

# Pertemuan 2 - Sorting dan Data Visualization

plirapli

2024-09-08

## Sorting dan Data Visualization

### Sorting

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

Nampilin dataset murder

```
# View(murders) # <- Pas mau diknit di-comment dulu kodenya
```

Nampilin hasil total murders

```
murders$total
```

```
## [1] 135 19 232 93 1257 65 97 38 99 669 376 7 12 364 142
## [16] 21 63 116 351 11 293 118 413 53 120 321 12 32 84 5
## [31] 246 67 517 286 4 310 111 36 457 16 207 8 219 805 22
## [46] 2 250 93 27 97 5
```

### Fungsi sort(), order(), dan rank()

**Fungsi sort()** Fungsi sort() digunakan untuk mengurutkan nilai-nilai dalam vektor dari yang terkecil hingga terbesar (ascending) atau dari terbesar hingga terkecil (descending). Contohnya di sini kita bakalan nyoba buat ngurutin total kasus pembunuhan dari yg terkecil dan juga sebaliknya.

```
urutan_terkecil = sort(murders$total)
urutan_terbesar = sort(murders$total, decreasing = TRUE)
```

```
urutan_terkecil
```

```
## [1] 2 4 5 5 7 8 11 12 12 16 19 21 22 27 32
## [16] 36 38 53 63 65 67 84 93 93 97 97 99 111 116 118
## [31] 120 135 142 207 219 232 246 250 286 293 310 321 351 364 376
## [46] 413 457 517 669 805 1257
```

```
urutan_terbesar
```

```
## [1] 1257 805 669 517 457 413 376 364 351 321 310 293 286 250 246
## [16] 232 219 207 142 135 120 118 116 111 99 97 97 93 93 84
## [31] 67 65 63 53 38 36 32 27 22 21 19 16 12 12 11
## [46] 8 7 5 5 4 2
```

**Fungsi order()** Fungsi `order()` bakalan ngasih *index* yang akan mengurutkan elemen vektor. Artinya, dia ngasih tau urutan *index elemen* yang akan menghasilkan urutan yang benar. Kata kuncinya di *INDEX ELEMEN*. Kalo `sort` itu ngasih nilai/value, sedangkan kalo `order` itu ngasih *URUTAN INDEX ELEMEN*. Ingat, `index` di R itu *dimulai dari 1*.

```
index_urutan_kasus_pembunuhan_terkecil = order(murders$total)
index_urutan_kasus_pembunuhan_terbesar = order(murders$total, decreasing = TRUE)

index_urutan_kasus_pembunuhan_terkecil
```

```
## [1] 46 35 30 51 12 42 20 13 27 40 2 16 45 49 28 38 8 24 17 6 32 29 4 48 7
## [26] 50 9 37 18 22 25 1 15 41 43 3 31 47 34 21 36 26 19 14 11 23 39 33 10 44
## [51] 5
```

```
index_urutan_kasus_pembunuhan_terbesar
```

```
## [1] 5 44 10 33 39 23 11 14 19 26 36 21 34 47 31 3 43 41 15 1 25 22 18 37 9
## [26] 7 50 4 48 29 32 6 17 24 8 38 28 49 45 16 2 40 13 27 20 42 12 30 51 35
## [51] 46
```

Penjelasan: - Pada “`index_urutan_kasus_pembunuhan_terkecil`”, urutan pertama kasus pembunuhan terbanyak diduduki oleh `index 46`, juara kedua `index 35`, juara ketiga `index 30`, dsb. - Pada “`index_urutan_kasus_pembunuhan_terbesar`”, urutan pertama kasus pembunuhan terkecil diduduki oleh `index 5`, juara kedua `index 44`, juara ketiga `index 10`, dsb.

**Fungsi rank()** Fungsi `rank()` memberikan *ranking (peringkat)* untuk setiap elemen dalam vektor berdasarkan urutan nilainya.

```
x <- c(11, 26, 56, 99, 8)
rank(x)
```

```
## [1] 2 3 4 5 1
```

```
rank(-x)
```

```
## [1] 4 3 2 1 5
```

Penjelasan: - Pada `rank` dari yg terkecil ke terbesar, Angka 11 berada di urutan ke-2, angka 26 di urutan ke-3, angka 56 di urutan 4, dst. - Pada `rank` dari yg terbesar ke terkecil, Angka 11 berada di urutan ke-4, angka 26 di urutan ke-3, angka 56 di urutan 2, dst.

