MODUL 4
FUNGSI SORTING DAN DASAR VISUALISASI DATA

A. Tujuan Praktikum
- Memahami fungsi-fungsi yang berkaitan dengan proses *sorting* pada R
- Memahami fungsi-fungsi dasar visualisasi data pada R

B. Alokasi Waktu
1 x pertemuan = 120 menit

C. Dasar Teori
Setelah menguasai beberapa pengetahuan dasar R, dengan *dataset* yang sama, pada bagian ini kita akan mencoba beberapa fungsi lain yang umumnya digunakan dalam proses analisis data.

Fungsi *Sorting*:

1. `sort`
Pembahasan mengenai fungsi ini akan dilakukan menggunakan contoh studi kasus. Sebagai contoh, kita ingin menampilkan peringkat negara bagian (*state*) yang mengalami kejadian pembunuhan bersenjata dari jumlah kejadian yang paling sedikit hingga paling besar. Fungsi yang dibutuhkan diharapkan dapat mengurutkan vektor dalam urutan yang meningkat (dari data terkecil hingga terbesar). Langkah pertama, kita akan melihat jumlah pembunuhan bersenjata terbesar dengan menggunakan *script*:

```
library(dslabs)
data(murders)
sort(murders$total)
#> [1] 2 4 5 5 7 8 11 12 12 16 19 21 22
#> [14] 27 32 36 38 53 63 65 67 84 93 93 97 97
#> [27] 99 111 116 118 120 135 142 207 219 232 246 250 286
#> [40] 293 310 321 351 364 376 413 457 517 669 805 1257
```

Fungsi `sort` saja tidak dapat memberikan informasi mengenai negara bagian mana yang memiliki total pembunuhan terbesar. Kita tidak tahu negara bagian mana yang memiliki jumlah pembunuhan sebesar 1257.

2. `order`
Fungsi ini dapat menghasilkan *output* yang lebih sesuai dengan apa yang kita inginkan pada contoh kasus diatas. Fungsi `order` membutuhkan vektor sebagai input dan memanfaatkan *indexing* vektor untuk mengurutkan vector yang telah diinput. Agar lebih mudah dipahami, penjelasan mengenai fungsi `sort` dapat dilihat pada contoh sederhana dibawah ini. Pertama, kita akan membuat vektor yang disimpan pada variabel '*x*' dan kemudian mengurutkannya menggunakan dua fungsi yang berbeda, yaitu: `sort` dan `order` untuk dapat memahami perbedaan dari kedua fungsi tersebut:

```
x <- c(31, 4, 15, 92, 65)
sort(x)
#> [1] 4 15 31 65 92
```

Untuk menampilkan hasil pengurutan, fungsi `order` membutuhkan argumen indeks untuk mengurutkan vector yang diinput:

```
index <- order(x)
x[index]
#> [1] 4 15 31 65 92
```

Output yang ditampilkan oleh kedua fungsi akan bernilai sama, karena kedua fungsi dapat digunakan untuk mengurutkan data yang diinput. Perbedaan proses fungsi `sort` dan `order` dapat dilihat jika dilakukan analisa lebih lanjut dengan melihat indeks dari data yang diinputkan seperti yang telah ditampilkan pada *script* berikut:

```
x
#> [1] 31 4 15 92 65
order(x)
#> [1] 2 3 1 5 4
```

Data kedua pada variabel "*x*" adalah yang terkecil, sehingga `order(x)` dimulai dengan menampilkan indeks 2. Nilai terkecil berikutnya adalah data ketiga, dan indeks yang ditampilkan selanjutnya adalah 3 dan seterusnya. Setelah memahami mengenai penggunaan fungsi `sort`, selanjutnya akan kita gunakan fungsi tersebut untuk mengidentifikasi *state* yang memiliki jumlah pembunuhan bersenjata terbesar. Pertama, kita akan melakukan akses pada data vektor menggunakan operator aksesor ($) terhadap variabel *total* pada *dataset* "*murders*". Kemudian hasilnya akan disimpan pada variabel baru '*ind*'. Pada langkah terakhir, operator aksesor digunakan untuk menampilkan vektor singkatan nama negara (variabel *abb*) yang indeksnya sama dengan variabel *total*:

```
ind <- order(murders$total)
murders$abb[ind]
#> [1] "VT" "ND" "NH" "WY" "HI" "SD" "ME" "ID" "MT" "RI" "AK" "IA" "UT"
#> [14] "WV" "NE" "OR" "DE" "MN" "KS" "CO" "NM" "NV" "AR" "WA" "CT" "WI"
#> [27] "DC" "OK" "KY" "MA" "MS" "AL" "IN" "SC" "TN" "AZ" "NJ" "VA" "NC"
#> [40] "MD" "OH" "MO" "LA" "IL" "GA" "MI" "PA" "NY" "FL" "TX" "CA"
```

Sehingga, berdasarkan implementasi script diatas, dapat diidentifikasi bahwa California adalah nagara bagian yang frekuensi pembunuhan bersenjatanya terbanyak.

