Metode Penggerombolan dan Evaluasinya

[Part 1]

Kuliah 8 | Teknik Pembelajaran Mesin

rahmaanisa@apps.ipb.ac.id

Outline

1. Analisis gerombol
2. Tahapan penggerombolan berhirarki
3. Beberapa metode penggerombolan berhirarki
4. Evaluasi penggerombolan

Analisis Gerombol

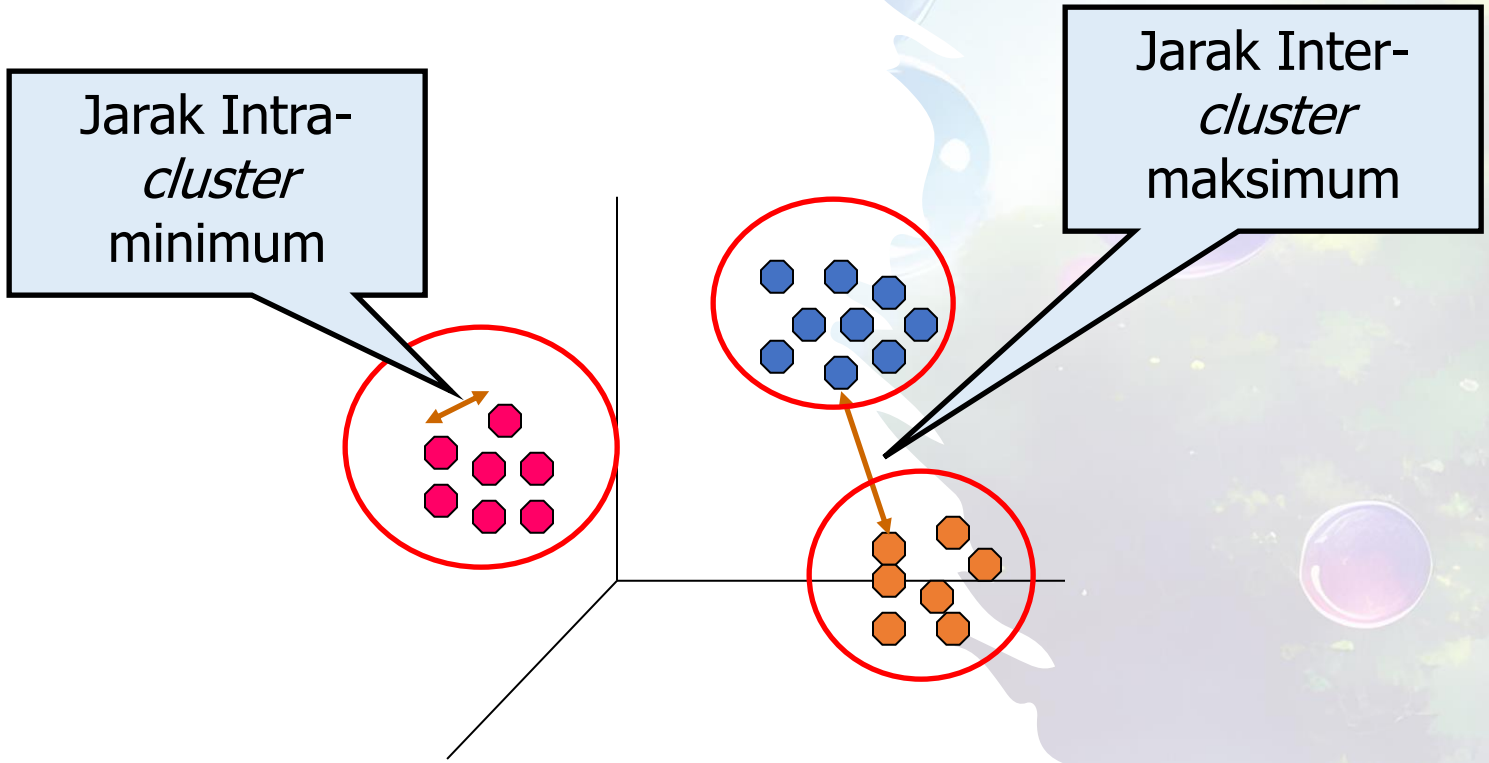
Tipe Klasifikasi

- **Supervised**
  - Segmen-nya telah diketahui
  - Tujuan analisis adalah menentukan kelompok dari individu tertentu berdasarkan karakteristik yang dia miliki, jika sifatnya mirip dengan sifat individu lain yang ada segmen tertentu maka dia dikategorikan ke segmen tersebut.
  - Analisis dilakukan dengan terlebih dahulu menentukan aturan (*rules*) yang digunakan untuk mengklasifikasikan individu
  - Aturan penentuan kelompok dapat berbasis pada model atau non-model
  - Sering disebut sebagai *classification analysis*

- **Unsupervised**
  - Segmen yang ada belum diketahui
  - Analisis lebih bersifat eksploratif dan deskriptif
  - Output berupa pengelompokan individu yang ada dalam data dengan ketentuan bahwa individu yang karakteristiknya mirip akan dikelompokkan pada grup yang sama
  - Sering disebut sebagai *cluster analysis* atau analisis gerombol

Analisis Gerombol

- Menemukan grup dari objek/individu sedemikian rupa sehingga objek dalam grup yang sama memiliki karakteristik yang mirip sedangkan objek dari grup yang berbeda memiliki karakteristik yang kontras/berbeda.



Mengapa Perlu Gerombol?

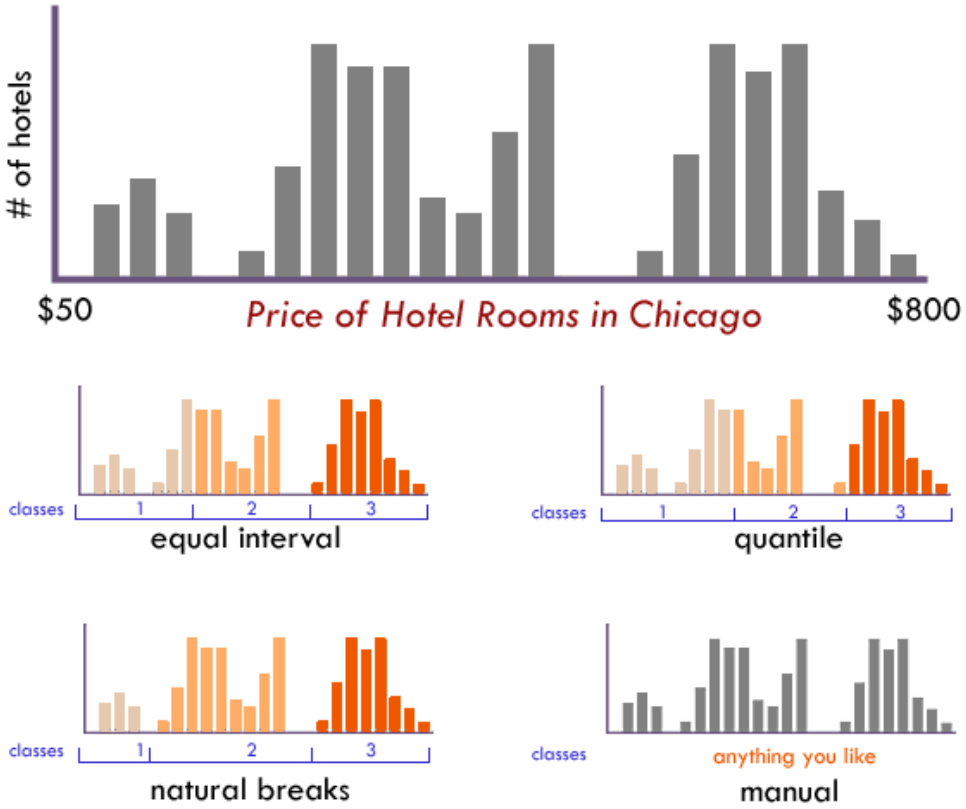
- **Penggerombolan**: memberikan kumpulan kelompok data objek kelompok sehingga
  - Dalam satu gerombol satu sama lain mirip
  - Objek antar gerombol tidak mirip

- Hasil penggerombolan dapat digunakan sebagai:
  - *stand-alone tool* untuk mendapatkan pengetahuan yang dalam sebaran data
    - Visualisasi kelompok dapat mengungkap informasi penting
  - *preprocessing step* untuk algoritma lain
    - Pengindeksan atau kompresi yang efisien sering kali bergantung pada clustering

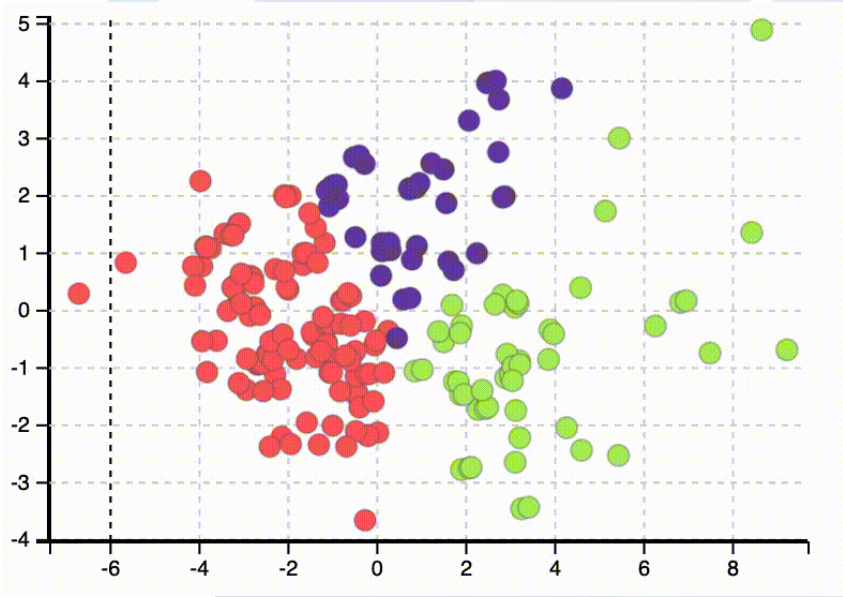
Penerapan Analisis Gerombol

- Segmentasi, contohnya segmentasi pelanggan menjadi beberapa kelompok dengan pola demografis atau perilaku pembelian yang mirip. Hal ini dapat dimanfaatkan untuk *marketing campaign* atau analisis yang lebih detail pada subgroup tertentu
- Deteksi suatu anomali pada data, misalnya pencilan, atau pengamatan dengan perilaku yang tidak biasa.
- Penyederhanaan gugus data yang amat sangat besar sekali, dengan mengelompokkan sejumlah besar *feature* yang mirip menjadi beberapa kategori yang homogen dan ukurannya jauh lebih sedikit.

Teknik Penggerombolan



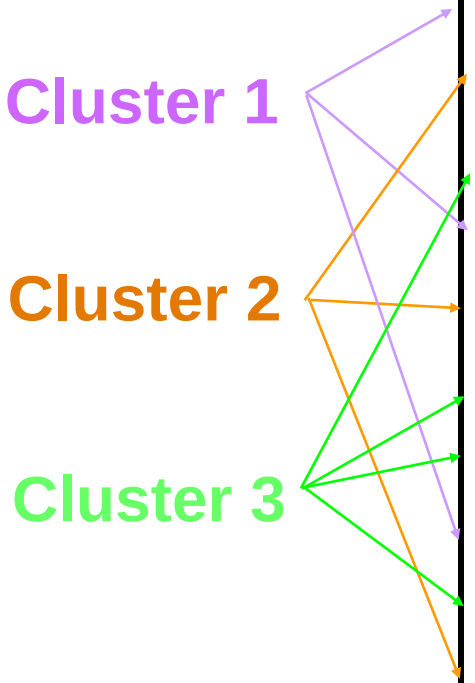
- Satu peubah → *equal intervals*, *mean-standard deviation*, *quantiles*, & *natural breaks*
- Dua peubah → penggerombolan secara visual
- Lebih dari dua peubah → sulit menggunakan visualisasi



Sumber:
https://www.axismaps.com/guide/data-classification
https://towardsdatascience.com/spotlighting-a-visual-approach-to-precisely-interpret-the-clustering-f4c56dba39bb

Teknik Penggerombolan

- Menggunakan ukuran kuantitatif untuk menyatakan kedekatan (kemiripan) antar pengamatan → menggunakan konsep jarak:
  - $\mathrm{d}(a, b) \geq 0$
  - $\mathrm{d}(a, a) = 0$
  - $\mathrm{d}(a, b) = \mathrm{d}(b, a)$
  - $\mathrm{d}(a, b)$ meningkat berarti kedua objek a dan b semakin tidak mirip
  - $\mathrm{d}(a, c) \leq \mathrm{d}(a, b) + \mathrm{d}(b, c)$
  - Semua peubah berupa bersifat numerik (*kontinu*)



	X1	X2	X3	X3
Obs1	5	2	1	3
Obs2	3	3	4	2
Obs3	2	4	3	5
Obs4	5	3	2	4
Obs5	.	.	.	.
Obs6	.	.	.	.
Obs7	.	.	.	.
Obs8	.	.	.	.
Obs9	.	.	.	.
Obs10	.	.	.	.

Konsep Jarak / Distance

- Jarak / distance merupakan konsep inti dari proses cluster analysis.
- Jarak sering juga disejajarkan dengan istilah dissimilarity
- Andaikan terdapat dua individu/objek yaitu A dan B, jarak antara keduanya dinotasikan d(A, B), maka sifat dari jarak adalah:
  - $d(A,B) = d(B,A) \geq 0$
  - $d(A,A) = d(B,B) = 0$
  - Jika ada individu lain C maka $d(A,B) \leq d(A,C) + d(C,B)$
- Terdapat bermacam-macam formula/definisi perhitungan jarak:
  - Euclidean, Weighted Euclidean, Mahalanobis, City block, …
  - Jaccard, Hamming, …

Struktur Data

- *data* matrix

attributes/dimensions

tuples/objects

$$\begin{bmatrix} x_{11} & \dots & x_{1\ell} & \dots & x_{1d} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{i\ell} & \dots & x_{id} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{n\ell} & \dots & x_{nd} \end{bmatrix}$$

- *Distance* matrix

objects

objects

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Euclidean Distance

$$d(A,B)=\sqrt{\sum_{k=1}^{p}(a_k-b_k)^2}$$

Dengan p adalah banyaknya dimensi/ atribut/ variabel dan a_k *dan* b_k adalah nilai dari objek A dan B untuk variabel ke-k.

Ilustrasi

Nama	Tinggi Badan	Berat Badan	Umur
Ani	160	80	36
Beni	180	70	40
Carla	180	60	45

$$d(\text{Ani},\text{Beni})=\sqrt{(160-180)^2+(80-70)^2+(36-40)^2}=22.72$$

$$d(\text{Ani},\text{Carla})=\sqrt{(160-180)^2+(80-60)^2+(36-45)^2}=29.68$$

$$d(\text{Beni},\text{Carla})=\sqrt{(180-180)^2+(70-60)^2+(40-45)^2}=11.18$$

Matriks Jarak

Ilustrasi

Nama	Tinggi Badan	Berat Badan	Umur
Ani	160	80	36
Beni	180	70	40
Carla	180	60	45

$$d(\text{Ani},\text{Beni}) = \sqrt{(160-180)^2 + (80-70)^2 + (36-40)^2} = 22.72$$

$$d(\text{Ani},\text{Carla}) = \sqrt{(160-180)^2 + (80-60)^2 + (36-45)^2} = 29.68$$

$$d(\text{Beni},\text{Carla}) = \sqrt{(180-180)^2 + (70-60)^2 + (40-45)^2} = 11.18$$

Nama	Ani	Beni	Carla
Ani	0	22.72	29.68
Beni	22.72	0	11.18
Carla	29.68	11.18	0

Minkowski Distance

- Formula

$$dist = (\sum_{k=1}^{p} | a_k - b_k |^r)^{\frac{1}{r}}$$

Dengan *r* adalah sebuah parameter, *p* adalah banyanya dimensi (atribut), a_k dan b_k masing-masing adalah nilai atribut ke *k* dari objek A dan B.