## Fungsi max() dan which.max()

Fungsi max() digunakan untuk mengidentifikasi data dengan nilai terbesar, sedangkan which.max digunakan untuk menampilkan *index* data yang memiliki nilai terbesar. Kebalikan dari max() dan which.max() adalah min() dan which.min()

```
highest = max(murders$total)
lowest = min(murders$total)

index_max = which.max(murders$total)
index_min = which.min(murders$total)

# menampilkan angka pembunuhan terbesar dan terkecil
highest
```

```
## [1] 1257
```

```
lowest
```

```
## [1] 2
```

```
# menampilkan wilayah dengan total kasus pembunuhan terbesar dan terkecil
murders$state[index_max]
```

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

```
murders$state[index_min]
```

```
## [1] "Vermont"
```

## Data Visualization

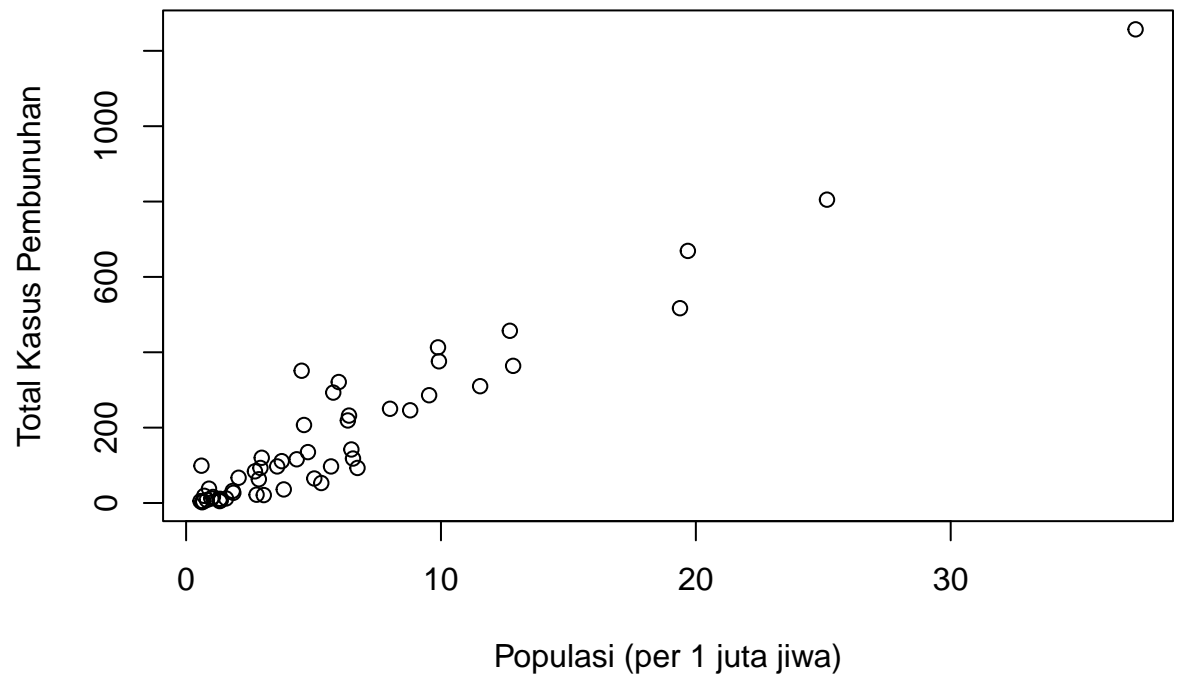
### Plot

Fungsi plot() di R adalah fungsi dasar yang digunakan untuk membuat berbagai jenis grafik atau plot. Dengan plot(), kita bisa bikin scatter plot, line plot, dan berbagai jenis grafik lainnya tergantung pada data yang diberikan.

```
# Scatter Plot (Total kasus pembunuhan pada tiap populasi)
populasi = murders$population / 10^6
total_pembunuhan = murders$total

plot(
  populasi,
  total_pembunuhan,
  main = "Total kasus pembunuhan pada tiap populasi",
  xlab = "Populasi (per 1 juta jiwa)",
  ylab = "Total Kasus Pembunuhan"
)
```

