

Latihan Modul 4

Muhammad Fawwaz

2022-09-18

```
import dataset "murders";
```

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

1. Gunakan operator aksesori (\$) untuk mengakses variabel populasi dan menyimpannya pada objek baru "pop". Kemudian gunakan fungsi sort untuk mengurutkan variabel "pop". Pada langkah terakhir, gunakan operator [] untuk menampilkan nilai populasi terkecil.

```
pop <- murders$population
sorted_pop <- sort(pop)
sorted_pop

## [1] 563626 601723 625741 672591 710231 814180 897934
989415
## [9] 1052567 1316470 1328361 1360301 1567582 1826341 1852994
2059179
## [17] 2700551 2763885 2853118 2915918 2967297 3046355 3574097
3751351
## [25] 3831074 4339367 4533372 4625364 4779736 5029196 5303925
5686986
## [33] 5773552 5988927 6346105 6392017 6483802 6547629 6724540
8001024
## [41] 8791894 9535483 9883640 9920000 11536504 12702379 12830632
19378102
## [49] 19687653 25145561 37253956

sorted_pop[which.min(sorted_pop)]

## [1] 563626
```

2. Tampilkan indeks dari data yang memiliki nilai populasi terkecil. Petunjuk: gunakan fungsi order.

```
new_pop <- order(murders$population)
head(new_pop, 1)

## [1] 51
```

3. Dengan fungsi which.min, Tulis satu baris kode yang dapat menampilkan hasil yang sama dengan langkah diatas.

```
pop[which.min(order(pop, decreasing = TRUE))]
```

```
## [1] 9883640
```

4. Tampilkan nama negara yang memiliki populasi terkecil.

```
state_pop <- order(murders$population)
state <- murders$state[state_pop]
head(state, 1)
```

```
## [1] "Wyoming"
```

5. Untuk membuat data frame baru, contoh script yang dapat digunakan adalah sebagai berikut: Gunakan fungsi rank untuk menentukan peringkat populasi dari tiap negara bagian, dimulai dari nilai terkecil hingga terbesar. Simpan hasil pemeringkatan di objek baru "ranks", lalu buat data frame baru yang berisi nama negara bagian dan peringkatnya dengan nama "my_df".

```
ranks <- rank(murders$population)
my_df <- data.frame(state = murders$state, rank = ranks)
my_df
```

```
##           state rank
## 1      Alabama   29
## 2       Alaska    5
## 3     Arizona   36
## 4    Arkansas   20
## 5   California   51
## 6     Colorado   30
## 7   Connecticut   23
## 8     Delaware    7
## 9 District of Columbia  2
## 10      Florida   49
## 11      Georgia   44
## 12       Hawaii   12
## 13       Idaho   13
## 14     Illinois   47
## 15     Indiana   37
## 16       Iowa   22
## 17      Kansas   19
## 18     Kentucky   26
## 19    Louisiana   27
## 20       Maine   11
## 21     Maryland   33
## 22 Massachusetts   38
## 23      Michigan   43
## 24     Minnesota   31
## 25    Mississippi   21
## 26      Missouri   34
## 27      Montana    8
## 28     Nebraska   14
## 29      Nevada   17
## 30   New Hampshire   10
## 31    New Jersey   41
```

```
## 32      New Mexico  16
## 33      New York   48
## 34    North Carolina 42
## 35    North Dakota   4
## 36      Ohio        45
## 37    Oklahoma      24
## 38    Oregon        25
## 39    Pennsylvania  46
## 40    Rhode Island   9
## 41    South Carolina 28
## 42    South Dakota   6
## 43    Tennessee     35
## 44    Texas         50
## 45    Utah          18
## 46    Vermont        3
## 47    Virginia      40
## 48    Washington    39
## 49    West Virginia  15
## 50    Wisconsin     32
## 51    Wyoming       1
```

6. Ulangi langkah sebelumnya, namun kali ini urutkan `my_df` dengan fungsi `order` agar data yang ditampilkan merupakan data yang telah diurutkan dari populasi yang paling tidak padat hingga ke yang terpadat. Petunjuk: buat objek "ind" yang akan menyimpan indeks yang diperlukan dalam mengurutkan data populasi

```
ind <- my_df[order(my_df$rank),]
ind

##           state rank
## 51      Wyoming    1
## 9  District of Columbia 2
## 46      Vermont    3
## 35    North Dakota   4
## 2       Alaska      5
## 42    South Dakota   6
## 8       Delaware    7
## 27      Montana     8
## 40    Rhode Island   9
## 30    New Hampshire 10
## 20      Maine       11
## 12      Hawaii      12
## 13      Idaho       13
## 28      Nebraska    14
## 49    West Virginia 15
## 32      New Mexico  16
## 29      Nevada      17
## 45      Utah        18
## 17      Kansas      19
## 4       Arkansas    20
## 25    Mississippi  21
```