- Jika *r* = 1 → City Block (Manhattan, taxicab, L_1 norm) distance.
- Jika *r* = 2. Euclidean distance
- Minkowski Distance dapat dipandang sebagai bentuk umum dari Euclidean Distance

Similarity Between Binary Vectors

- Pada situasi tertentu, setiap objek hanya dicirikan oleh atribut yang bersifat biner (hanya bernilai 0 atau 1)

- Proses mendapatkan jarak dapat dilakukan dengan menghitung beberapa nilai berikut
  M_{01} = banyaknya atribut dimana A bernilai 0 dan B bernilai 1
  M_{10} = banyaknya atribut dimana A bernilai 1 dan B bernilai 0
  M_{00} = banyaknya atribut dimana A bernilai 0 dan B bernilai 0
  M_{11} = banyaknya atribut dimana A bernilai 1 dan B bernilai 1

- Simple Matching and Jaccard Distance/Coefficients
  SMC = number of matches / number of attributes
  = $(M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$

  J = number of value-1-to-value-1 matches / number of not-both-zero attributes values
  = $(M_{11}) / (M_{01} + M_{10} + M_{11})$

Pengaruh satuan pengukuran variabel

Nama	Tinggi Badan(cm)	Berat Badan	Umur
Ani	160	80	36
Beni	180	70	40
Carla	180	60	45

Nama	Tinggi Badan (m)	Berat Badan	Umur
Ani	1.60	80	36
Beni	1.80	70	40
Carla	1.80	60	45

Nama	Ani	Beni	Carla
Ani	0	22.72	29.68
Beni	22.72	0	11.18
Carla	29.68	11.18	0

Nama	Ani	Beni	Carla
Ani	0	10.72	21.93
Beni	10.72	0	11.18
Carla	21.93	11.18	0

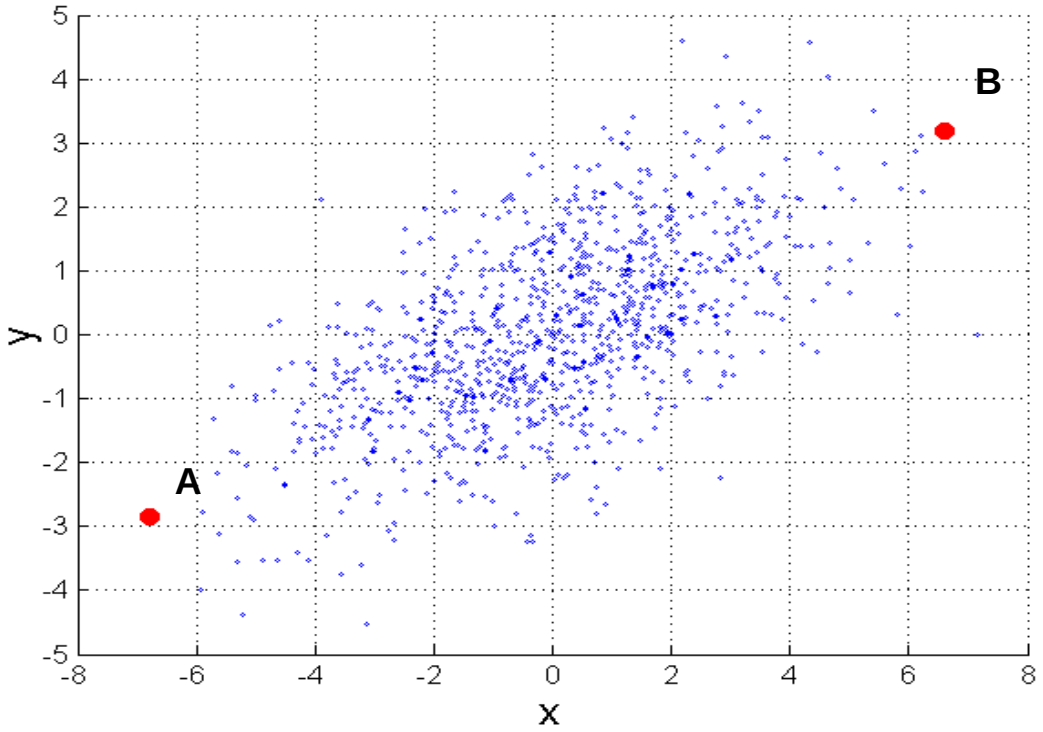
Semula, BENI lebih mirip CARLA…. jika menggunakan satuan yang baru (nilai datanya tetap) BENI lebih mirip ANI

Pengaruh Satuan Pengukuran Variabel

- Perhatikan bahwa ketika tinggi badan dalam satuan meter, nilai-nilai datanya menjadi lebih kecil dibandingkan variabel lain.
- Variabel seperti ini menjadi “tidak penting” dalam penentuan jarak, dan seolah tersisihkan untuk menentukan kedekatan antar objek.
- Untuk menghindari kejadian semacam ini, diusulkan untuk menggunakan data yang terbakukan (standardized) dan bukan data asli.
- Data yang terbakukan menyebabkan semua variabel memiliki keragaman yang sama sehingga akan dianggap memiliki kepentingan yang seimbang dalam menentukan besar kecilnya jarak antar objek.

Mahalanobis Distance

$$\text{mahalanobis}(p,q) = (p-q)\Sigma^{-1}(p-q)^T$$



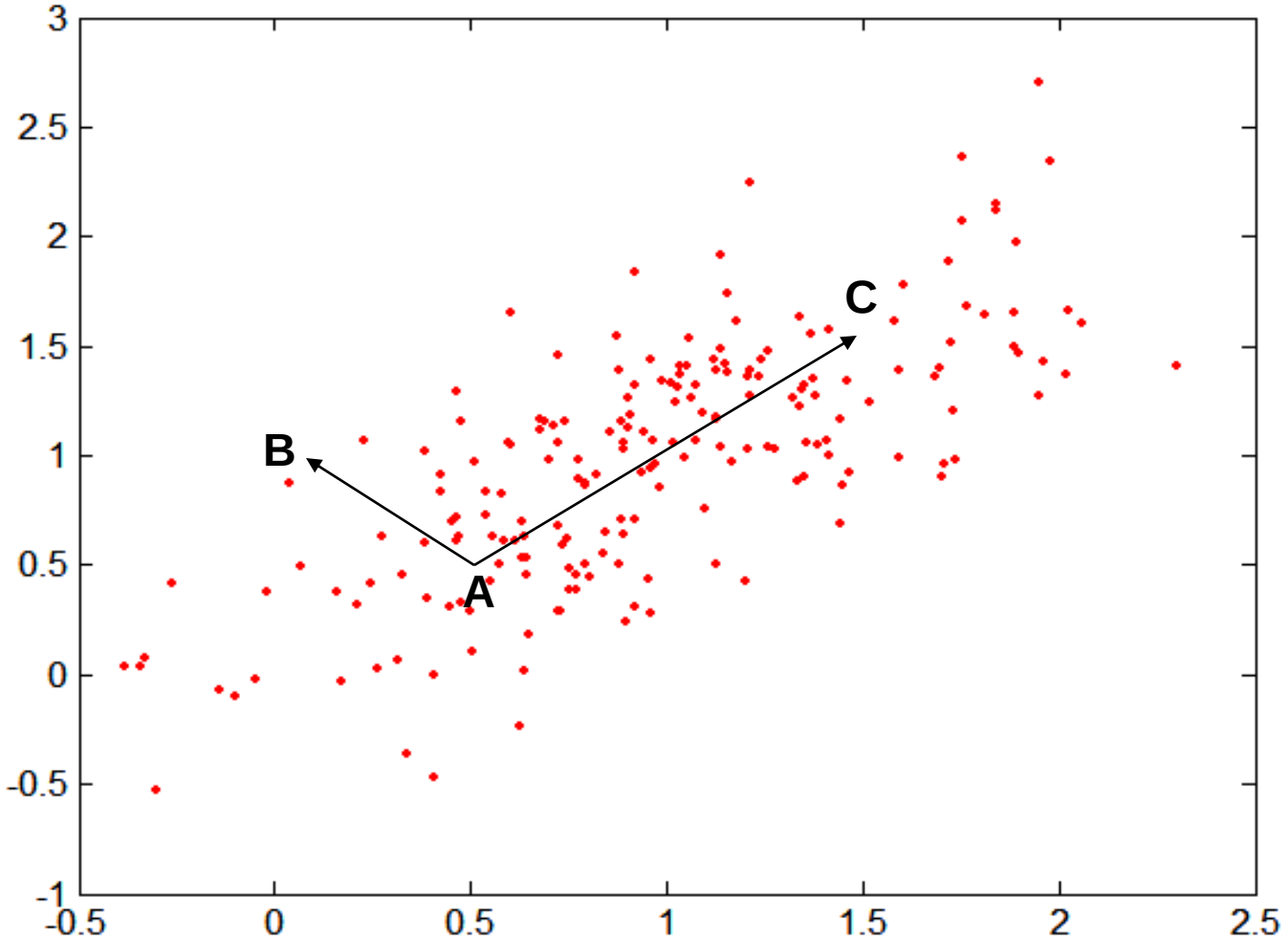
Σ adalah matriks ragam peragam dari data X.

$$\Sigma_{j,k} = \frac{1}{n-1}\sum_{i=1}^{n}(X_{ij} - \overline{X}_j)(X_{ik} - \overline{X}_k)$$

Jika antar variabel pada data X memiliki korelasi 0, maka matriks kovarian akan berupa matriks identitas dan jarak mahalanobis akan sama dengan jarak euclid.

Useful for detecting outliers.

Mahalanobis Distance



A: (0.5, 0.5)

B: (0, 1)

C: (1.5, 1.5)

Mahal(A,B) = 5

Mahal(A,C) = 4

Prosedur Penggerombolan

- Penggerombolan berhirarki
  - Aglomeratif (dimulai dari *n* gerombol menjadi 1 gerombol)
  - Divisif (dimulai dari 1 gerombol menjadi *n* gerombol)
  - Banyaknya gerombol ditentukan berdasarkan dendogram

- Penggerombolan non-hirarki
  - Banyaknya gerombol yang ingin dibentuk sudah diketahui sejak awal

PENGGEROMBOLAN BERHIRARKI

Penggerombolan Berhirarki

- Agglomeratif:
  - Setiap n pengamatan Each of the *n* observations dianggap merupakan gerombol terpisah
  - Setiap dua gerombol yang mirip berdasarkan aturan jarak digabungkan, sehingga pada tahap ke-1 akan terdapat $n - 1$ gerombol.
  - Pada tahap ke-2, gerombol lain digabungkan sehingga terbentuk $n - 2$ gerombol, dan seterusnya.
  - Terdapat penggabungan pada setiap tahap sampai akhirnya seluruh pengamatan tergabung ke dalam satu gerombol pada tahap terakhir.

- Divisif
  - Seluruh pengamatan dianggap sebagai satu gerombol.
  - Pengamatan yang dianggap paling berbeda akan dikeluarkan menjadi gerombol terpisah.
  - Pada tahap ke-1 akan terdapat dua gerombol, pada tahap ke-2 akan terdapat tiga gerombol, dan seterusnya, hingga pada tahap terakhir akan terbentuk sejumlah n gerombol, sama dengan banyaknya pengamatan.

- Banyaknya gerombol menentukan kapan algoritma akan berhenti.

Penggerombolan Aglomeratif Berhirarki

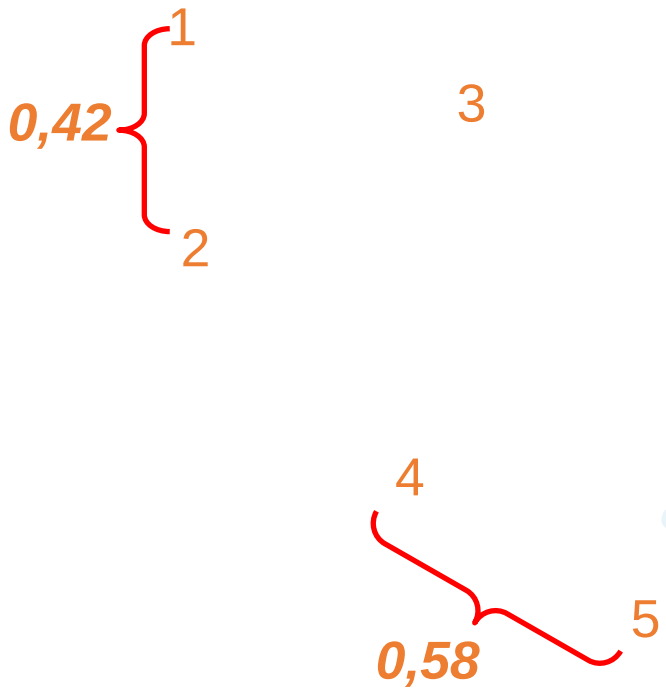
- Kedekatan antara dua gerombol menjadi penentu dalam komputasinya.
  - Terdapat beberapa pendekatan yang berbeda dalam menentukan jarak antar-gerombol.
  - Dapat menggunakan pendekatan single linkage, complete linkage, average linkage, Ward's method, dsb.

Agglomerative Clustering Algorithm

Basic algorithm is :

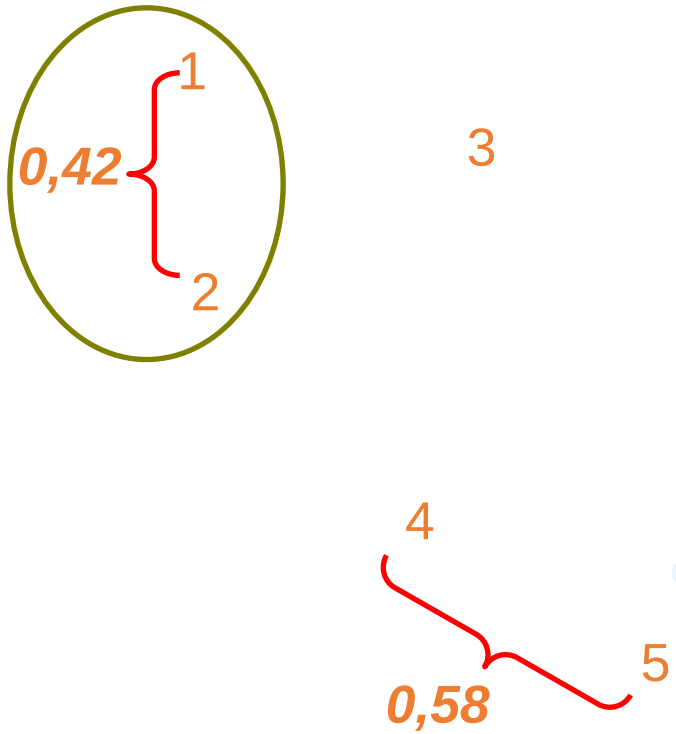
1. Compute the proximity matrix
2. Let each data point be a cluster
3. **Repeat**
4. Merge the two closest clusters
5. Update the proximity matrix
6. **Until** only a single cluster remains