## Total kasus pembunuhan pada tiap populasi



### Scatter Plot

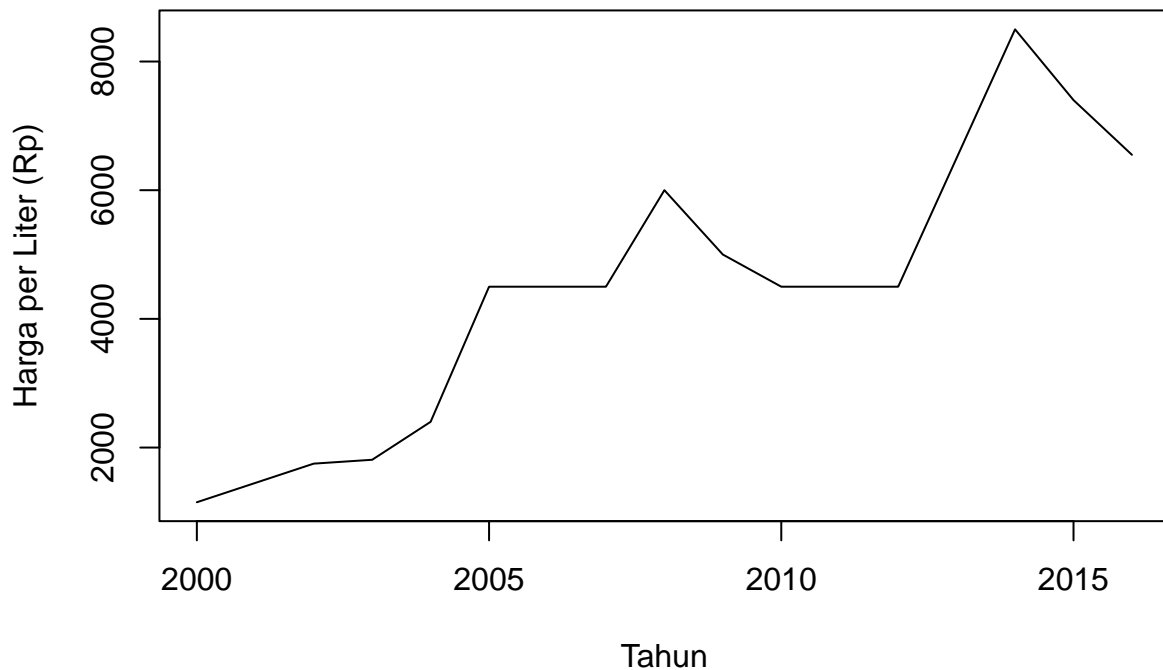
**Line Plot** Line plot biasanya dipakai untuk data timeseries. Salah satu contohnya adalah grafik kenaikan harga BBM. Pada praktikum ini, kita akan mencoba membuat sebuah line plot untuk menampilkan data grafik kenaikan harga BBM (premium) dari tahun 2000 s/d 2016.

Catatan: data diambil dari <https://databoks.katadata.co.id/datapublish/2016/08/05/harga-premium-di-indonesia-1970-2015>

```
data_bbm = data.frame(
  tahun = c(2000:2016),
  harga = c(
    1150, 1450, 1750, 1810, 2400,
    4500, 4500, 4500, 6000, 5000,
    4500, 4500, 4500, 6500, 8500,
    7400, 6550
  )
)

plot(
  data_bbm$tahun,
  data_bbm$harga,
  main = "Harga Premium di Indonesia 2000-2016",
  xlab = "Tahun",
  ylab = "Harga per Liter (Rp)",
  type = "l"
)
```