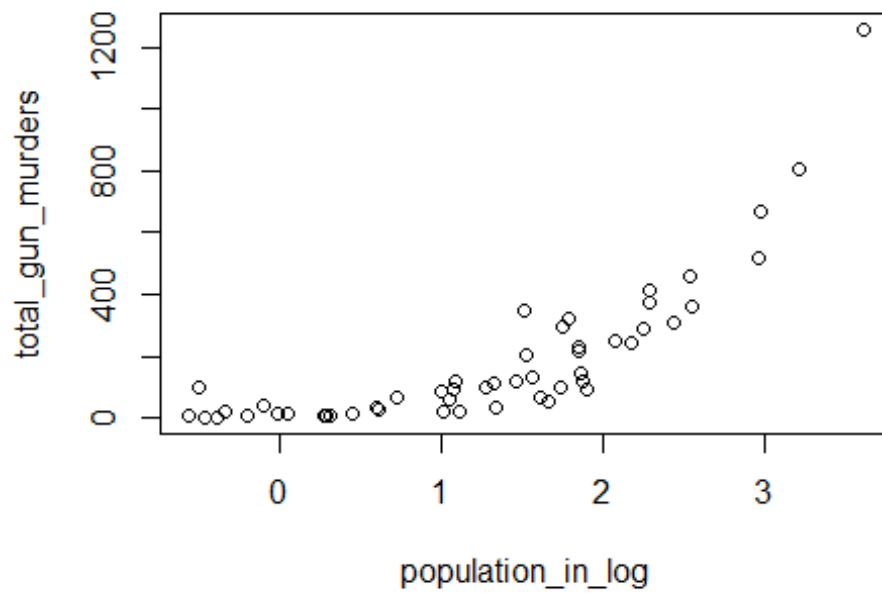
## 16	Iowa	22
## 7	Connecticut	23
## 37	Oklahoma	24
## 38	Oregon	25
## 18	Kentucky	26
## 19	Louisiana	27
## 41	South Carolina	28
## 1	Alabama	29
## 6	Colorado	30
## 24	Minnesota	31
## 50	Wisconsin	32
## 21	Maryland	33
## 26	Missouri	34
## 43	Tennessee	35
## 3	Arizona	36
## 15	Indiana	37
## 22	Massachusetts	38
## 48	Washington	39
## 47	Virginia	40
## 31	New Jersey	41
## 34	North Carolina	42
## 23	Michigan	43
## 11	Georgia	44
## 36	Ohio	45
## 39	Pennsylvania	46
## 14	Illinois	47
## 33	New York	48
## 10	Florida	49
## 44	Texas	50
## 5	California	51

- Untuk keperluan analisis data, akan dibuat plot yang memvisualisasikan total pembunuhan terhadap populasi dan mengidentifikasi hubungan antara keduanya. Script yang digunakan: Perlu diingat bahwa beberapa negara bagian memiliki populasi di bawah 5 juta, sehingga untuk mempermudah analisis, buat plot dalam skala log. Transformasi nilai variabel menggunakan transformasi log10, kemudian tampilkan plot-nya.

```

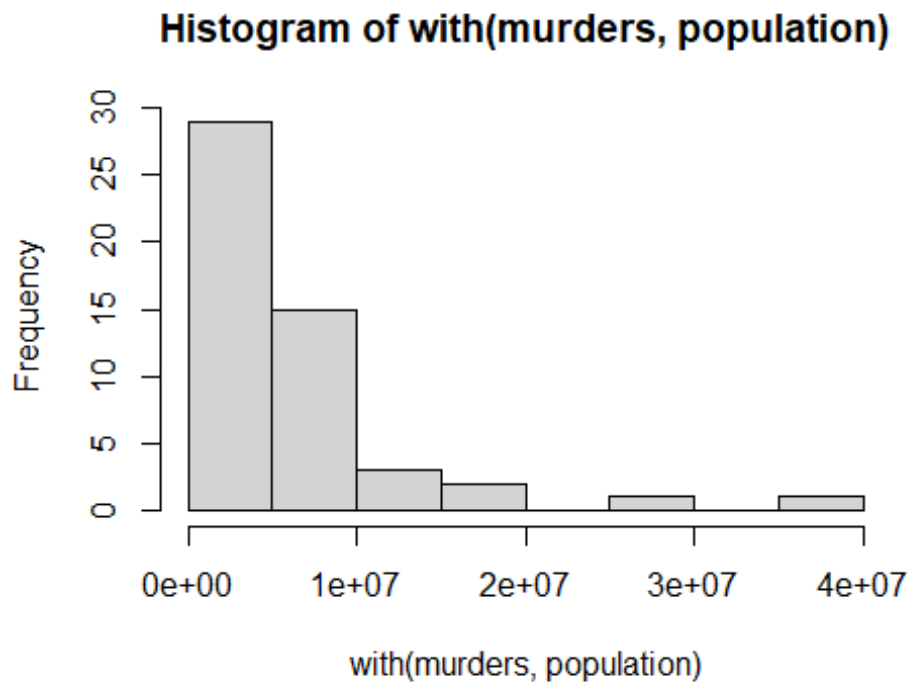
population_in_millions <- murders$population/10^6
population_in_log <- log(population_in_millions)
total_gun_murders <- murders$total
plot(population_in_log, total_gun_murders)