Small artificial example

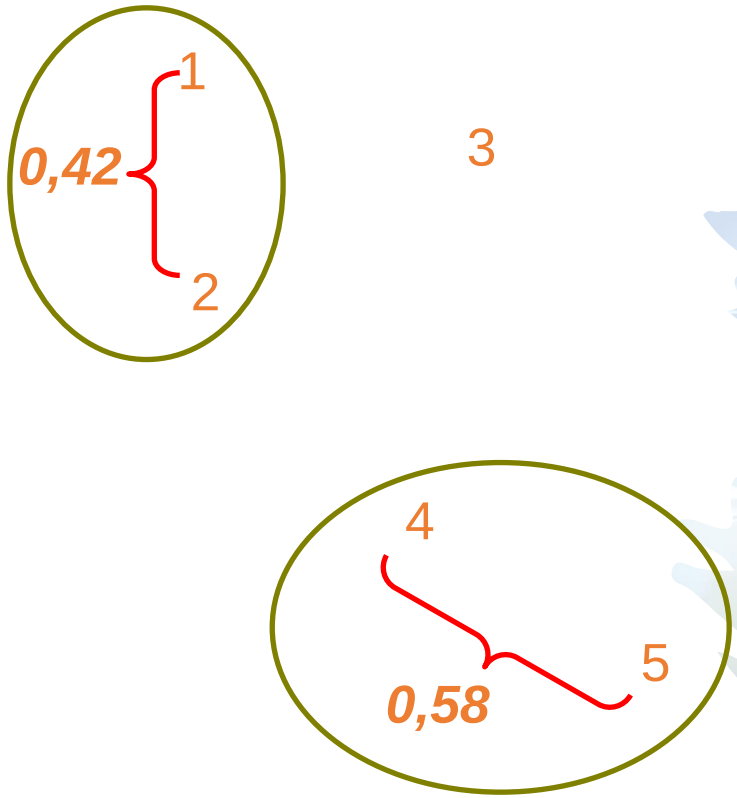


Note: 6 points yield 15 possible pairwise distances - [n*(n-1)]/2

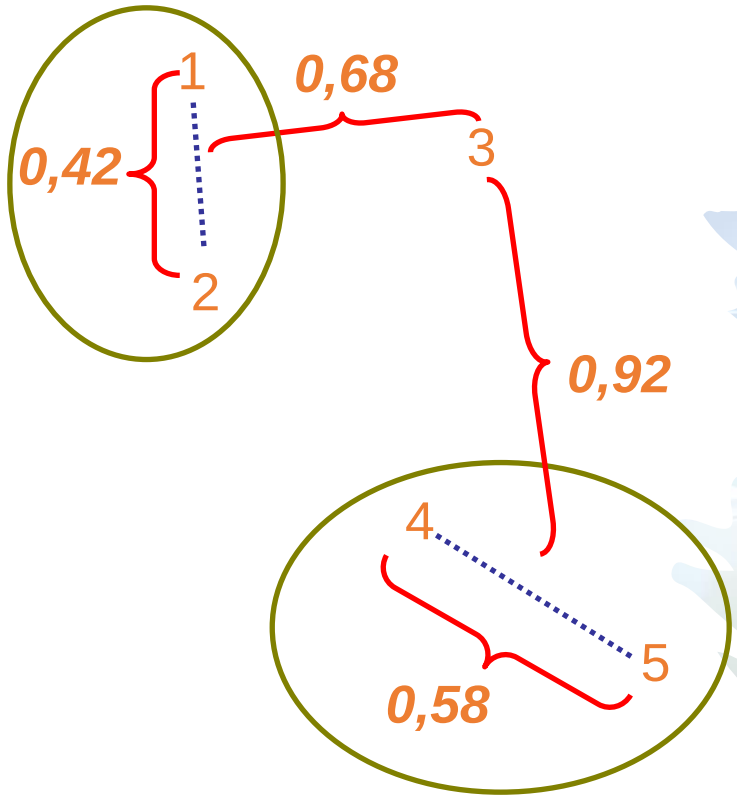
Small artificial example



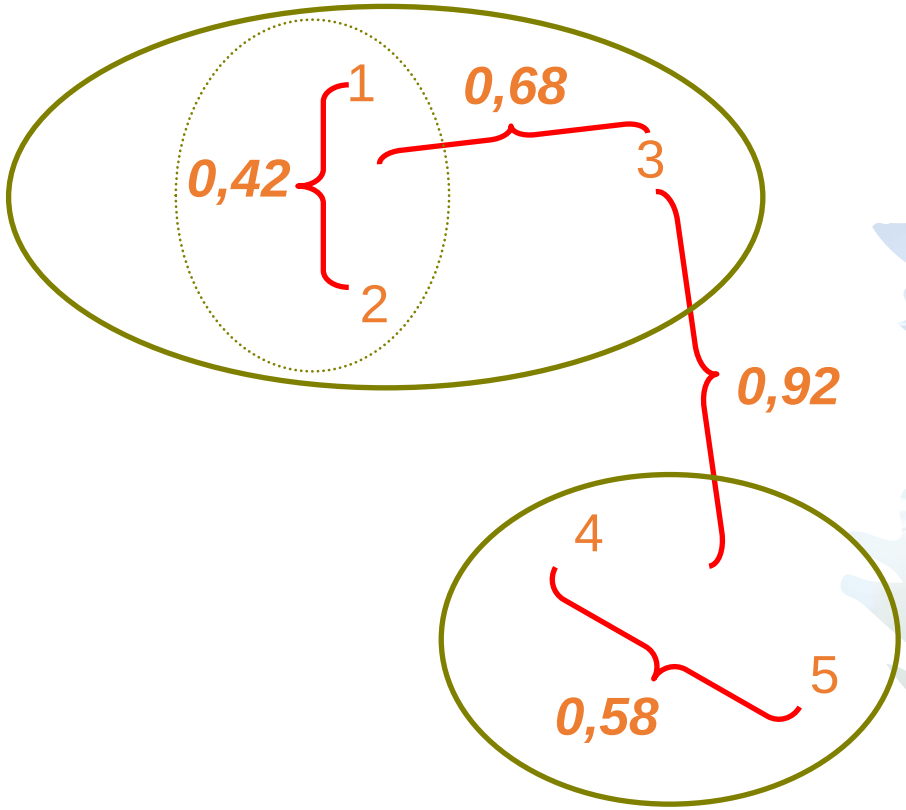
Small artificial example



Small artificial example

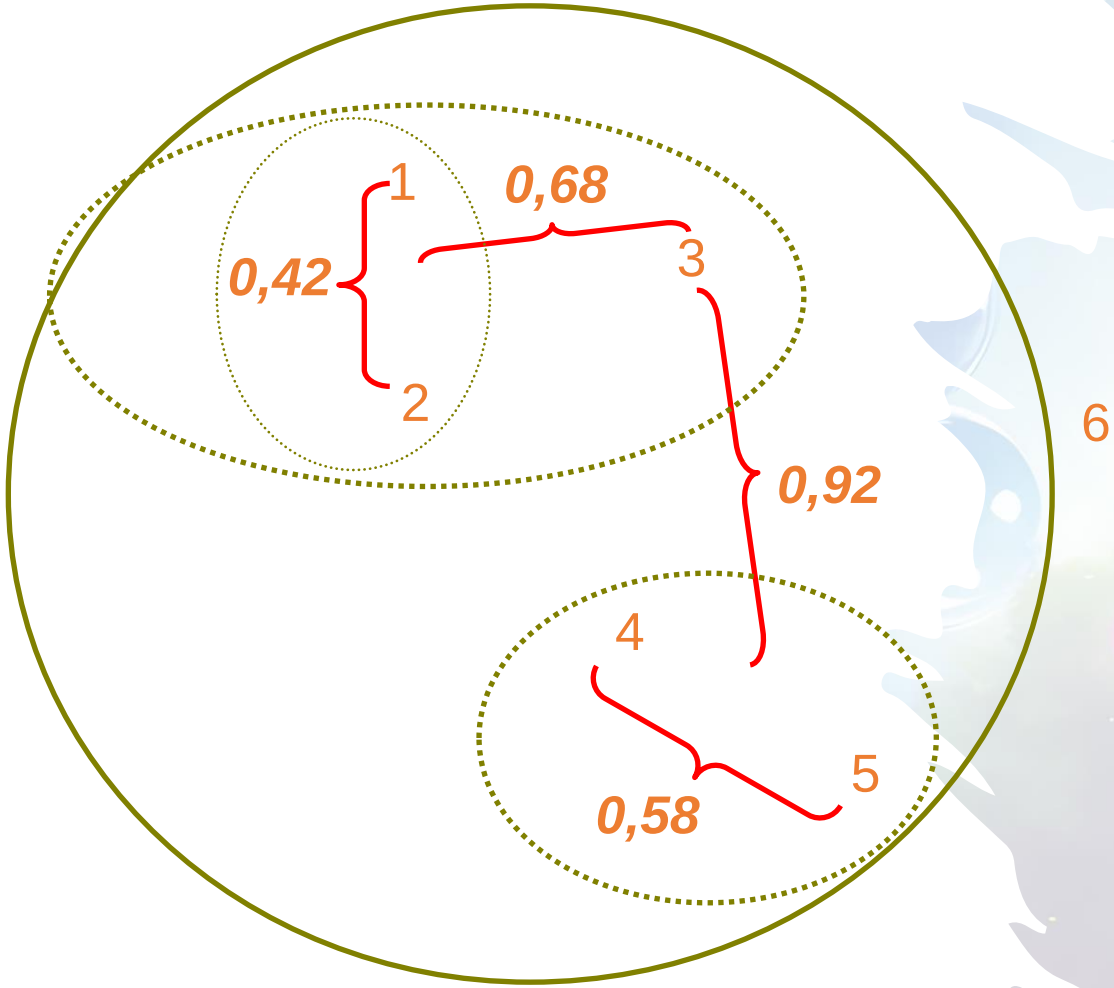


Small artificial example

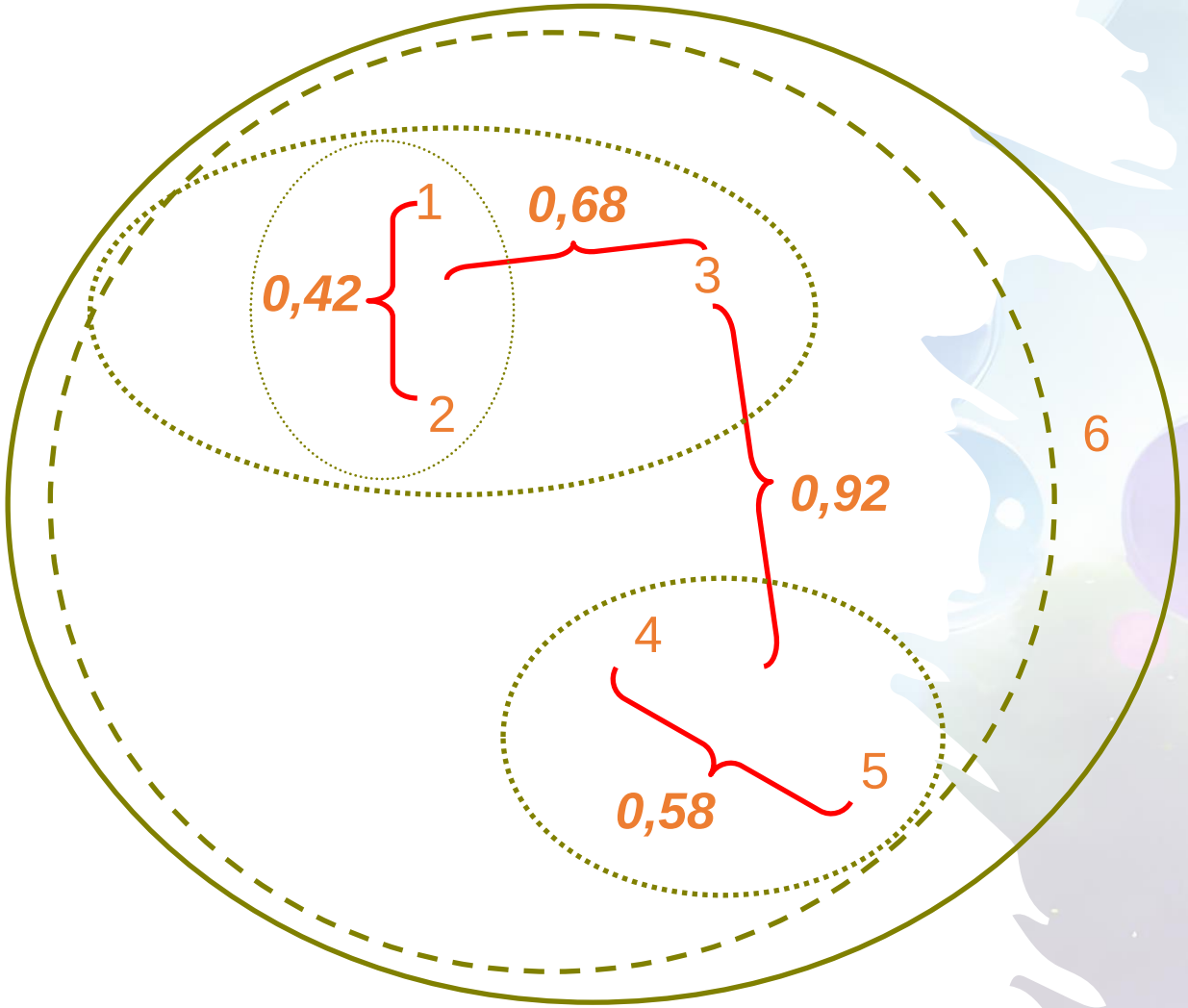


6

Small artificial example

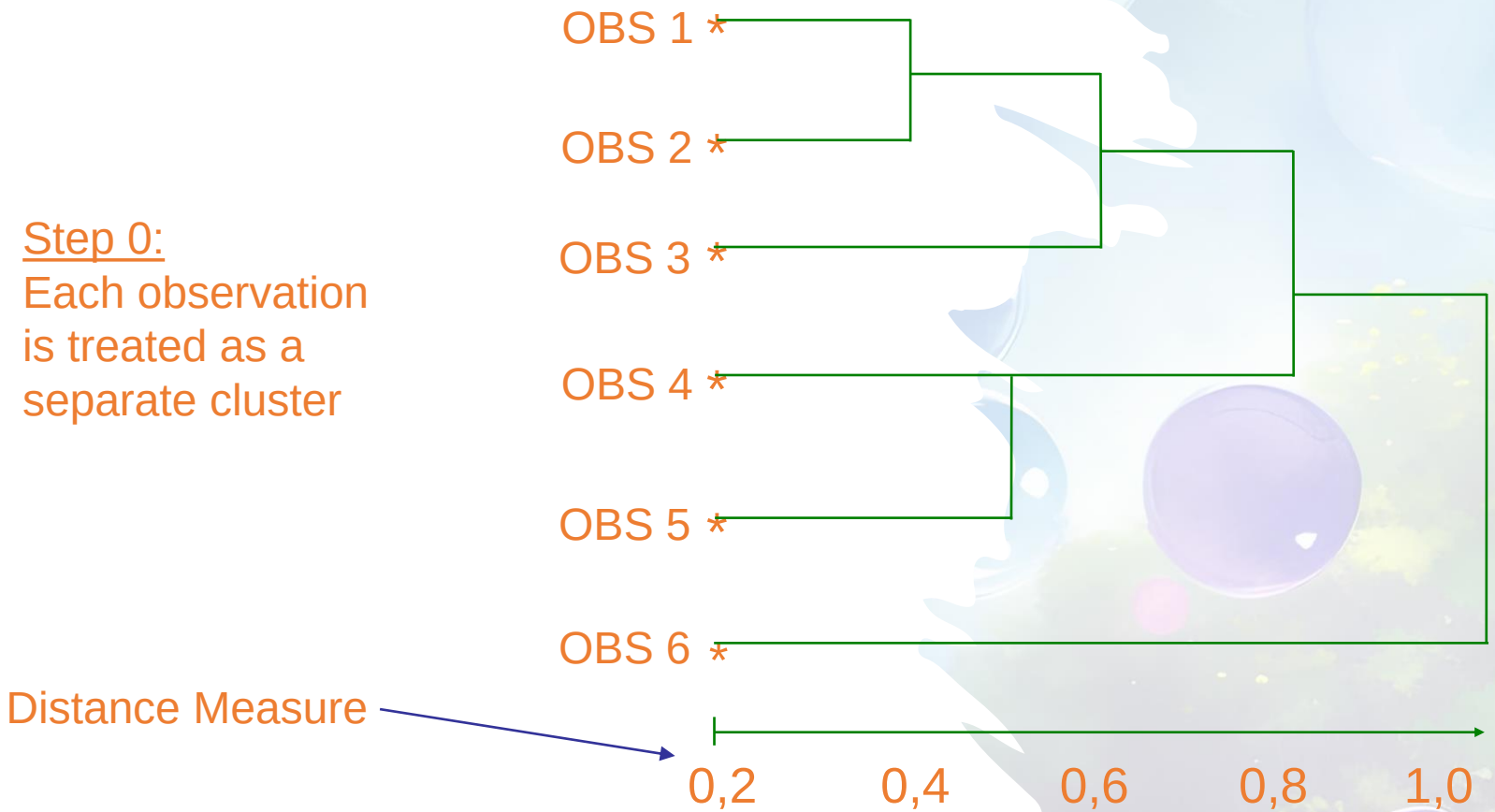


Small artificial example



Dendrogram

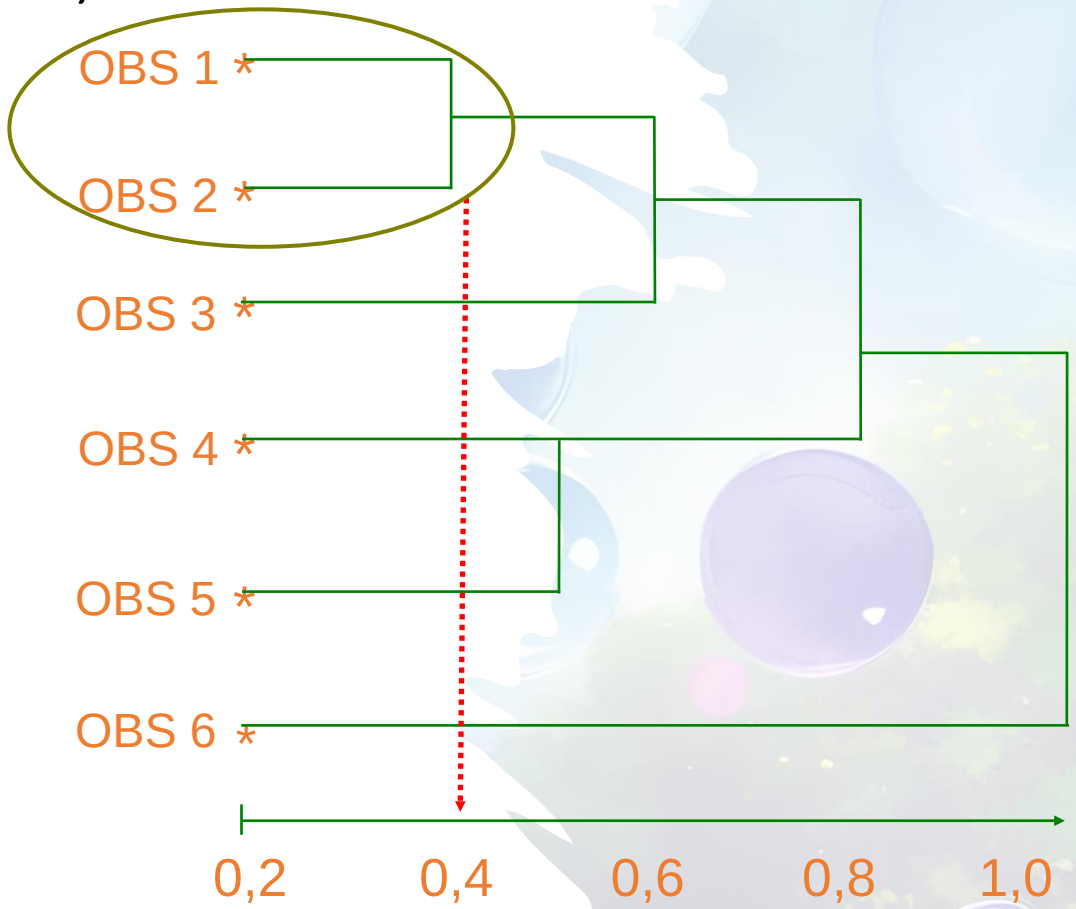
Step 0:
Each observation is treated as a separate cluster