3. `max` dan `which.max`

Alternatif lain yang lebih mudah untuk menampilkan data pembunuhan terbanyak, yang berarti bahwa kita hanya akan mengidentifikasi data dengan nilai terbesar, dapat menggunakan fungsi `max`:

```
max(murders$total)
#> [1] 1257
```

dan `which.max` untuk menampilkan indeks data yang memiliki nilai terbesar:

```
i_max <- which.max(murders$total)
murders$state[i_max]
#> [1] "California"
```

Untuk identifikasi data yang bernilai minimum, kita dapat menggunakan fungsi `min` dan `which.min` dengan cara yang sama.

4. `rank`

Meskipun tidak sering digunakan sebagai fungsi pengurutan seperti `sort` dan `order`, fungsi `rank` juga memiliki langkah pengurutan yang mirip dengan `order`. Untuk setiap vektor yang diberikan, `rank` akan menampilkan peringkat dari data pertama, data kedua, dan seterusnya. Contoh sederhananya adalah:

```
x <- c(31, 4, 15, 92, 65)
rank(x)
#> [1] 3 1 2 5 4
```

Dari seluruh fungsi pengurutan (*sorting*) yang telah dibahas diatas, sebagai ringkasan, perbedaan *output* / hasil dari implementasi fungsi `sort`, `order`, dan `rank` dapat dilihat pada tebel berikut:

original	sort	order	rank
31	4	2	3
4	15	3	1
15	31	1	2
92	65	5	5
65	92	4	4

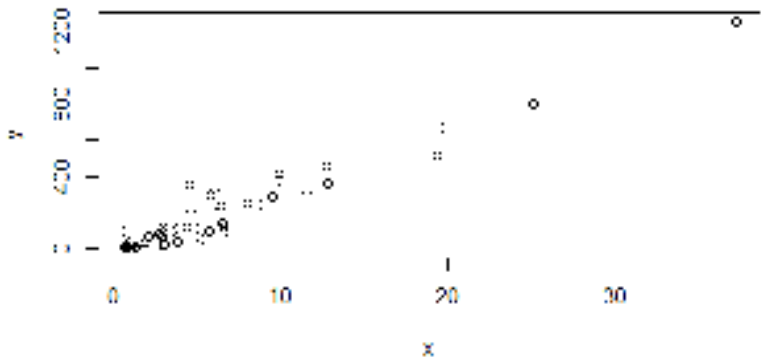
Dasar Visualisasi Data

Pada modul 8, kita akan membahas lebih rinci mengenai penggunaan paket *add-on* yang dapat memberikan kemudahan visualisasi data pada R. Pada modul ini, akan dijelaskan secara singkat beberapa fungsi visualisasi data (plot) yang disediakan di instalasi dasar R.

1. `plot`

Fungsi `plot` dapat digunakan untuk membuat visualisasi data dalam bentuk sebaran titik. Berdasarkan *dataset* "*murders*" yang kita miliki, untuk menampilkan *plot* total pembunuhan versus populasi, *script*-nya adalah:

```
x <- murders$population / 10^6
y <- murders$total
plot(x, y)
```



Alternatif lain, dapat digunakan fungsi `with` untuk menampilkan *plot* dengan *script* yang lebih mudah dan menghindari pengaksesan variabel yang sama:

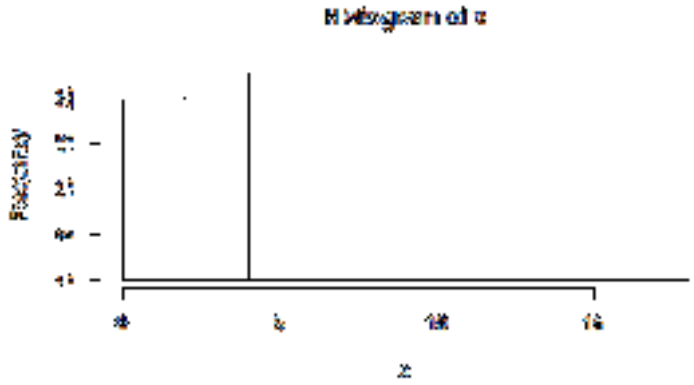
```
with(murders, plot(population, total))
```

Fungsi `with` juga dapat digunakan pada *data frame* apapun dan bersama dengan fungsi apapun.