## Harga Premium di Indonesia 2000–2016

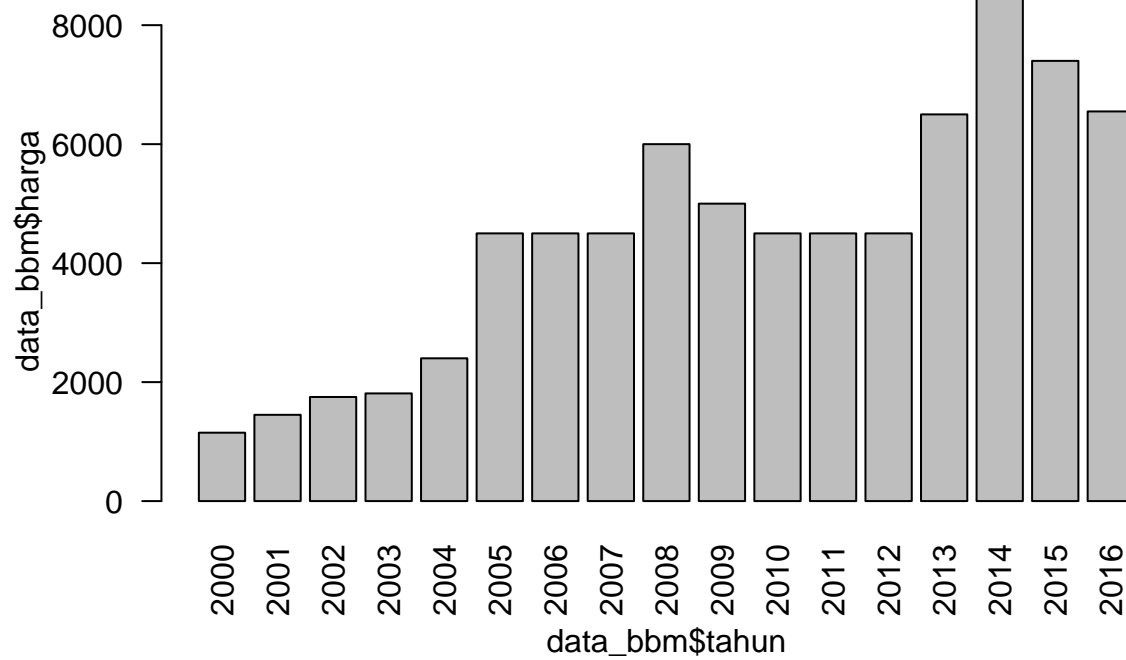


# type-nya ada p (scatter), l (line), s (stair), h (histogram), b (s dan l), o (kaya b tapi nyambung ga

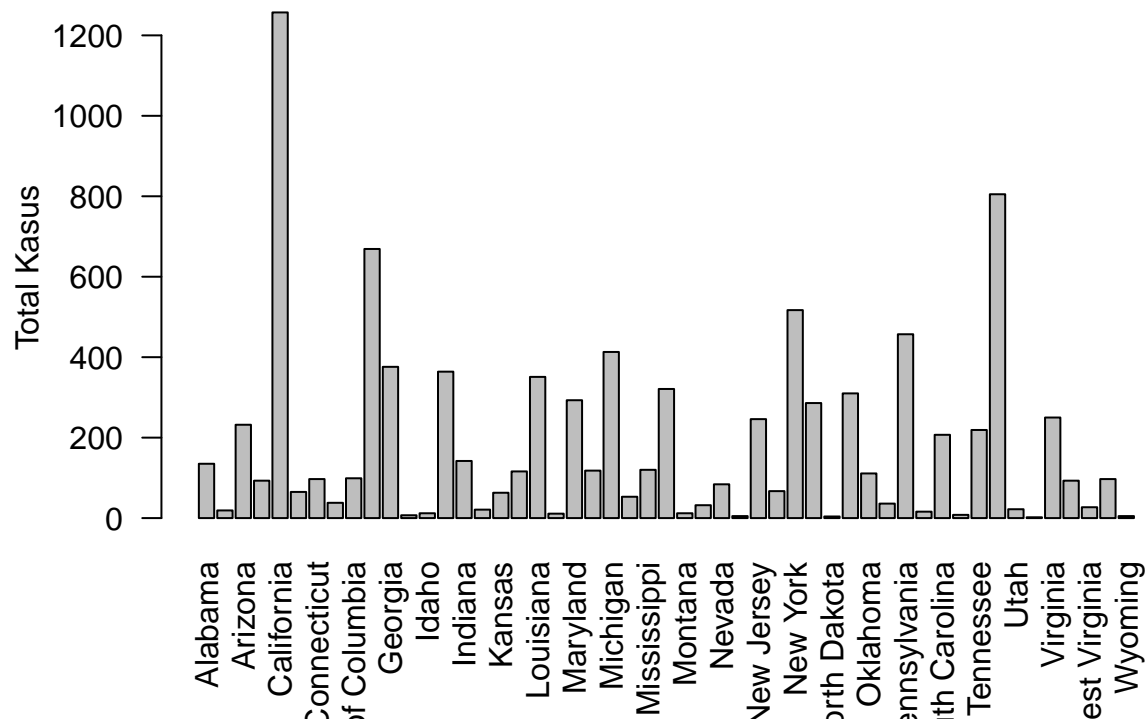
### Barplot

Selain pake plot, kita juga bisa make fungsi barplot() buat bikin plot yang bentukannya kek batang. Berikut adalah contoh untuk membuat grafik BBM menggunakan barplot(). Pastiin temen-temen udah bikin data\_bbm.

```
# Dataset bbm
barplot(
  data_bbm$harga ~ data_bbm$tahun,
  las = 2 # <- buat nge-rotate x label biar semua labelnya keliatan
)
```



```
# Dataset murders
barplot(
  murders$total ~ murders$state,
  las = 2, # <- buat nge-rotate x label biar semua labelnya keliatan
  xlab = "",
  ylab = "Total Kasus"
)
```