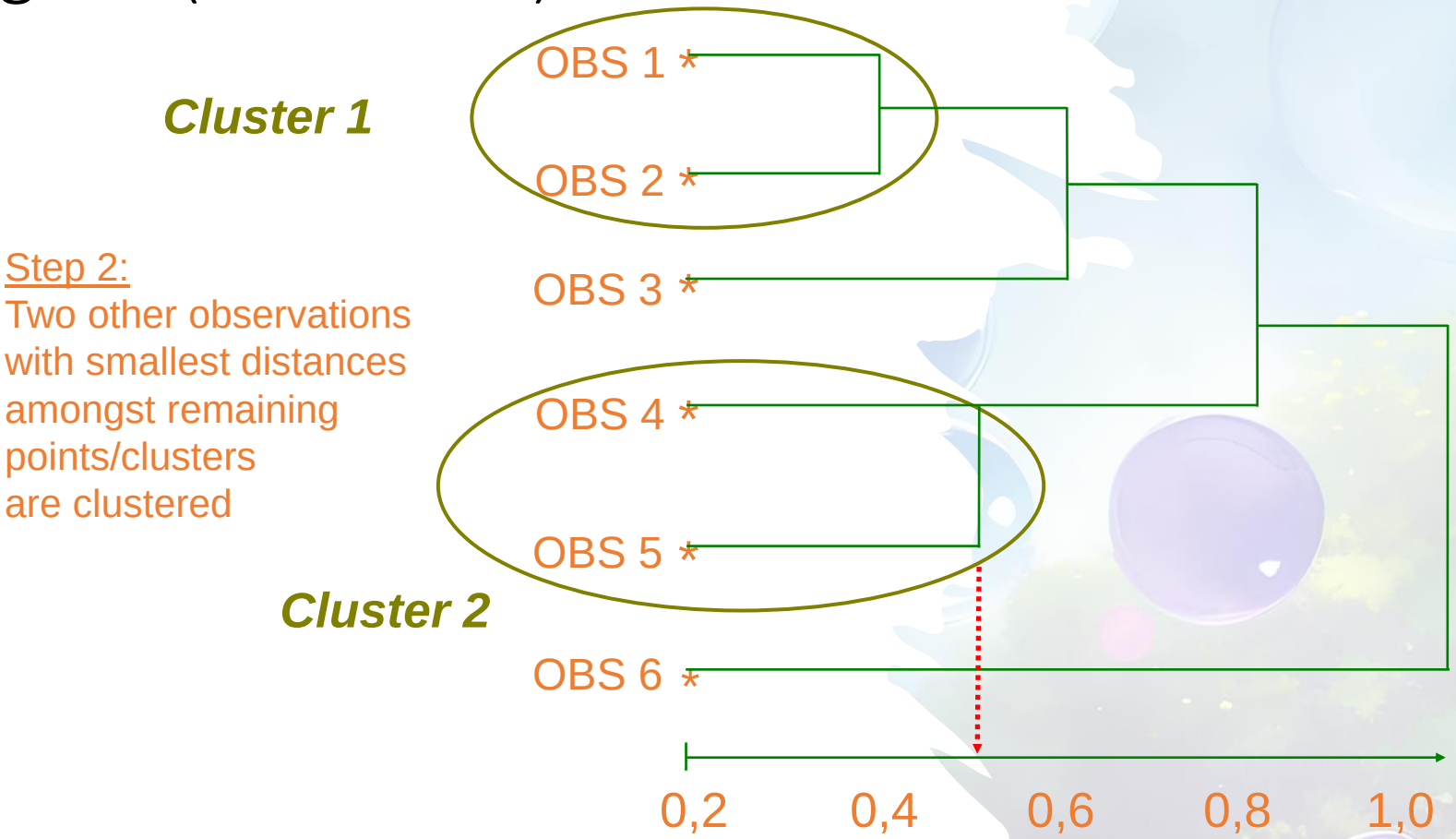
Dendrogram (Continued)

Cluster 1

<u>Step 1:</u>
Two observations with smallest pairwise distances are clustered

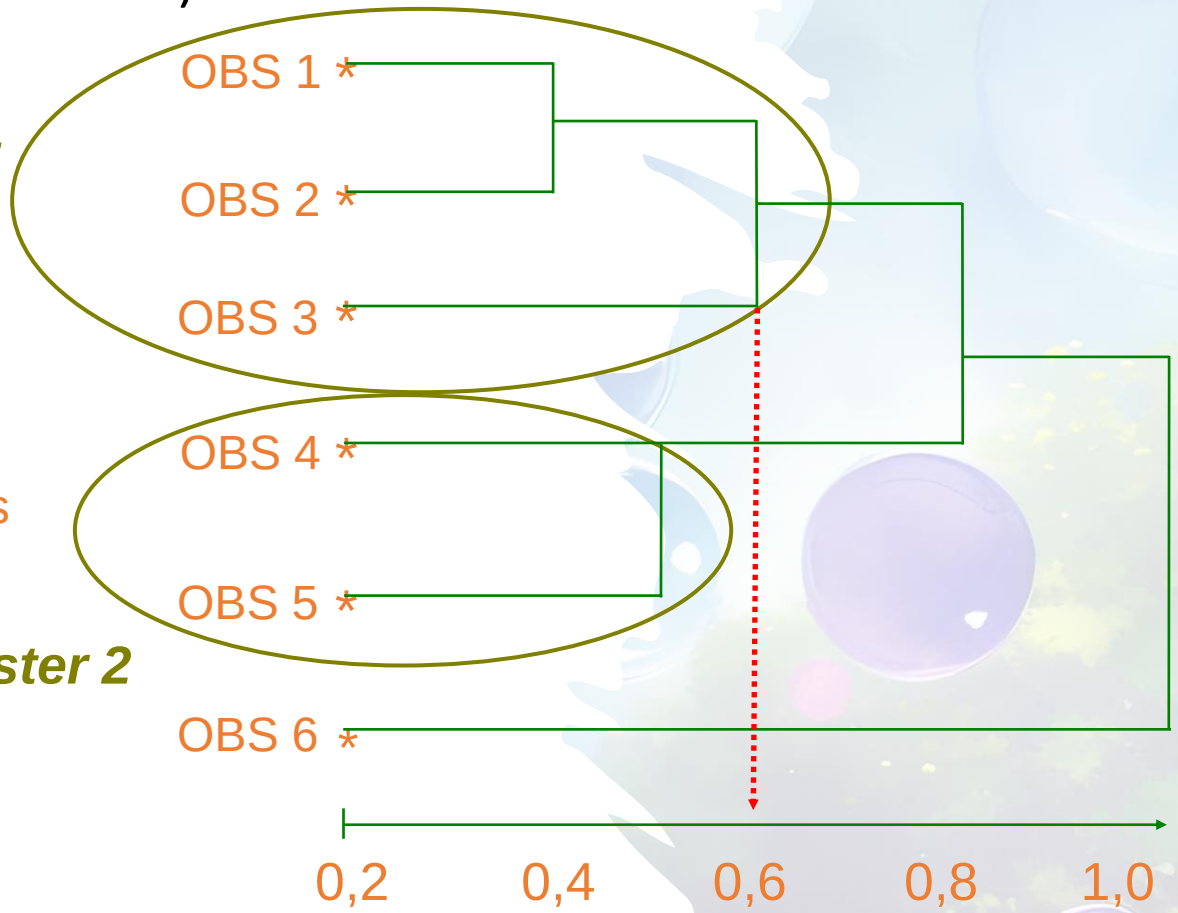


Dendrogram (Continued)



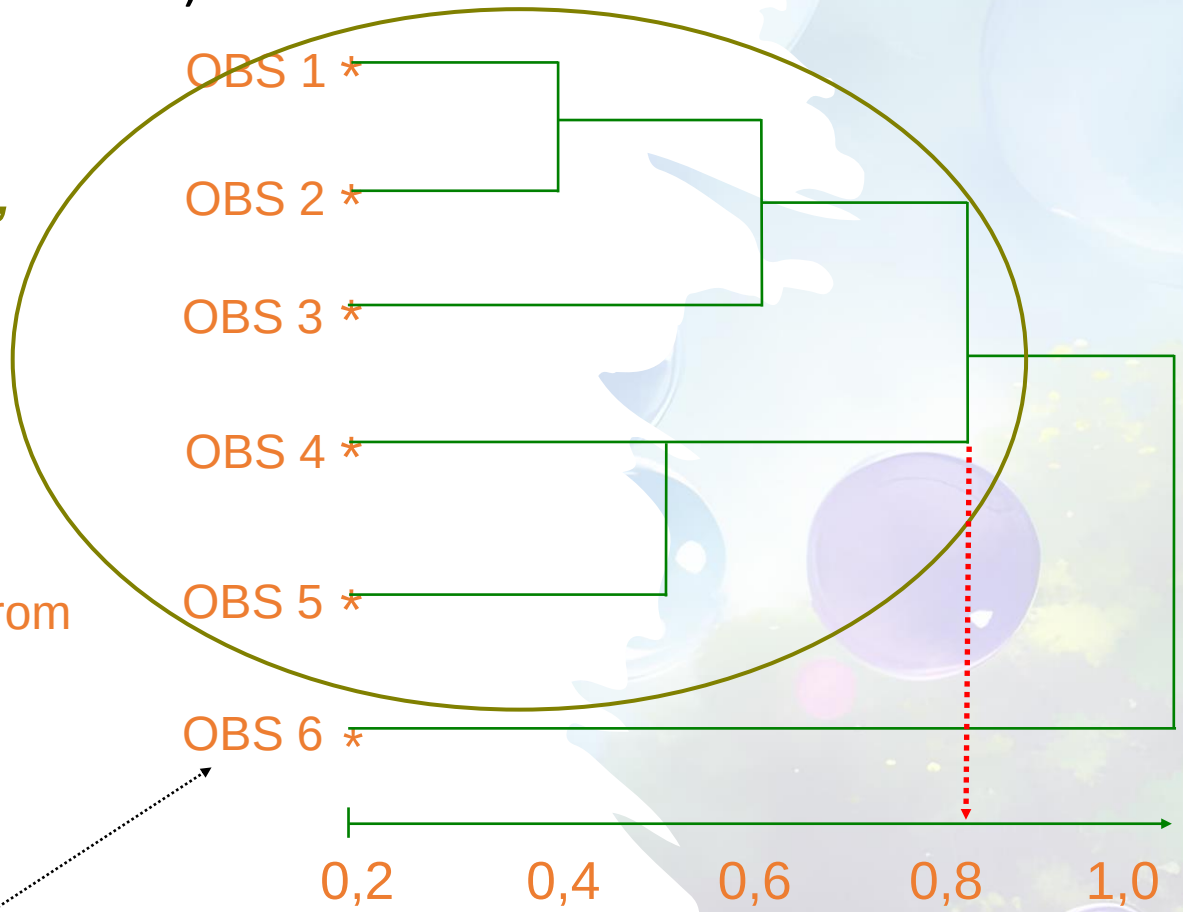
Step 2:
Two other observations with smallest distances amongst remaining points/clusters are clustered

Dendrogram (Continued)



Step 3:
Observation 3 joins with *cluster 1*

Dendrogram (Continued)



"Supercluster"

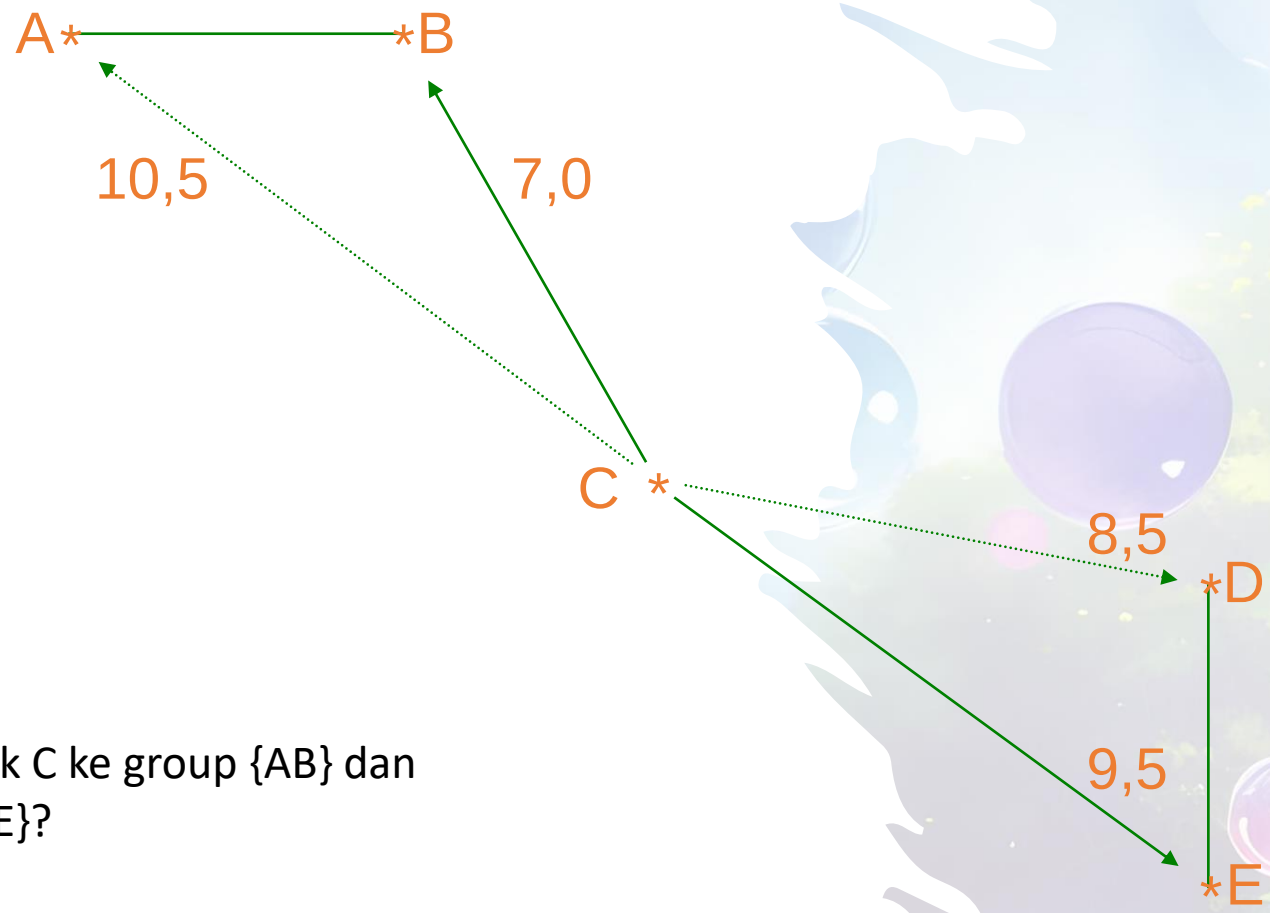
Step 4:
Cluster 1 and 2 - from Step 3 joint into a "Supercluster"

A single observation remains unclustered (Outlier)

Bagaimana Menggabungkan Cluster?