2. `hist`

Histogram merupakan visualisasi data yang cukup baik untuk mengidentifikasi daftar nilai yang sering muncul pada data yang dievaluasi. Untuk menampilkan histogram yang memuat '*murder_rate*' dari keseluruhan *dataset*, script yang digunakan:

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



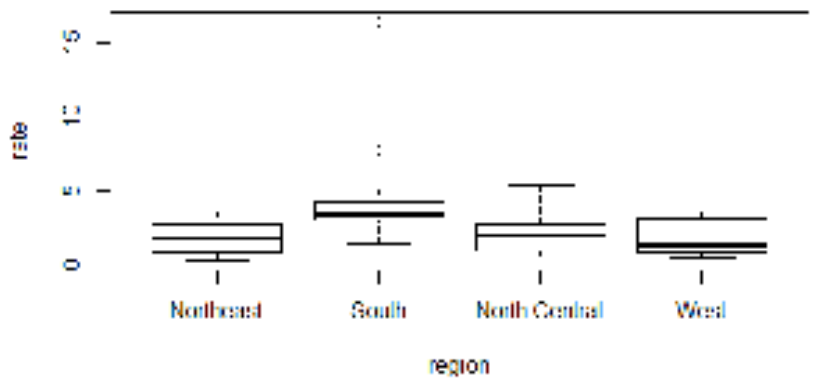
Melalui histogram diatas, dapat dievaluasi bahwa sebagian besar "*murder_rate*" memiliki frekuensi antara 2 hingga 3. Selain itu, dapat disimpulkan pula bahwa terdapat satu kasus yang nilai '*murder_rate*'-nya sangat ekstrim, yaitu lebih dari 15. Untuk mengidentifikasi *state* mana yang memiliki nilai '*murder_rate*' tertinggi, dapat dilakukan dengan:

```
murders$state[which.max(x)]
#> [1] "District of Columbia"
```

3. `boxplot`

`Boxplots` dapat memberikan ringkasan visualisasi yang lebih singkat daripada histogram. Sebagai contoh, di sini kita dapat menggunakan `boxplot` untuk membandingkan '*murder_rate*' dari berbagai *region*:

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

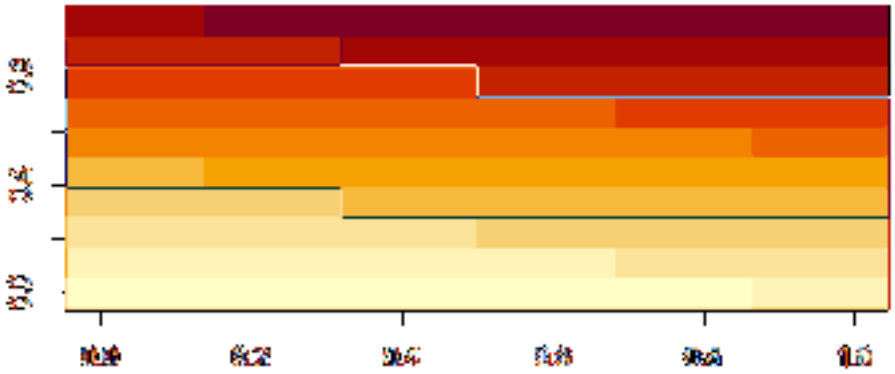


Melalui visualisasi diatas, dapat disimpulkan bahwa *South region* memiliki tingkat pembunuhan yang lebih tinggi daripada tiga wilayah lainnya.

4. `Image`

Fungsi `image` dapat digunakan untuk menampilkan nilai-nilai matriks dalam bentuk rentang warna. Contohnya:

```
x <- matrix(1:120, 12, 10)
image(x)
```



D. Latihan

Pada latihan ini seluruh soal akan akan menggunakan *dataset*: AS *murders*.

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

1. Gunakan operator aksesor ($) 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.
2. Tampilkan indeks dari data yang memiliki nilai populasi terkecil.
   Petunjuk: gunakan fungsi `order`.
3. Dengan fungsi `which.min`, Tulis satu baris kode yang dapat menampilkan hasil yang sama dengan langkah diatas.
4. Tampilkan nama negara yang memiliki populasi terkecil.
5. Untuk membuat *data frame* baru, contoh *script* yang dapat digunakan adalah sebagai berikut:

   ```
   temp <- c(35, 88, 42, 84, 81, 30)
   city <- c("Beijing", "Lagos", "Paris", "Rio de Janeiro",
   "San Juan", "Toronto")
   city_temps <- data.frame(name = city, temperature = temp)
   ```

   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*".
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
7. Untuk keperluan analisis data, akan dibuat plot yang memvisualisasikan total pembunuhan terhadap populasi dan mengidentifikasi hubungan antara keduanya. *Script* yang digunakan:

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

   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.
8. Buat histogram dari populasi negara bagian.
9. Hasilkan *boxplot* dari populasi negara bagian berdasarkan wilayahnya.