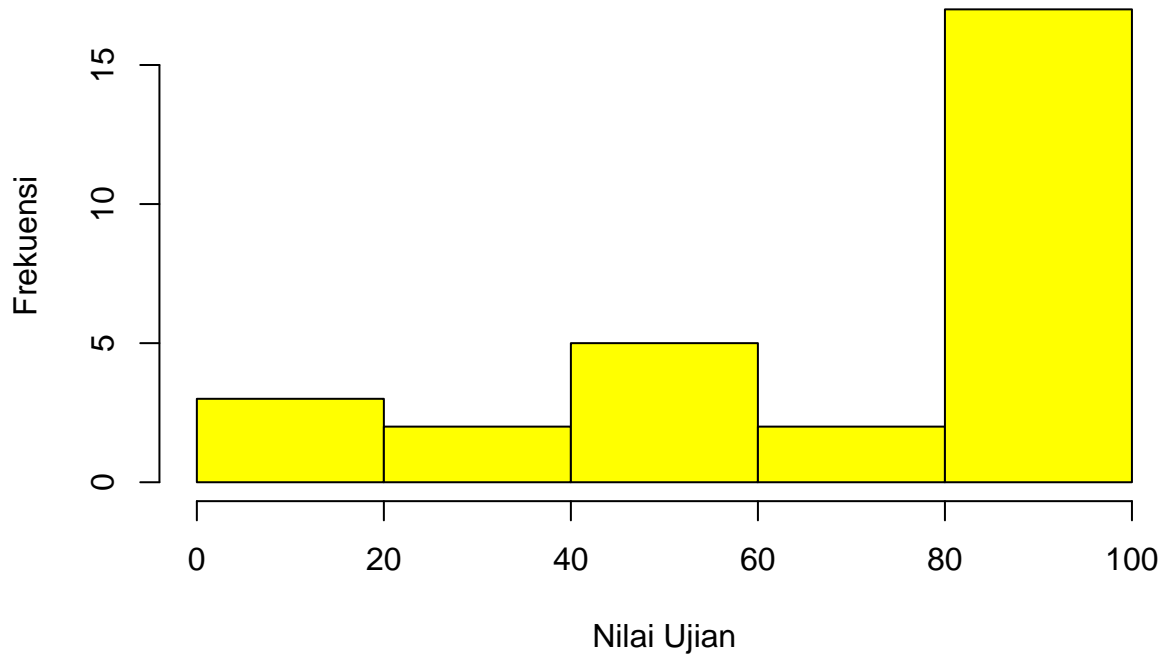
## Histogram

Histogram adalah grafik yang digunakan untuk menampilkan distribusi frekuensi dari suatu dataset.

```
# Tampilkan data (vector) yang mau dihitung frekuensinya dalam bentuk histogram pake fungsi hist(). Dalam
nilai_ujian = c(
  100, 100, 100, 0, 85, 0, 100, 95, 90, 100,
  90, 0, 100, 75, 100, 85, 88, 90, 88, 100,
  60, 75, 85, 45, 40, 45, 40, 50, 60
)

hist(
  nilai_ujian,
  main = "Frekuensi Nilai Ujian",
  xlab = "Nilai Ujian",
  ylab = "Frekuensi",
  col = "yellow",
  border = "black",
)
```