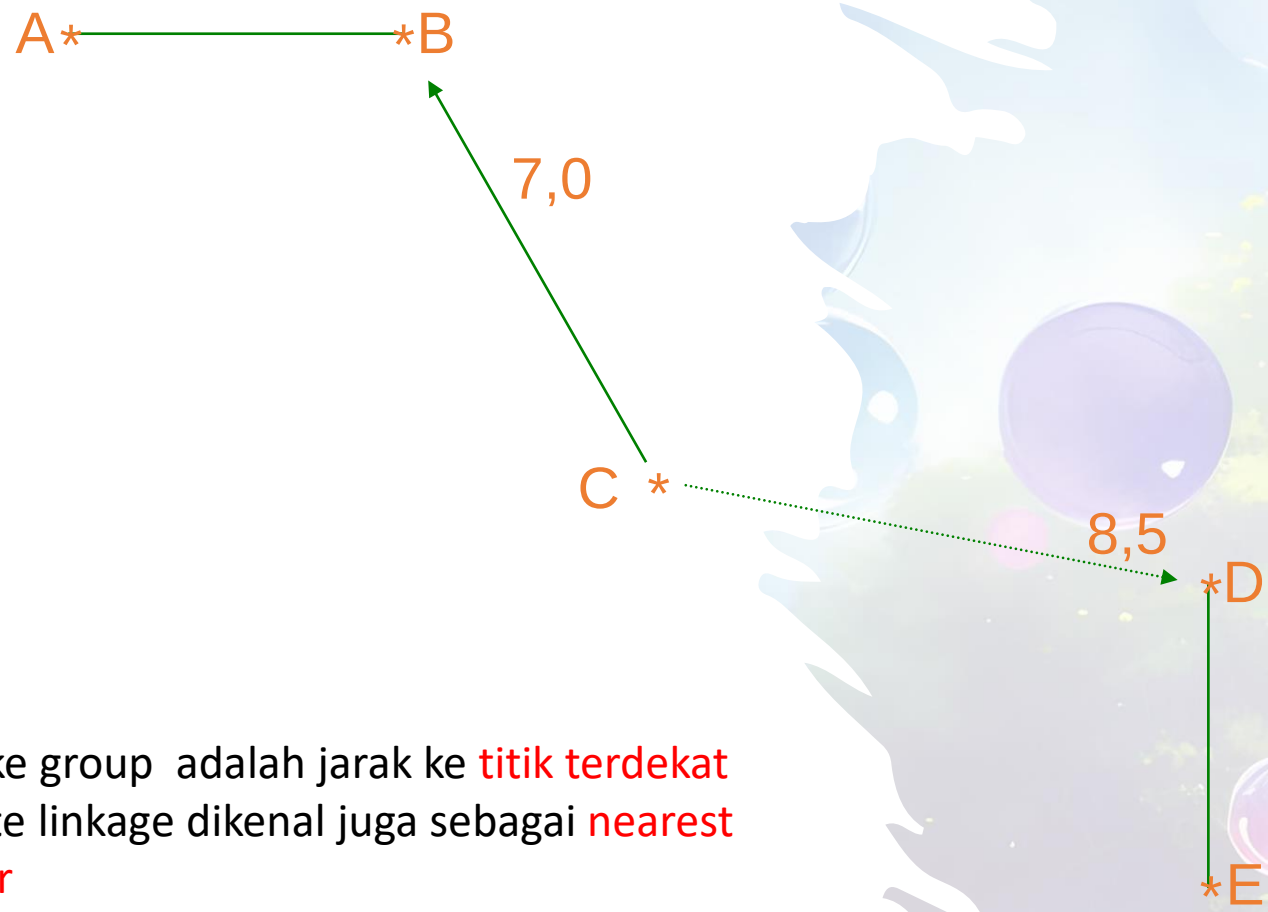
- Pautan Tunggal (*Single Linkage, Nearest Neighbor)*
- Pautan Lengkap (*Complete Linkage, Farthest Neighbor*)
- Pautan Centroid (*Centroid Linkage*)
- Pautan Median (*Median Linkage*)
- Pautan Rataan (*Average Linkage*)

Ilustrasi Linkage: jarak antar grup individu



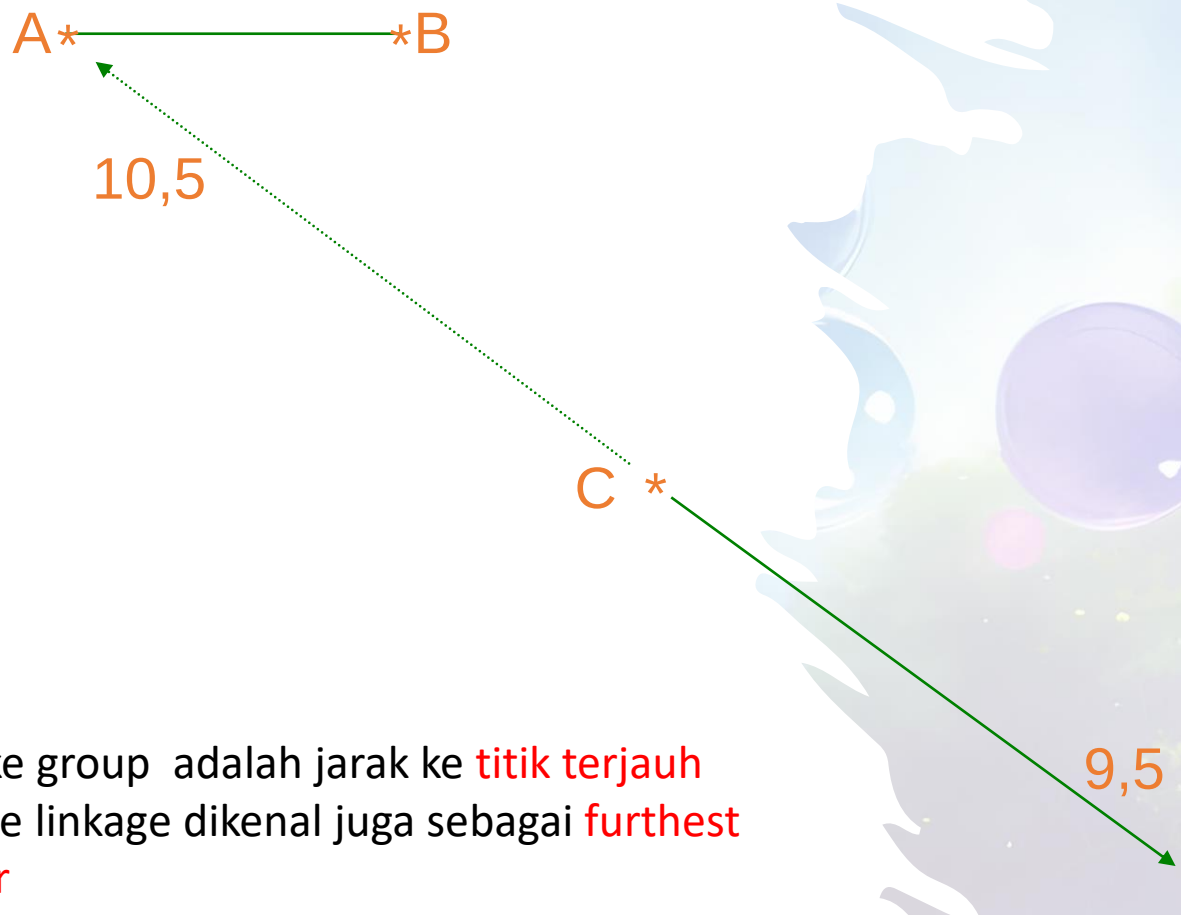
Berapa jarak C ke group {AB} dan ke group{DE}?

Single linkage



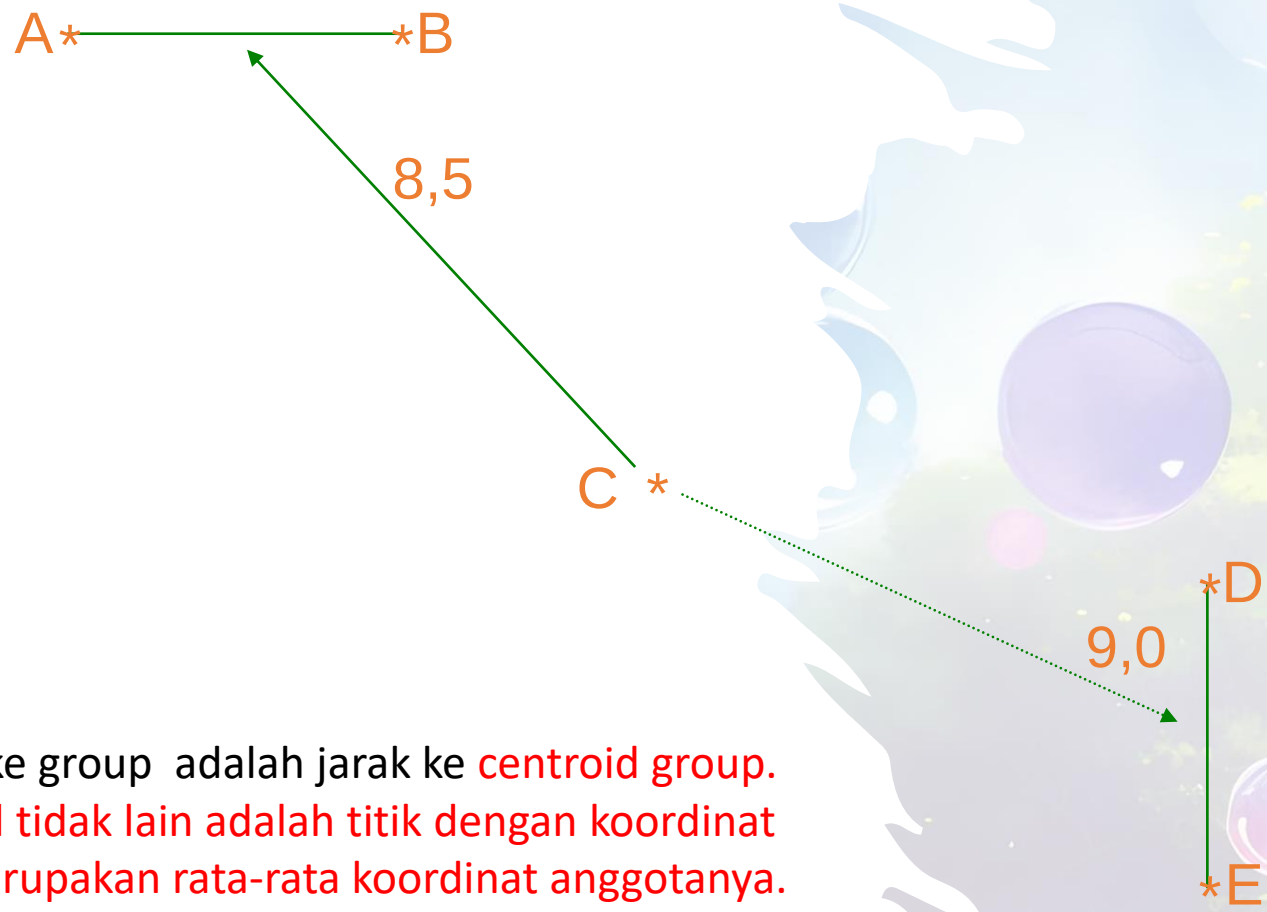
- Jarak C ke group adalah jarak ke titik terdekat
- Complete linkage dikenal juga sebagai nearest neighbor

Complete



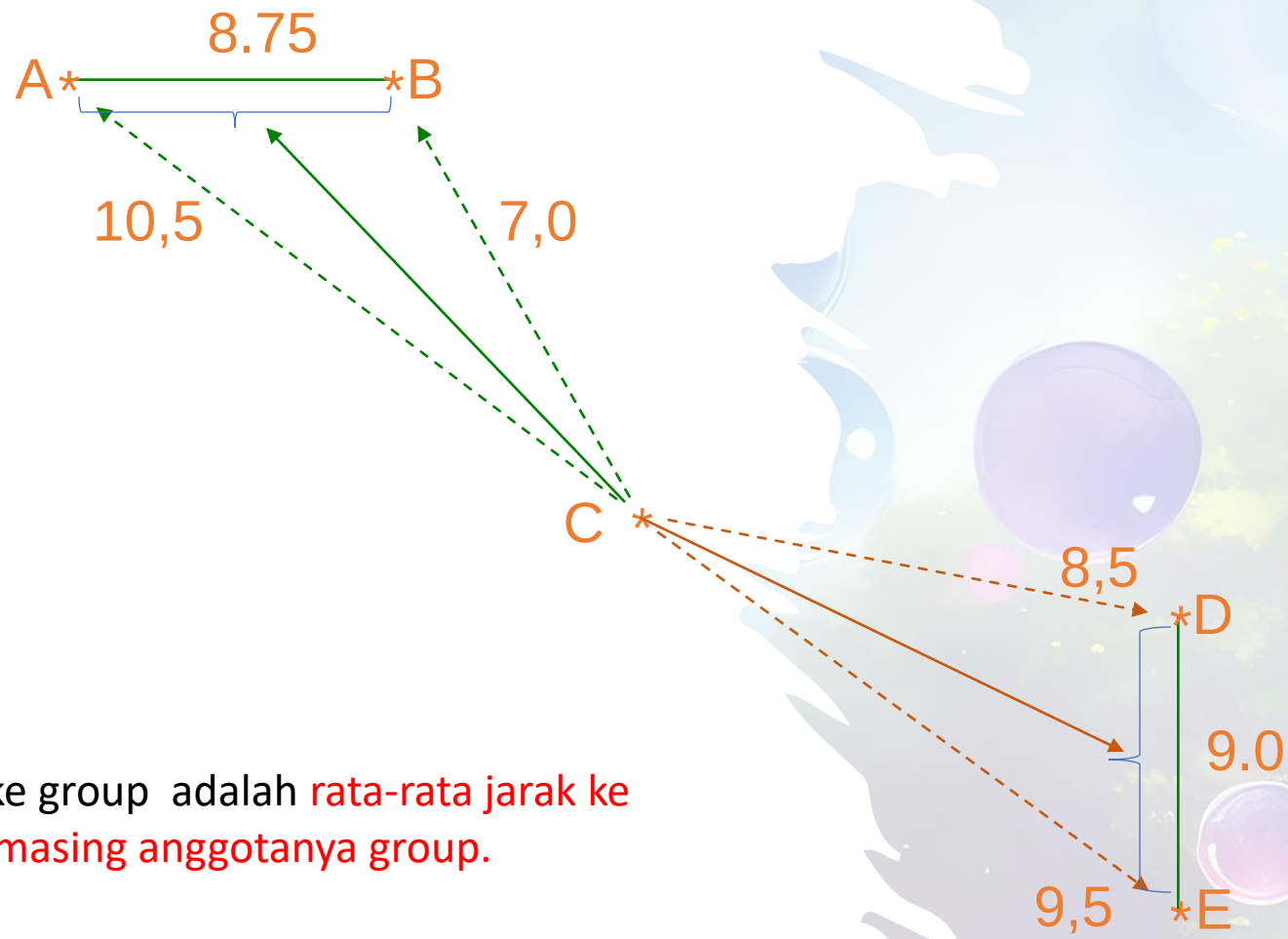
- Jarak C ke group adalah jarak ke titik terjauh
- Complete linkage dikenal juga sebagai furthest neighbor

Centroid linkage



- Jarak C ke group adalah jarak ke centroid group. Centroid tidak lain adalah titik dengan koordinat yang merupakan rata-rata koordinat anggotanya.

