

Latihan3_123190046

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.

```
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
sort(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
```

```
sort(pop)[1]
```

```
## [1] 563626
```

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

```
order(pop)[1]
```

```
## [1] 51
```

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

```
which.min(pop)
```

```
## [1] 51
```

4. Tampilkan nama negara yang memiliki populasi terkecil

```
ind <- order(murders$population)
murders$state[ind][1]
```

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

5. 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(pop)
city = murders$state
my_df = data.frame(name = city, rank = ranks)
my_df
```

```
##           name 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

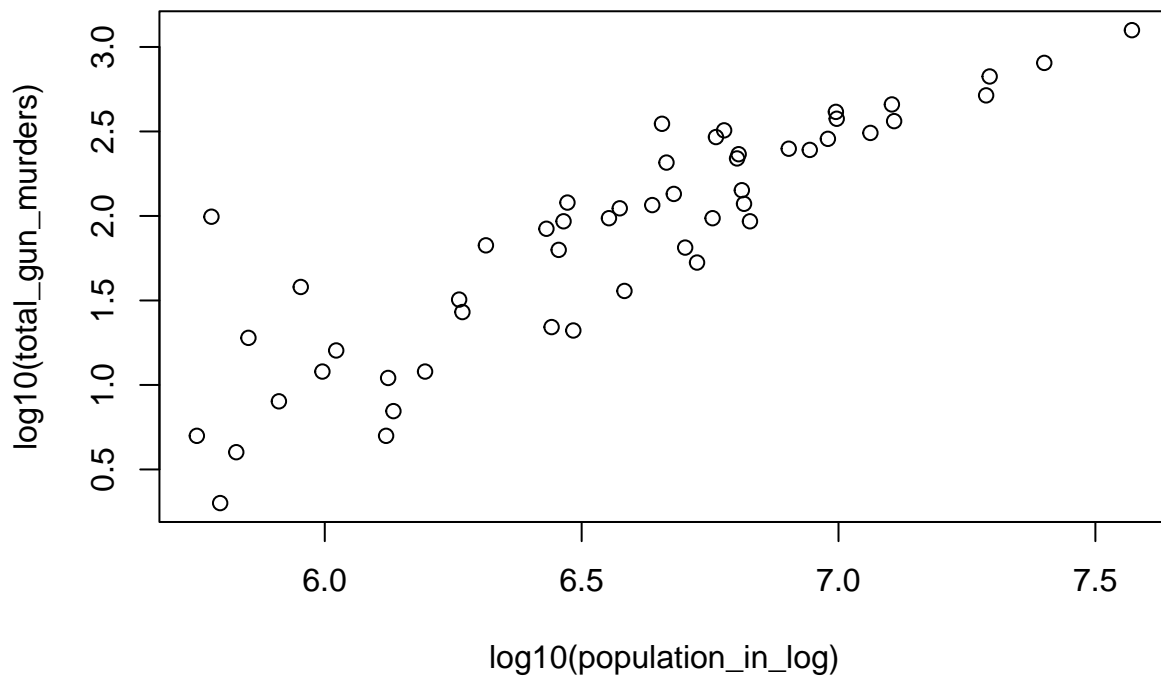
```
ranks = rank(pop)
city = murders$state
my_df = data.frame(name = city, rank = ranks)
ind = order(my_df$rank)
my_df = data.frame(nama = my_df$name[ind], rank = my_df$rank[ind])
my_df
```

```
##          nama rank
## 1      Wyoming    1
## 2 District of Columbia 2
## 3      Vermont    3
## 4    North Dakota    4
## 5        Alaska    5
## 6    South Dakota    6
## 7      Delaware    7
## 8        Montana    8
## 9    Rhode Island    9
## 10   New Hampshire   10
## 11        Maine     11
## 12        Hawaii    12
## 13        Idaho     13
## 14       Nebraska    14
## 15   West Virginia   15
## 16    New Mexico     16
## 17        Nevada     17
## 18        Utah      18
## 19        Kansas     19
## 20       Arkansas     20
## 21    Mississippi     21
## 22        Iowa      22
## 23    Connecticut     23
## 24       Oklahoma     24
## 25        Oregon     25
## 26       Kentucky     26
## 27       Louisiana     27
## 28   South Carolina     28
## 29        Alabama     29
## 30       Colorado     30
## 31       Minnesota     31
## 32       Wisconsin     32
## 33       Maryland     33
## 34       Missouri     34
## 35       Tennessee     35
## 36        Arizona     36
## 37       Indiana     37
## 38   Massachusetts     38
```

```
## 39      Washington 39
## 40      Virginia 40
## 41      New Jersey 41
## 42      North Carolina 42
## 43      Michigan 43
## 44      Georgia 44
## 45      Ohio 45
## 46      Pennsylvania 46
## 47      Illinois 47
## 48      New York 48
## 49      Florida 49
## 50      Texas 50
## 51      California 51
```