## Frekuensi Nilai Ujian



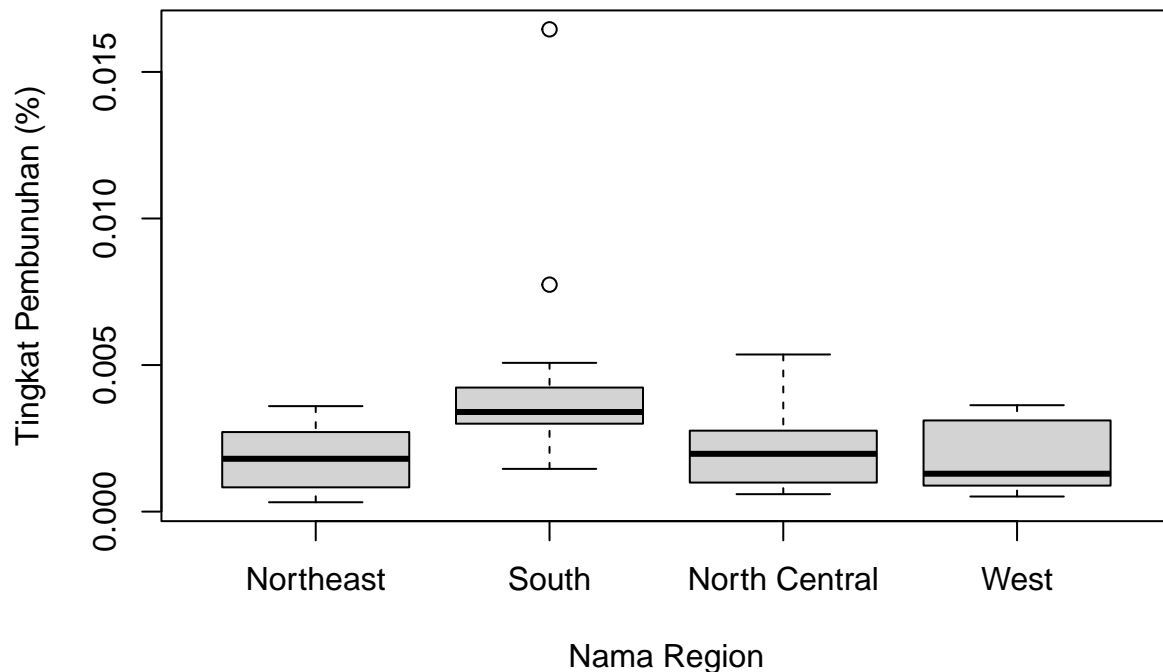
## Boxplot

Boxplot adalah grafik yang digunakan untuk menunjukkan distribusi data berdasarkan kuartil. Grafik ini membantu kita memahami penyebaran, simetri, dan potensi outlier dalam data. Berikut adalah contoh untuk menampilkan persentase tingkat pembunuhan pada tiap-tiap region (Northeast, South, North Central, dan West).

```
murders$rate = with(murders, total / population * 100)
boxplot(
  rate~region,
  data = murders,
  main = "Tingkat Pembunuhan tiap Region",
  xlab = "Nama Region",
  ylab = "Tingkat Pembunuhan (%)"
)
```