Average linkage



- Jarak C ke group adalah rata-rata jarak ke masing-masing anggotanya group.

Evaluasi Penggerombolan

Validasi Penggerombolan

Terdapat beberapa hal yang biasanya dievaluasi:

- *Clustering tendency*
  - Visualisasi, uji hipotesis, misalnya uji Hopkins, SigClust
- Banyaknya gerombol
  - Berdasarkan pengetahuan *domain,* metode elbow, *gap statistics*
- Kualitas penggerombolan
  - Ekstrinsik : memerlukan label "*ground truth*"
  - Instrinsik : tidak perlu label
    - Indeks Calinski-Harabasz, Davies-Bouldin, Koefisien Silhouette

Kriteria Berbasis Rasio Keragaman Intra/Antar - Gerombol (*Within/Between Clusters*) : Indeks Calinski-Harabasz (CH)

- The criterion is based on balancing the within-cluster variation:

$$W_{C_K} = \sum_{j=1}^{K} \sum_{c(i)=j} (x_i - \bar{x}_j)(x_i - \bar{x}_j)^t$$

- Against the between-cluster variation:

$$B_{C_K} = \sum_{j=1}^{K} n_j (x_j - \bar{x}\ \ )(x_j - \bar{x}\ \ )^t$$

- CH's variance ration criterion:

$$CH(C_K) = \frac{trace(B_{C_K})}{trace(W_{C_K})} \times \frac{n-K}{K-1}$$

Kriteria Berbasis Rasio Keragaman Intra/Antar - Gerombol (*Within/Between Clusters*) : Indeks Calinski-Harabasz (CH)

- Indeks CH mengukur keragaman antar-gerombol terhadap keragaman di dalam gerombol.
- Nilai indeks CH yang lebih tinggi menunjukkan penggerombolan yang lebih baik.
- Rentang nilai indeks CH > 0, dimana jika **rasio CH lebih besar**, artinya perbedaan antar gerombol lebih besar dibandingkan dengan di dalam gerombol.

Kriteria Berbasis Rasio Keragaman Intra/Antar - Gerombol (*Within/Between Clusters*) : Indeks Calinski-Harabasz (CH)

Beberapa catatan tentang kriteria indeks CH:

- relatif dapat dihitung dengan cepat dan mudah.
- mengasumsikan gerombol berbentuk bulat di sekitar pusat gerombol, dan tidak memperhitungkan pemisahan antar anggota gerombol yang cenderung bernilai ekstrim.
- menganggap gerombol sebagai terpisah jika rataannya jauh berbeda satu sama lain.
- tidak memperhitungkan ukuran dimensi peubah (p) dalam perhitungannya
- pada kasus data berdimensi besar, mungkin kinerjanya tidak sebaik jika dimensinya kecil $(p \leq 10)$
- sebaiknya tidak digunakan untuk membandingkan penggerombolan dengan algoritma berbeda.

Kriteria Berbasis Rasio Keragaman Intra/Antar - Gerombol (*Within/Between Clusters*)

Beberapa pengembangan dari kriteria indeks Calinski-Harabasz:

- Krzanowski dan Lai (1988) → indeks KL
  - Memperhitungkan ukuran dimensi peubah (p)
- Sugar dan James (2003) → indeks SJ
  - Memperhitungkan teori distorsi asimptotik pada sebaran campuran Gaussian
- Halkidi et al. (2000) → indeks validitas SD
  - Memperhitungkan penalti untuk ukuran gerombol (K) yang terlalu besar jika jarak antar rataan gerombol terlalu kecil.

Indeks Davies-Bouldin

Menurut Hennig et al. (2016) pada subbab 26.2.2, kriteria indeks Davies-Bouldin (DB) diperoleh melalui perhitungan berikut.

$$S_k = \left(\frac{1}{n_k} \sum_{c(i)=k} \|\mathbf{x}_i - \bar{\mathbf{x}}_k\|_2^q \right)^{1/q}, \quad M_{ij} = \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|_p$$

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}, \qquad D_i = \max_{j \neq i} R_{ij}$$

$$\mathrm{DB}(\mathcal{C}_K) = \frac{1}{K} \sum_{i=1}^{K} D_i$$

Indeks Davies-Bouldin

- Biasanya digunakan $p = q = 2$, sehingga DB dapat dianggap sebagai kriteria berbasis keragaman.
- Setiap gerombol diberikan bobot yang sama pada perhitungan DB, terlepas dari seberapa besar ukurannya.
- Penggerombolan yang baik akan menghasilkan **nilai indeks DB yang kecil**, karena S_i (dispersi intra-gerombol) akan bernilai kecil sedangkan M_{ij} (jarak antar gerombol) akan bernilai besar.
- Nilai Indeks DB berkisar antara 0 hingga 1, sehingga lebih mudah diinterpretasikan.

Kriteria *Average Silhouette Width (ASW)*

- Kriteria ini berdasarkan kompromi antara homogenitas di dalam gerombol dengan pemisahan antar gerombol, yang mengukur bagaimana kemungkinan suatu titik ditempatkan pada gerombol lain.
- Koefisien Silhouette dapat dihitung dengan formula berikut:

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

where $a_i = \frac{1}{n_k - 1} \sum_{c(j)=k} d(\mathbf{x}_i, \mathbf{x}_i)$ is the average dissimilarity to points of $\mathbf{x}_i$'s own cluster ($n_k = 1 \Rightarrow s_i = 0$) and $b_i = \min_{l \neq k} \frac{1}{n_l} \sum_{c(j)=l} d(\mathbf{x}_i, \mathbf{x}_i)$ is the average dissimilarity to the closest other cluster. With this,

Kriteria *Average Silhouette Width (ASW)*

Sehingga ASW dapat dihitung sebagai rataan dari koefisien Silhouette dari setiap gerombol ke-i

$$\mathrm{ASW}(\mathcal{C}_K) = \frac{1}{n}\sum_{i=1}^{n} s_i$$

Kriteria *Average Silhouette Width (ASW)*

- Jika $b_i \gg a_i$, artinya x_i sudah sangat tepat berada di gerombolnya.
- Koefisien Silhouette berkisar antara -1 dan 1, dimana penggerombolan dianggap baik jika **nilainya semakin besar.**
- Serupa dengan banyak kriteria lain, hasil dari kriteria ini mungkin akan bermasalah pada data dengan amatan pencilan.

Kriteria Evaluasi Lainnya

- Silahkan pelajari pada Hennig et al. (2016) pada bagian subbab 2.6 lainnya.

ILUSTRASI

Data

Sebagai ilustrasi, akan digunakan data fiktif 12 kota, dengan lima peubah sebagai berikut:

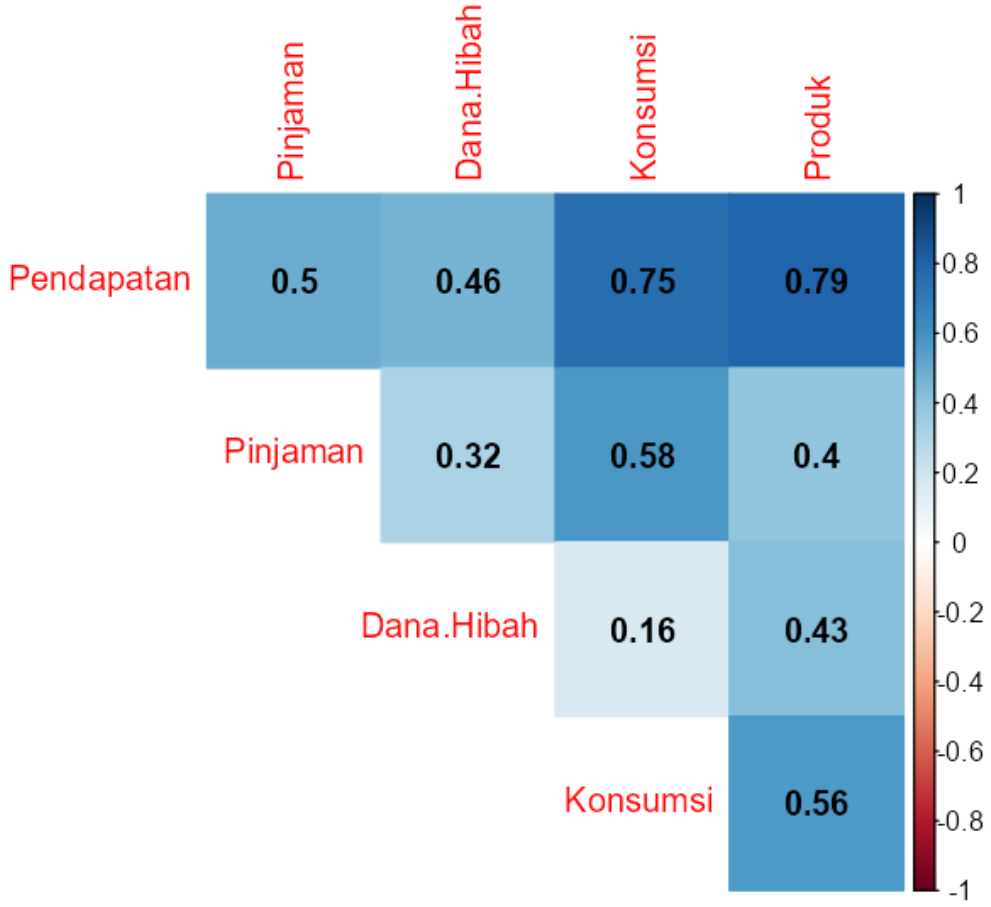
1) Pendapatan kota (trilyun Rupiah)
2) Punjaman pemerintah kota (milyar Rupiah)
3) Dana hibah yang dimiliki kota (milyar Rupiah)
4) Total pengeluaran konsumsi (milyar Rupiah)
5) Banyaknya komoditas produk lokal unggulan

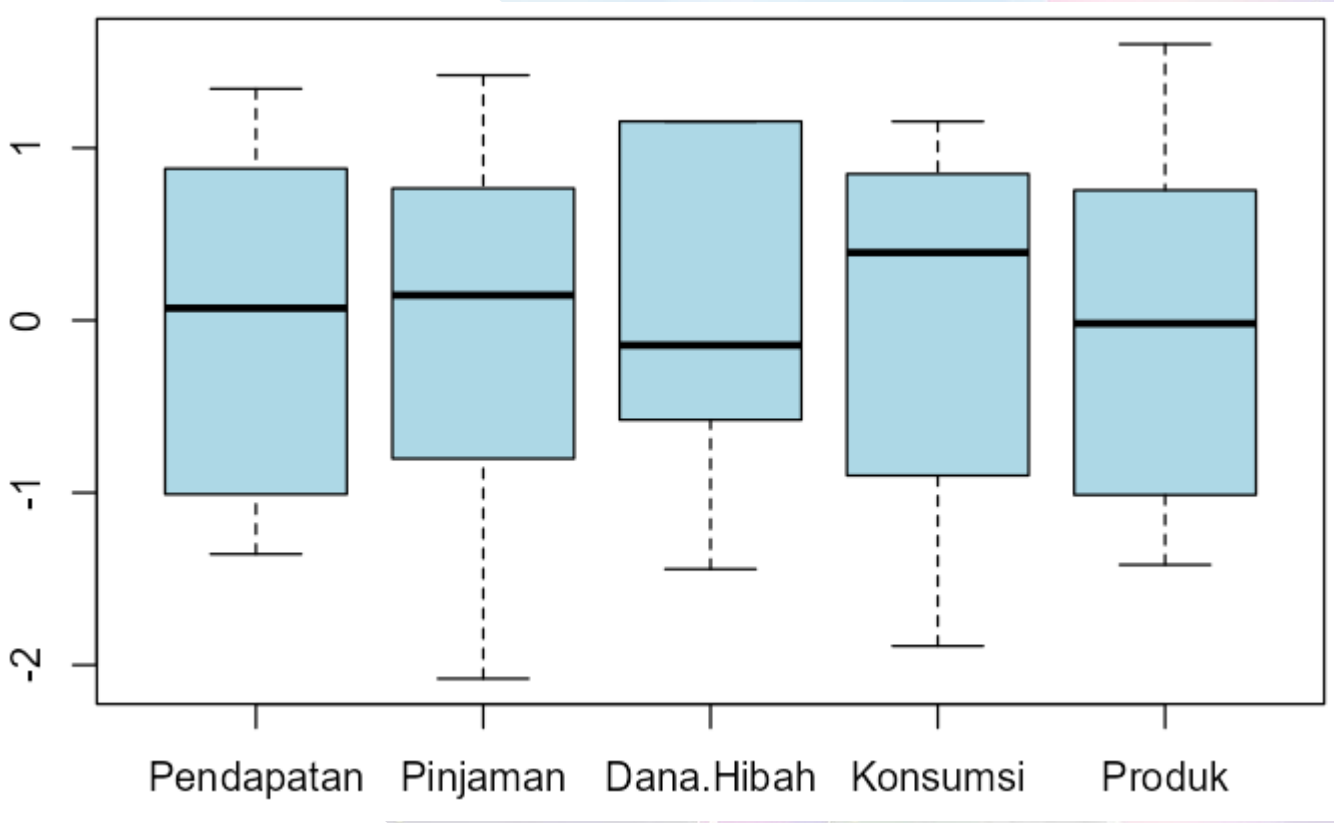
Sumber: https://audhiaprilliant.github.io/assets/docs/Definition%20and%20Procedures%20of%20Cluster%20Analysis.pdf

Data

Kota	Pendapatan	Pinjaman	Dana Hibah	Konsumsi	Produk
Inazuma	55	5.6	9	50	25
Sumeru	61	8	7	62	41
Fontaine	58	3.9	7	60	32
Natlan	67	5.5	7	64	51
Snezhnaya	71	5.7	6	70	42
Teyvat	76	7.6	8	80	29
Guyot	81	8.7	9	80	57
Sakala	56	7.1	6	86	29
Gaia	84	7.6	7	82	46
Mondstat	88	6.5	8	86	52
Kagmaw	84	6.8	9	88	61
Liyue	90	8	9	90	66

