

# R Notebook

```
#install.packages("philentropy")
#install.packages("readxl")

library(philentropy)

## Warning: package 'philentropy' was built under R version 4.0.5

library(readxl)

## Import data
data <- read_excel("DataCoba.xlsx")
data <- data[1:8, 2:4]

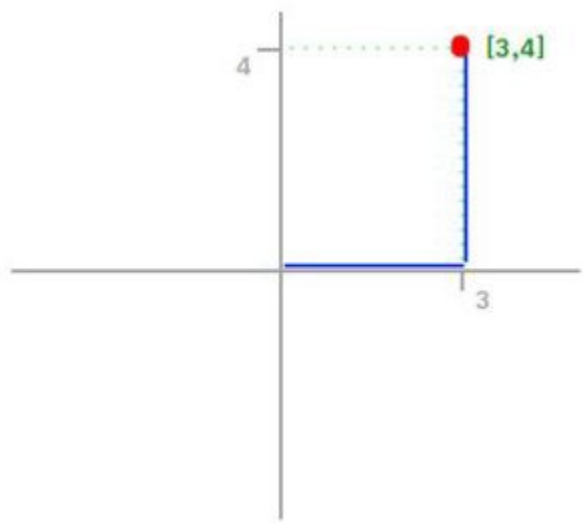
data <- as.matrix(data)
data

##      Feat 1 Feat 2 Feat 3
## [1,]      15      0      5
## [2,]      0     20      0
## [3,]      6      3      5
## [4,]      2      7      0
## [5,]      1      0      3
## [6,]      0      1      1
## [7,]      8      9      3
## [8,]      5      1      7
```

## L1 norm

L1 Norm atau juga biasa dikenal sebagai **Manhattan Distance** atau **Taxicab Norm** adalah penjumlahan dari besaran vektor pada suatu ruang.

Berikut contoh manhattan distance dari data point dengan nilai (0,0) dengan (3,4)



Jarak Manhattan distance dari kedua data point tersebut adalah

$$||X||_1 = |3| + |4| = 7$$

Manhattan distance ini adalah cara paling umum untuk mengukur jarak antar vektor, yaitu jumlah dari selisih absolut dari tiap-tiap komponen vektor.

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|,$$

```
L1 <- distance(data, method="manhattan")
df_L1 = as.data.frame(t(L1)) # Change to dataframe
df_L1

      v1      v2      v3      v4      v5      v6      v7      v8
      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
v1      0      40     12     25     16     20     18     13
v2     40      0     28     15     24     20     22     31
v3     12     28      0     13     10     12     10      5
v4     25     15     13      0     11      9     11     16
v5     16     24     10     11      0      4     16      9
v6     20     20     12      9      4      0     18     11
v7     18     22     10     11     16     18      0     15
v8     13     31      5     16      9     11     15      0

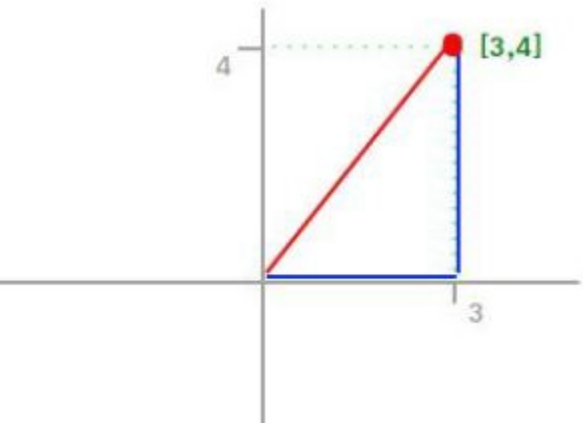
8 rows

# Export to csv
write.csv(df_L1, "export/L1.csv")
```

## L2 norm

L2 norm atau yang biasa dikenal dengan **Euclidean distance** adalah jarak terdekat pada antar **data point**.

Berikut contoh euclidean distance dari data point dengan nilai (0,0) dengan (3,4)



Contoh gambar penjelasan euclidean distance

Jarak Euclidean dari kedua data point tersebut adalah

$$||X||_2 = \sqrt{(|3|^2 + |4|^2)} = \sqrt{9+16} = \sqrt{25} = 5$$

Euclidean distance ini adalah cara yang paling sering digunakan untuk mengukur jarak antar dua titik.