## Tingkat Pembunuhan tiap Region



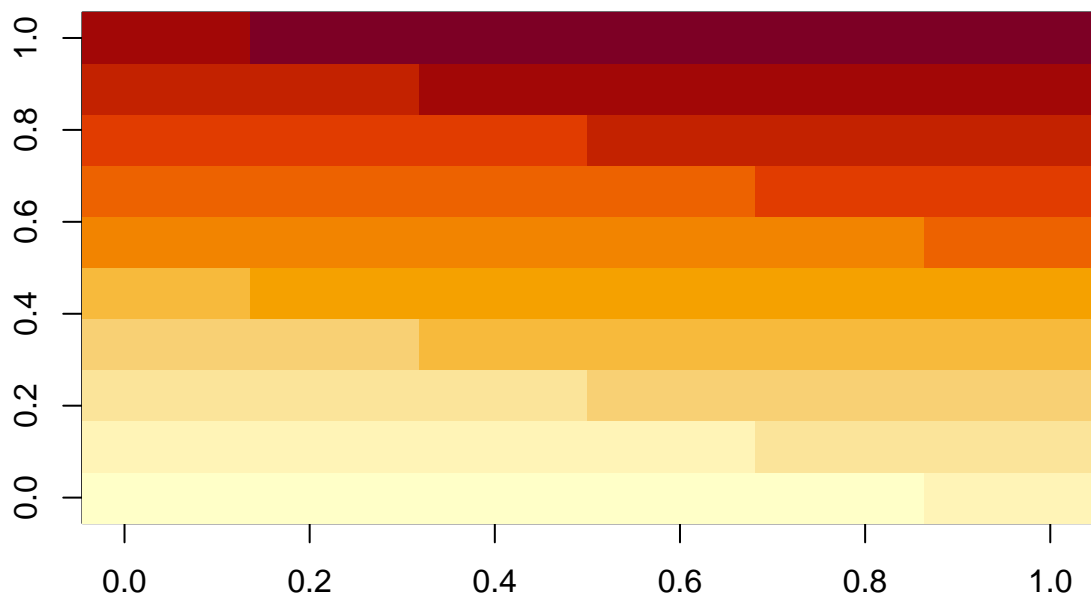
Bagian-bagian dari Boxplot: - Kotak (Box): Bagian tengah dari boxplot adalah kotak yang mewakili rentang interkuartil (interquartile range atau IQR). - Q1 (Kuartil 1): Titik di mana 25% data berada di bawahnya. - Q3 (Kuartil 3): Titik di mana 75% data berada di bawahnya. - Median (Garis di Dalam Kotak): Garis horizontal di dalam kotak menunjukkan nilai median (atau kuartil ke-2), yaitu titik di mana 50% data berada di bawahnya.

Cara Membaca Boxplot: - Letak Median: Jika median (garis di dalam kotak) berada di tengah kotak, distribusi data merata. Jika tidak, datanya miring (skewed). - Median di dekat Q1: Distribusi miring ke kanan (positively skewed). Datanya kecil-kecil. - Median di dekat Q3: Distribusi miring ke kiri (negatively skewed). Datanya gede-gede. - Ukuran Kotak (IQR): Semakin besar kotaknya, semakin besar variasi dalam data di antara kuartil 1 dan kuartil 3. - Garis: Jika garis panjangnya tidak sama, itu menunjukkan penyebaran data yang tidak simetris. - Outlier: Titik di luar garis adalah outlier. Semakin banyak outlier, semakin ekstrim beberapa data dalam dataset.

## Image

Jujur bingung ini apan

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



x

```
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]    1    13    25    37    49    61    73    85    97   109
## [2,]    2    14    26    38    50    62    74    86    98   110
## [3,]    3    15    27    39    51    63    75    87    99   111
## [4,]    4    16    28    40    52    64    76    88   100   112
## [5,]    5    17    29    41    53    65    77    89   101   113
## [6,]    6    18    30    42    54    66    78    90   102   114
## [7,]    7    19    31    43    55    67    79    91   103   115
## [8,]    8    20    32    44    56    68    80    92   104   116
## [9,]    9    21    33    45    57    69    81    93   105   117
## [10,]   10    22    34    46    58    70    82    94   106   118
## [11,]   11    23    35    47    59    71    83    95   107   119
## [12,]   12    24    36    48    60    72    84    96   108   120
```