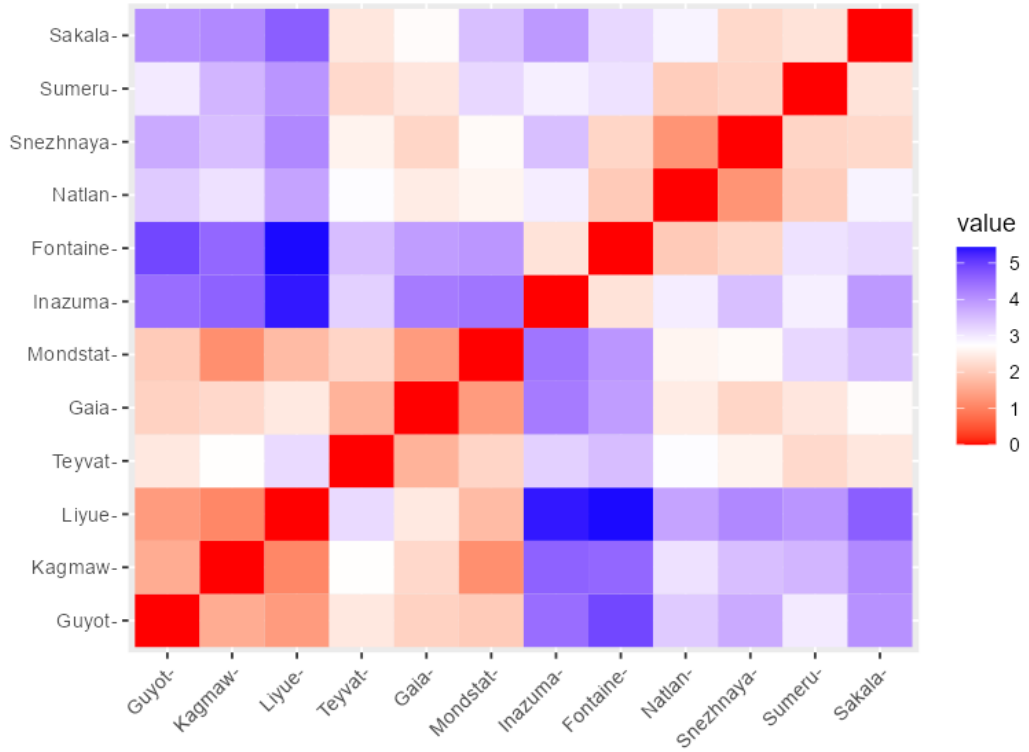
Eksplorasi Data





Identifikasi *Cluster Tendency*

Metode Visual : *Visual Assessment for cluster Tendency (VAT)*



Red → high similarity Blue → low similarity

Uji Hopkins

H_0: data tidak bergerombol

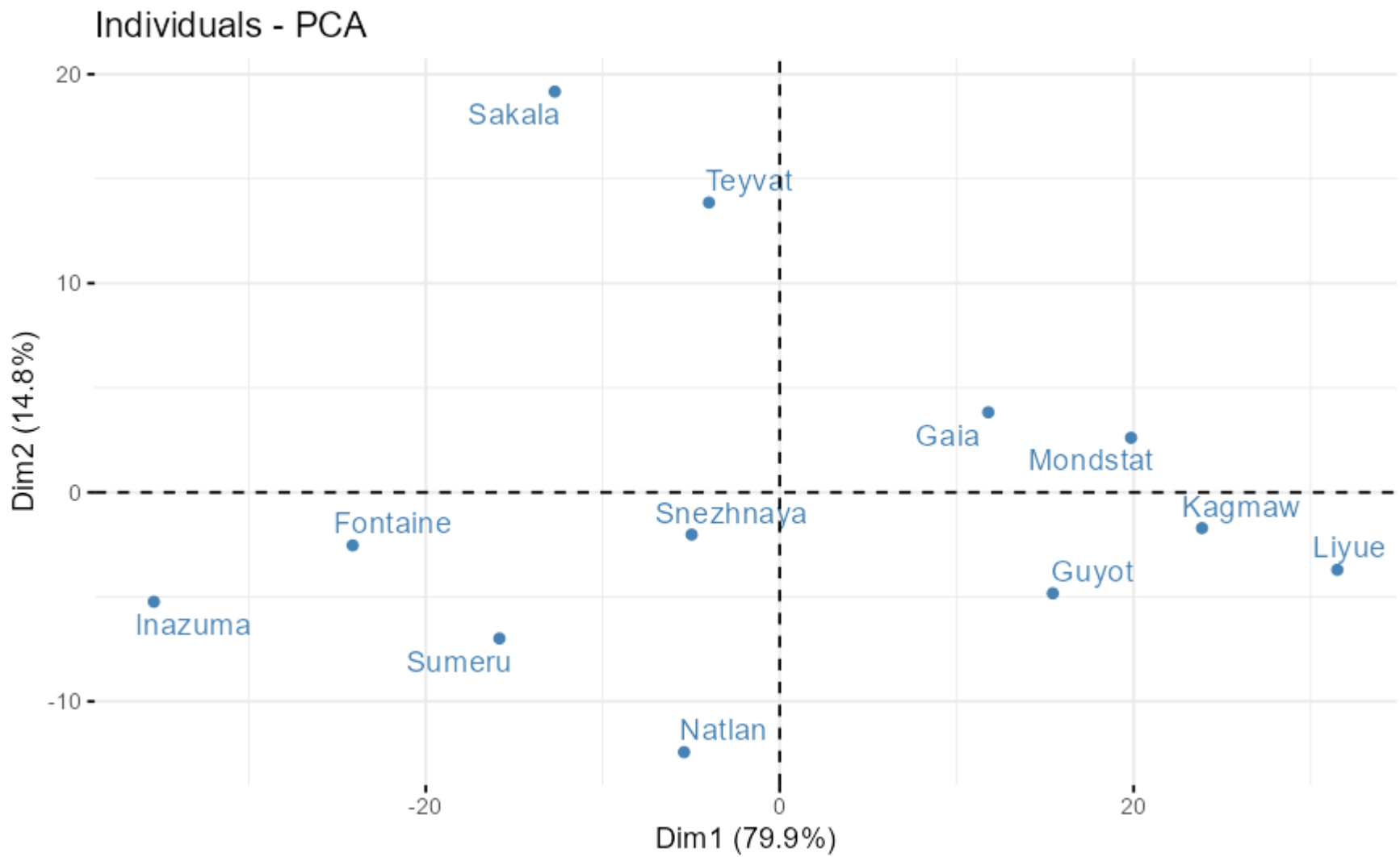
H_1: data bergerombol

$H = 0.0232$

$p - value = 0.0057$

H_0 dapat ditolak pada $\alpha = 5\%$, maka dapat disimpulkan bahwa data memiliki *cluster tendency.*

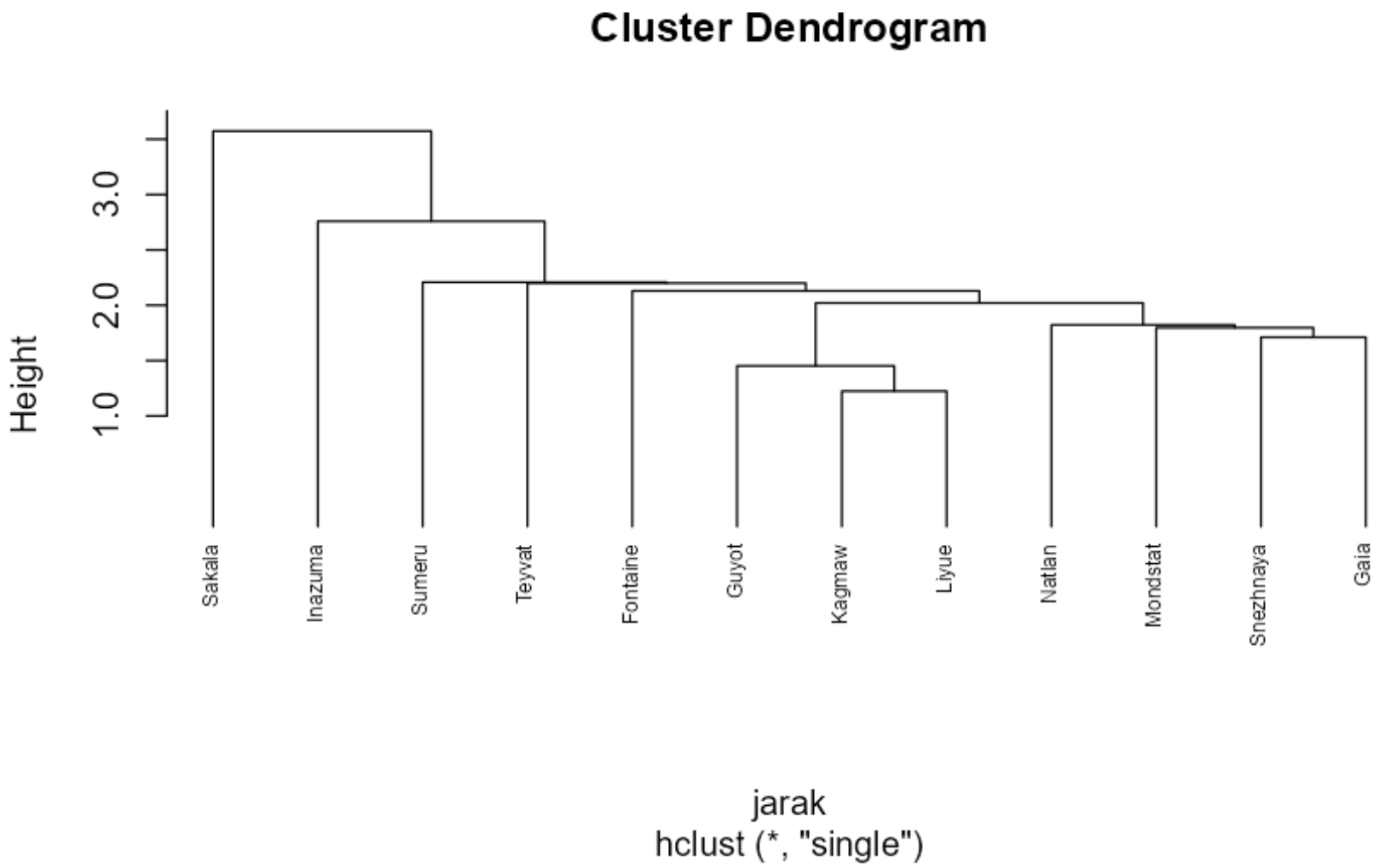
Identifikasi *Cluster Tendency*



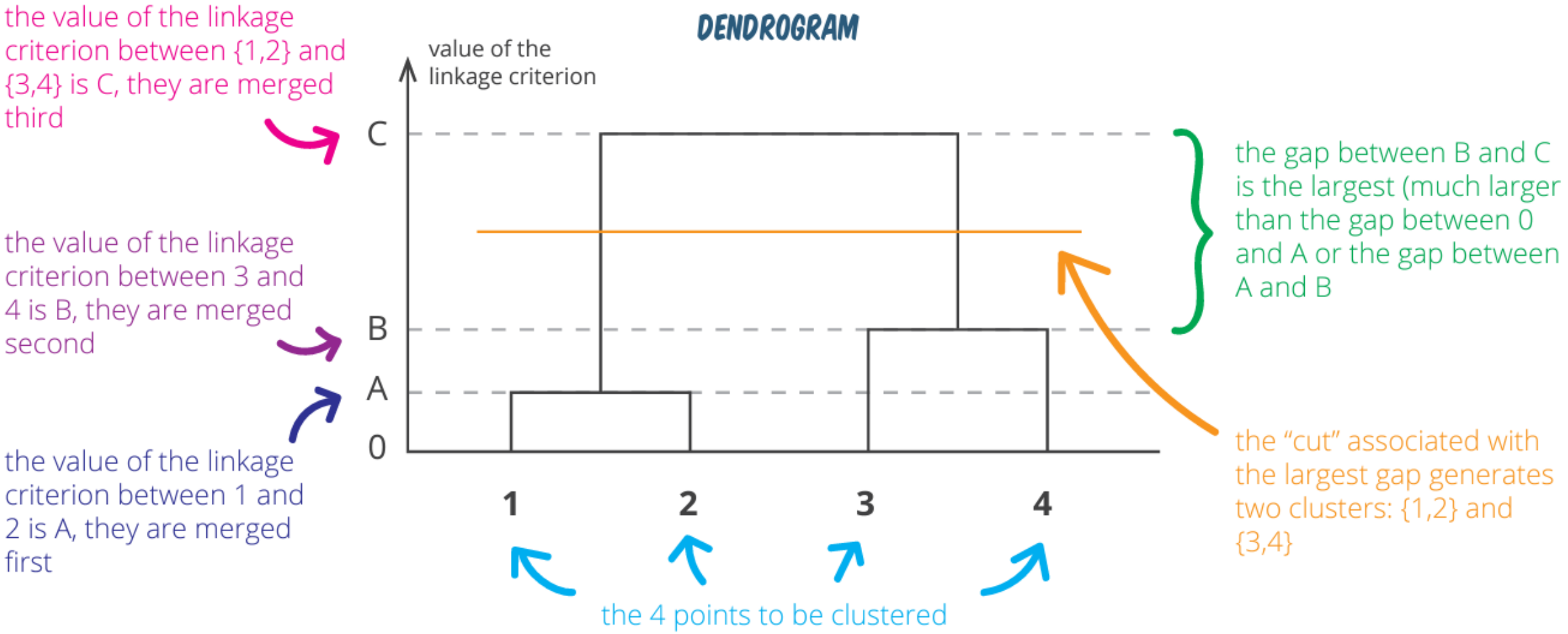
Matriks Jarak Mahalanobis

	Inazuma	Sumeru	Fontaine	Natlan	Snezhnaya	Teyvat	Guyot	Sakala	Gaia	Mondstat	Kagmaw	Liyue
Inazuma	0.00	3.48	2.76	3.45	3.89	3.06	3.12	4.55	3.92	3.57	3.51	3.67
Sumeru	3.48	0.00	3.81	2.58	2.90	3.69	2.21	3.85	2.92	3.82	3.89	3.23
Fontaine	2.76	3.81	0.00	2.13	2.37	3.49	3.68	3.67	3.36	2.55	2.79	3.34
Natlan	3.45	2.58	2.13	0.00	1.82	4.04	2.76	3.94	2.91	2.72	2.67	2.57
Snezhnaya	3.89	2.90	2.37	1.82	0.00	3.24	3.29	3.81	1.71	2.17	3.36	3.10
Teyvat	3.06	3.69	3.49	4.04	3.24	0.00	3.05	3.75	2.20	2.40	3.53	3.30
Guyot	3.12	2.21	3.68	2.76	3.29	3.05	0.00	3.85	2.71	2.78	2.33	1.45
Sakala	4.55	3.85	3.67	3.94	3.81	3.75	3.85	0.00	3.98	3.95	3.57	3.82
Gaia	3.92	2.92	3.36	2.91	1.71	2.20	2.71	3.98	0.00	1.80	3.36	2.72
Mondstat	3.57	3.82	2.55	2.72	2.17	2.40	2.78	3.95	1.80	0.00	2.12	2.02
Kagmaw	3.51	3.89	2.79	2.67	3.36	3.53	2.33	3.57	3.36	2.12	0.00	1.22
Liyue	3.67	3.23	3.34	2.57	3.10	3.30	1.45	3.82	2.72	2.02	1.22	0.00