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$

```
L2 <- distance(data, method="euclidean")
df_L2 = as.data.frame(t(L2)) # Change to dataframe
df_L2

      v1      v2      v3      v4      v5      v6      v7      v8
      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
v1  0.000000  25.49510  9.486833  15.588457  14.142136  15.556349  11.575837  10.246951
v2  25.495098  0.00000  18.708287  13.152946  20.248457  19.026298  13.928388  20.856654
v3   9.486833  18.70829  0.000000  7.549834  6.164414  7.483315  6.633250  3.000000
v4  15.588457  13.15295  7.549834  0.000000  7.681146  6.403124  7.000000  9.695360
v5  14.142136  20.24846  6.164414  7.681146  0.000000  2.449490  11.401754  5.744563
v6  15.556349  19.02630  7.483315  6.403124  2.449490  0.000000  11.489125  7.810250
v7  11.575837  13.92839  6.633250  7.000000  11.401754  11.489125  0.000000  9.433981
v8  10.246951  20.85665  3.000000  9.695360  5.744563  7.810250  9.433981  0.000000

8 rows

# Export to csv
write.csv(df_L2, "export/L2.csv")
```

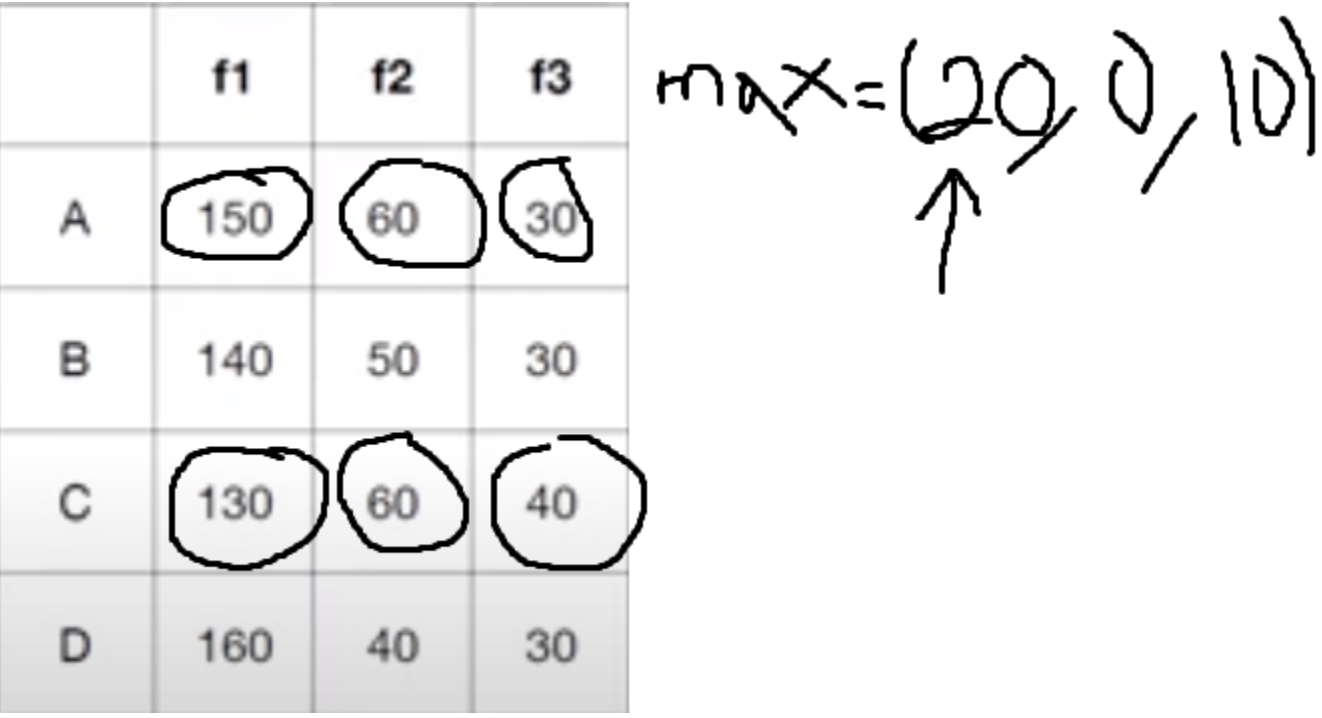
## L∞

L infinite atau **L supremum** adalah jarak nilai maksimum secara absolut antara tiap elemen vektor.