```



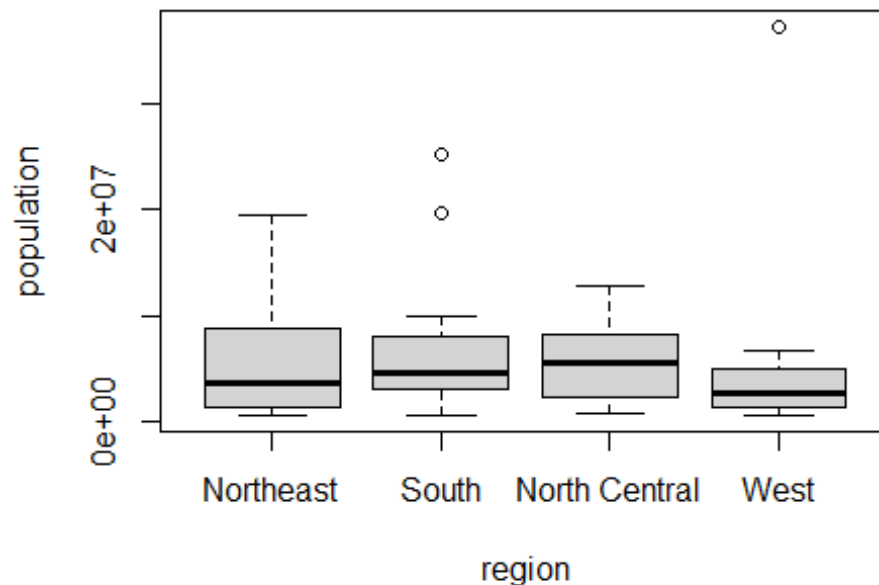
8. Buat histogram dari populasi negara bagian.

```
hist(with(murders, population))
```



9. Hasilkan boxplot dari populasi negara bagian berdasarkan wilayahnya.

```
boxplot(population~region, data = murders)
```



R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

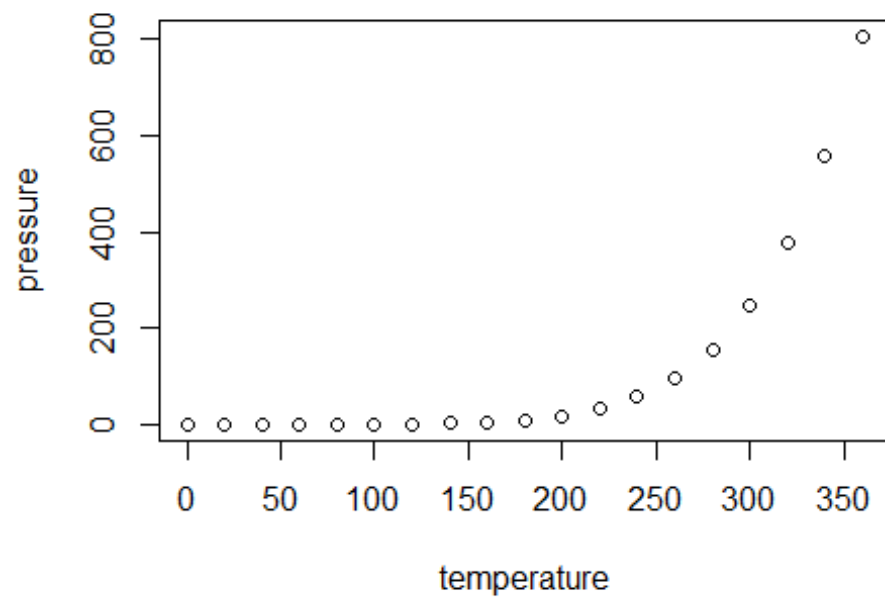
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   :  2.00
##  1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##  Mean   :15.4    Mean    : 42.98
##  3rd Qu.:19.0    3rd Qu.: 56.00
##  Max.   :25.0    Max.    :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.