ...
## $ region     : Factor w/ 4 levels "Northeast","South",...: 2 4 4 2 4 4 1 2 2 2 ...
## $ population: num [1:51] 4779736 710231 6392017 2915918 37253956 ...
## $ total      : num [1:51] 135 19 232 93 1257 ...
```

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
```

3. Tulis *script tidyverse* yang menghasilkan *output* yang sama dengan perintah berikut :

```
exp(mean(log(murderspopulation)))
```

Gunakan operator *pipe* sehingga setiap fungsi dapat dipanggil tanpa menambahkan argumen. Gunakan *dot operator* untuk mengakses populasi

```
murders %>%
  .$population %>%
  log() %>%
  mean() %>%
  exp()
```

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

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

```
n <- 1:100
f_s <- function(n){
  x <- 1:n
  tibble(sum = sum(x))}
s <- n
s_n <- map_df(n, f_s)
s_n_2 <- map_df(n, f_s)
df <- data.frame(s,s_n,s_n_2)
df %>%
  rename(s_n = sum , s_n_2 = sum.1)
```

```
##      s  s_n s_n_2
## 1    1    1     1
## 2    2    3     3
## 3    3    6     6
## 4    4   10    10
## 5    5   15    15
```

## 6	6	21	21
## 7	7	28	28
## 8	8	36	36
## 9	9	45	45
## 10	10	55	55
## 11	11	66	66
## 12	12	78	78
## 13	13	91	91
## 14	14	105	105
## 15	15	120	120
## 16	16	136	136
## 17	17	153	153
## 18	18	171	171
## 19	19	190	190
## 20	20	210	210
## 21	21	231	231
## 22	22	253	253
## 23	23	276	276
## 24	24	300	300
## 25	25	325	325
## 26	26	351	351
## 27	27	378	378
## 28	28	406	406
## 29	29	435	435
## 30	30	465	465
## 31	31	496	496
## 32	32	528	528
## 33	33	561	561
## 34	34	595	595
## 35	35	630	630
## 36	36	666	666
## 37	37	703	703
## 38	38	741	741
## 39	39	780	780
## 40	40	820	820
## 41	41	861	861
## 42	42	903	903
## 43	43	946	946
## 44	44	990	990
## 45	45	1035	1035
## 46	46	1081	1081
## 47	47	1128	1128
## 48	48	1176	1176
## 49	49	1225	1225
## 50	50	1275	1275
## 51	51	1326	1326
## 52	52	1378	1378
## 53	53	1431	1431
## 54	54	1485	1485
## 55	55	1540	1540
## 56	56	1596	1596
## 57	57	1653	1653
## 58	58	1711	1711
## 59	59	1770	1770

##	60	60	1830	1830
##	61	61	1891	1891
##	62	62	1953	1953
##	63	63	2016	2016
##	64	64	2080	2080
##	65	65	2145	2145
##	66	66	2211	2211
##	67	67	2278	2278
##	68	68	2346	2346
##	69	69	2415	2415
##	70	70	2485	2485
##	71	71	2556	2556
##	72	72	2628	2628
##	73	73	2701	2701
##	74	74	2775	2775
##	75	75	2850	2850
##	76	76	2926	2926
##	77	77	3003	3003
##	78	78	3081	3081
##	79	79	3160	3160
##	80	80	3240	3240
##	81	81	3321	3321
##	82	82	3403	3403
##	83	83	3486	3486
##	84	84	3570	3570
##	85	85	3655	3655
##	86	86	3741	3741
##	87	87	3828	3828
##	88	88	3916	3916
##	89	89	4005	4005
##	90	90	4095	4095
##	91	91	4186	4186
##	92	92	4278	4278
##	93	93	4371	4371
##	94	94	4465	4465
##	95	95	4560	4560
##	96	96	4656	4656
##	97	97	4753	4753
##	98	98	4851	4851
##	99	99	4950	4950
##	100	100	5050	5050