Berikut contoh L Supremum :

$$\mathbf{a} = (a_1, a_2, ..., a_n), \mathbf{b} = (b_1, b_2, ..., b_n)$$

$$d(\mathbf{a}, \mathbf{b}) = \max_i (|b_i - a_i|)$$



Maka didapatkan lah L Sup dari Vektor A dengan C yaitu bernilai 20

```
LSup <- distance(data, method="chebyshev")
df_LSup = as.data.frame(t(LSup)) # Change to dataframe
df_LSup

      v1      v2      v3      v4      v5      v6      v7      v8
      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
v1      0     20      9     13     14     15      9     10
v2     20      0     17     13     20     19     11     19
v3      9     17      0      5      5      6      6      2
v4     13     13      5      0      7      6      6      7
v5     14     20      5      7      0      2      9      4
v6     15     19      6      6      2      0      8      6
v7      9     11      6      6      9      8      0      8
v8     10     19      2      7      4      6      8      0

8 rows

# Export to csv
write.csv(df_LSup, "export/LSup.csv")
```

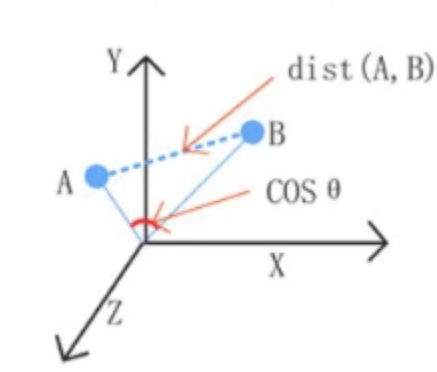
## Cosine

Cosine Similarity adalah ukuran kemiripan antara 2 vektor. Nilai cosine similarity berada pada range antara 0 dan 1.

Berikut adalah rumus hitung dari Cosine Similarity

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}$$

Analogi dari cosine similarity :



Contoh Cosine :

Indeks	Data	Kemunculan data	
		A	B
1	aldo	1	1
2	kenneth	2	1
3	marcel	2	2
4	chaniago	1	1
5	rivaldo	1	1
6	manuel	1	0
7	lieyanto	0	1
8	agus	0	1

Penyelesaian :

$$= \frac{(1x1) + (2x1) + (2x2) + (1x1) + (1x1) + (1x0) + (0x1) + (0x1)}{\sqrt{1^2 + 2^2 + 2^2 + 1^2 + 1^2 + 0^2} \times \sqrt{1^2 + 1^2 + 2^2 + 1^2 + 0^2 + 1^2 + 1^2}}$$

$$= 0.821584$$

```
# Cosine
LCosine <- distance(data, method="cosine")
df_LCosine = as.data.frame(t(LCosine))
df_LCosine

      v1      v2      v3      v4      v5      v6      v7      v8
      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
v1  1.0000000  0.0000000  0.8693183  0.26062335  0.60000000  0.2236068  0.6880237  0.8033264
v2  0.0000000  1.0000000  0.3585686  0.96152395  0.00000000  0.7071068  0.7252407  0.1154701
v3  0.8693183  0.3585686  1.0000000  0.54178501  0.79372539  0.6761234  0.8668284  0.9384892
v4  0.2606233  0.9615239  0.5417850  1.00000000  0.08687445  0.6799001  0.8744375  0.2696375
v5  0.6000000  0.0000000  0.7937254  0.08687445  1.00000000  0.6708204  0.4332001  0.9493858
v6  0.2236068  0.7071068  0.6761234  0.67990010  0.67082039  1.0000000  0.6837635  0.6531973
v7  0.6880237  0.7252407  0.8668284  0.87443754  0.43320011  0.6837635  1.0000000  0.6513389
v8  0.8033264  0.1154701  0.9384892  0.26963754  0.94938577  0.6531973  0.6513389  1.0000000

8 rows

# Export to csv
write.csv(df_LCosine, "export/LCosine.csv")
```