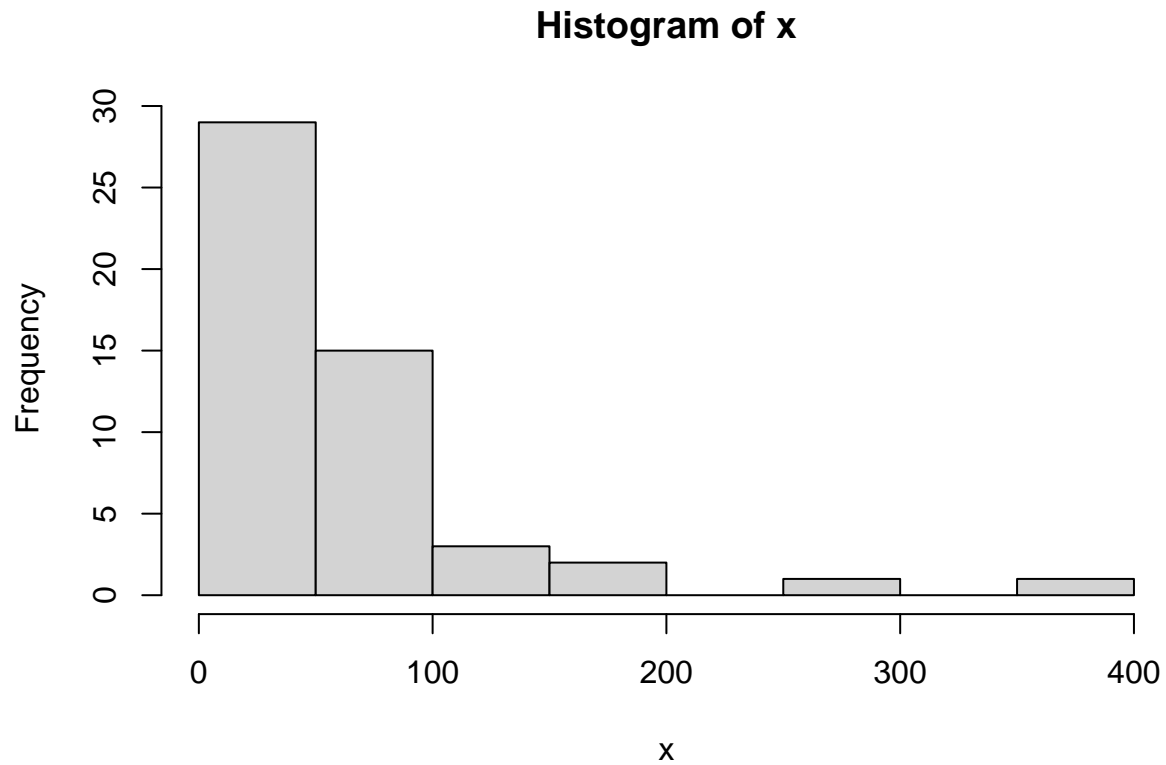
7. buat plot dalam skala log. Transformasi nilai variabel menggunakan transformasi log10, kemudian tampilkan plot-nya.

```
population_in_log <- murders$population
total_gun_murders <- murders$total
plot(log10(population_in_log), log10(total_gun_murders))
```



8. Buat histogram dari populasi negara bagian.

```
x <- with(murders, population/100000)
hist(x)
```



```
murders$state[which.max(x)]
```

```
## [1] "California"
```

9. Hasilkan boxplot dari populasi negara bagian berdasarkan wilayahnya.

```
murders$population <- with(murders, population/100000)
boxplot(population~region, data = murders)
```