Penggerombolan Pautan Tunggal

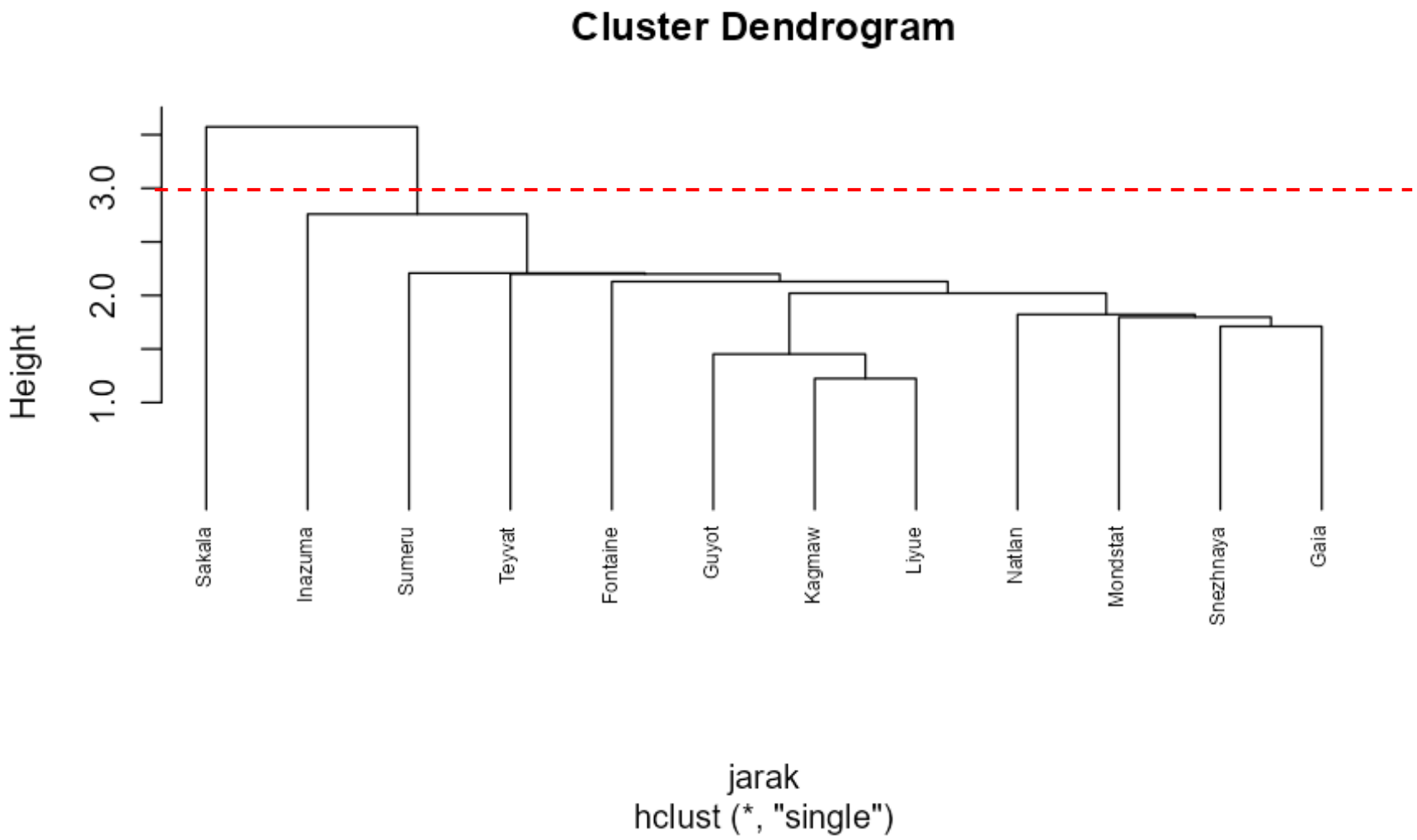


Menentukan Banyaknya Gerombol



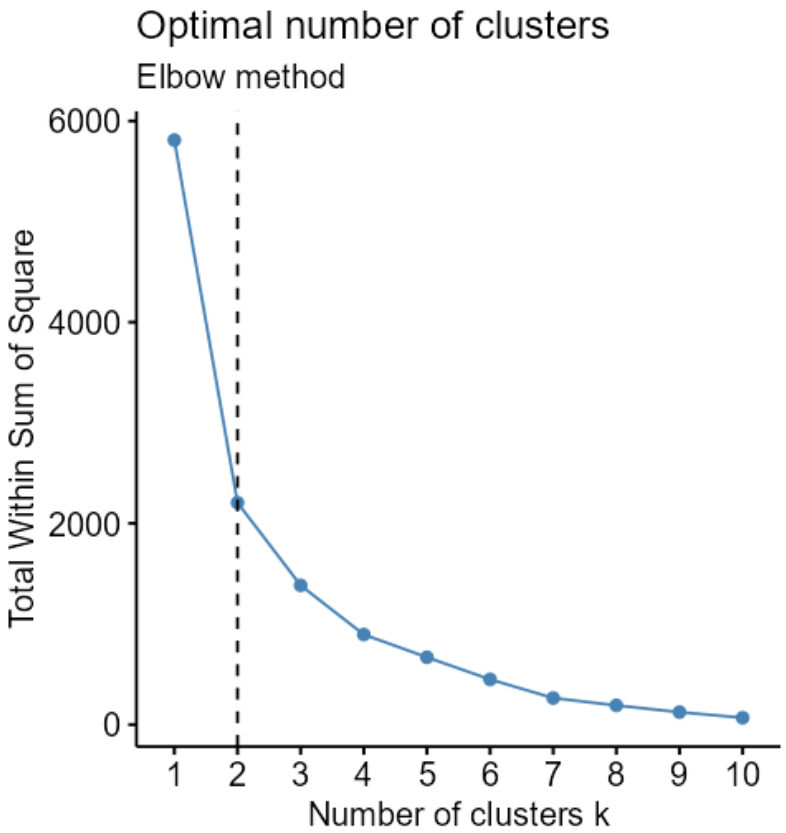
Sumber gambar: https://statsandr.com/blog/files/Hierarchical-clustering-cheatsheet.pdf

Penggerombolan Pautan Tunggal

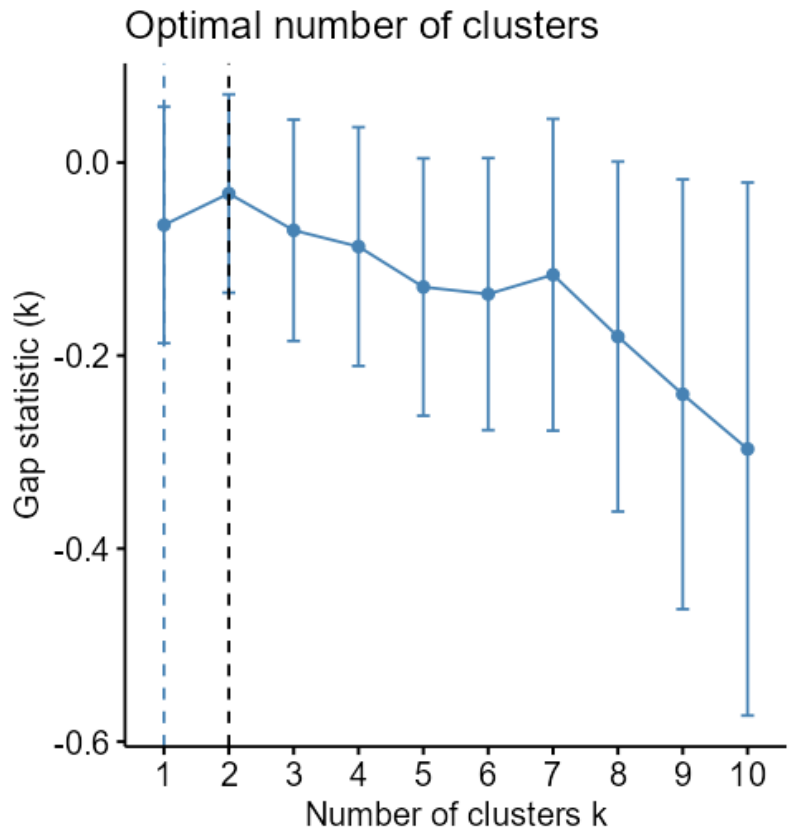


Alternatif Menentukan Banyaknya Gerombol

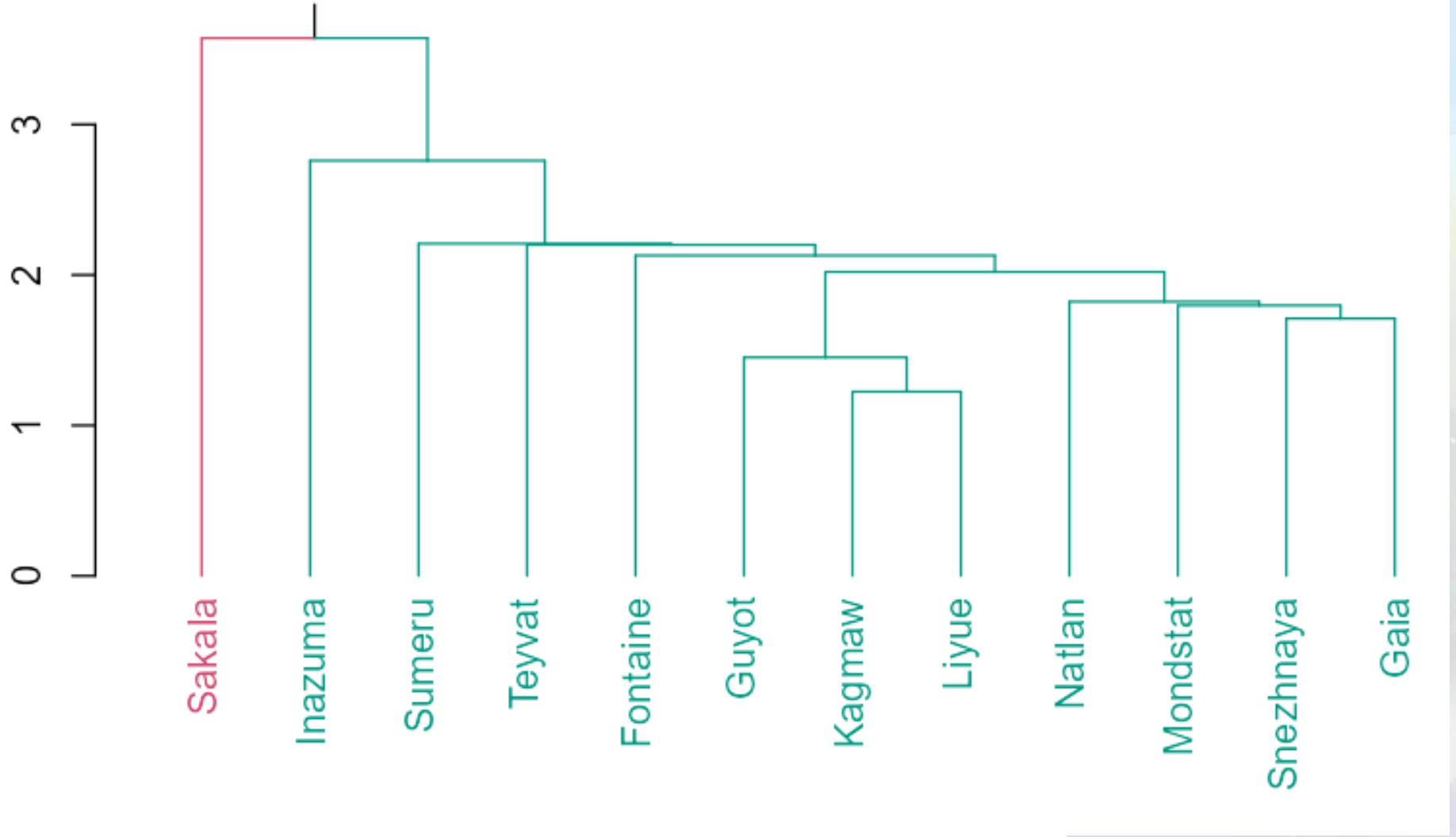
Metode *Elbow*



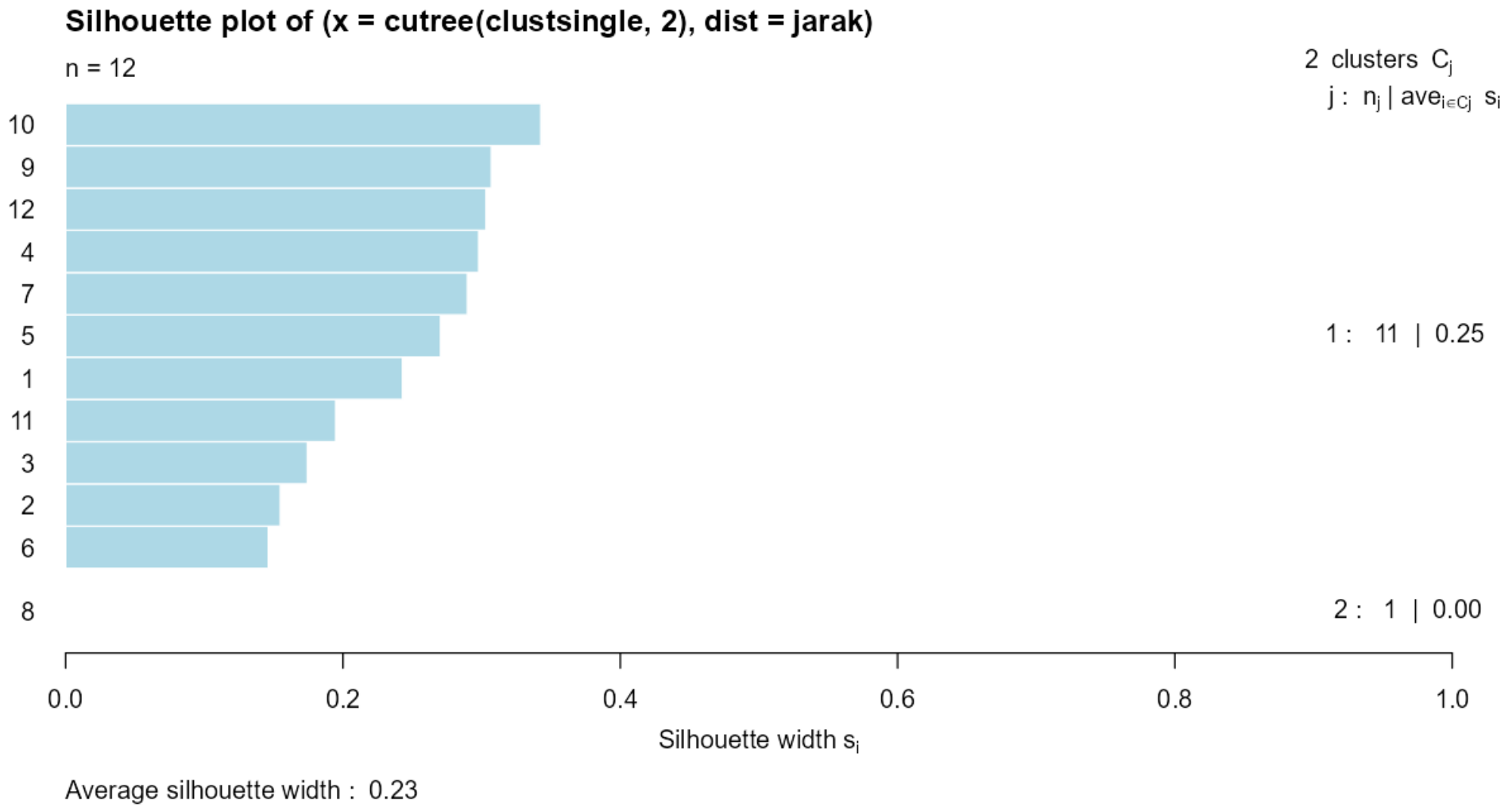
Gap *Statistics*



Hasil Penggerombolan Pautan Tunggal



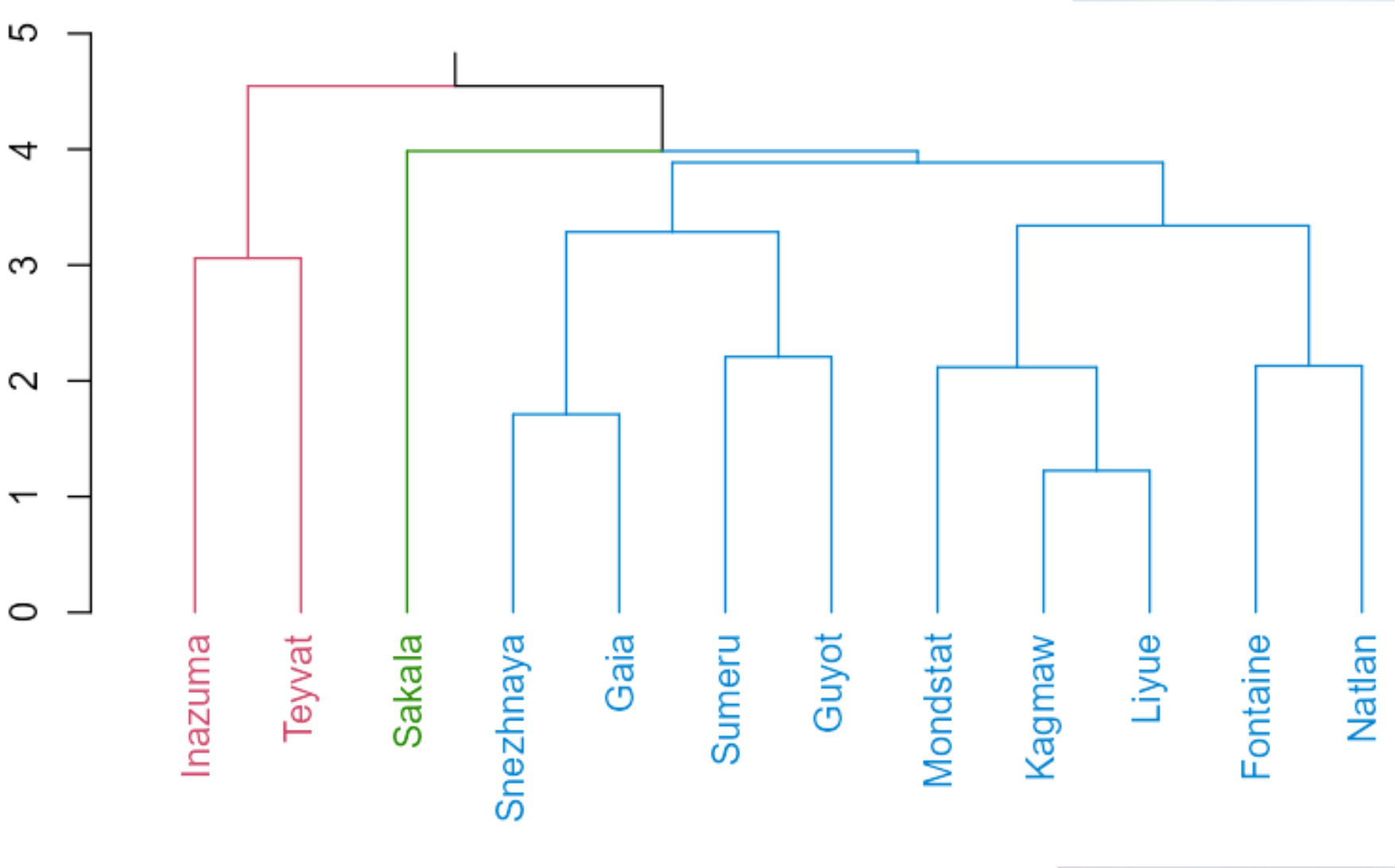
Evaluasi Penggerombolan Pautan Tunggal



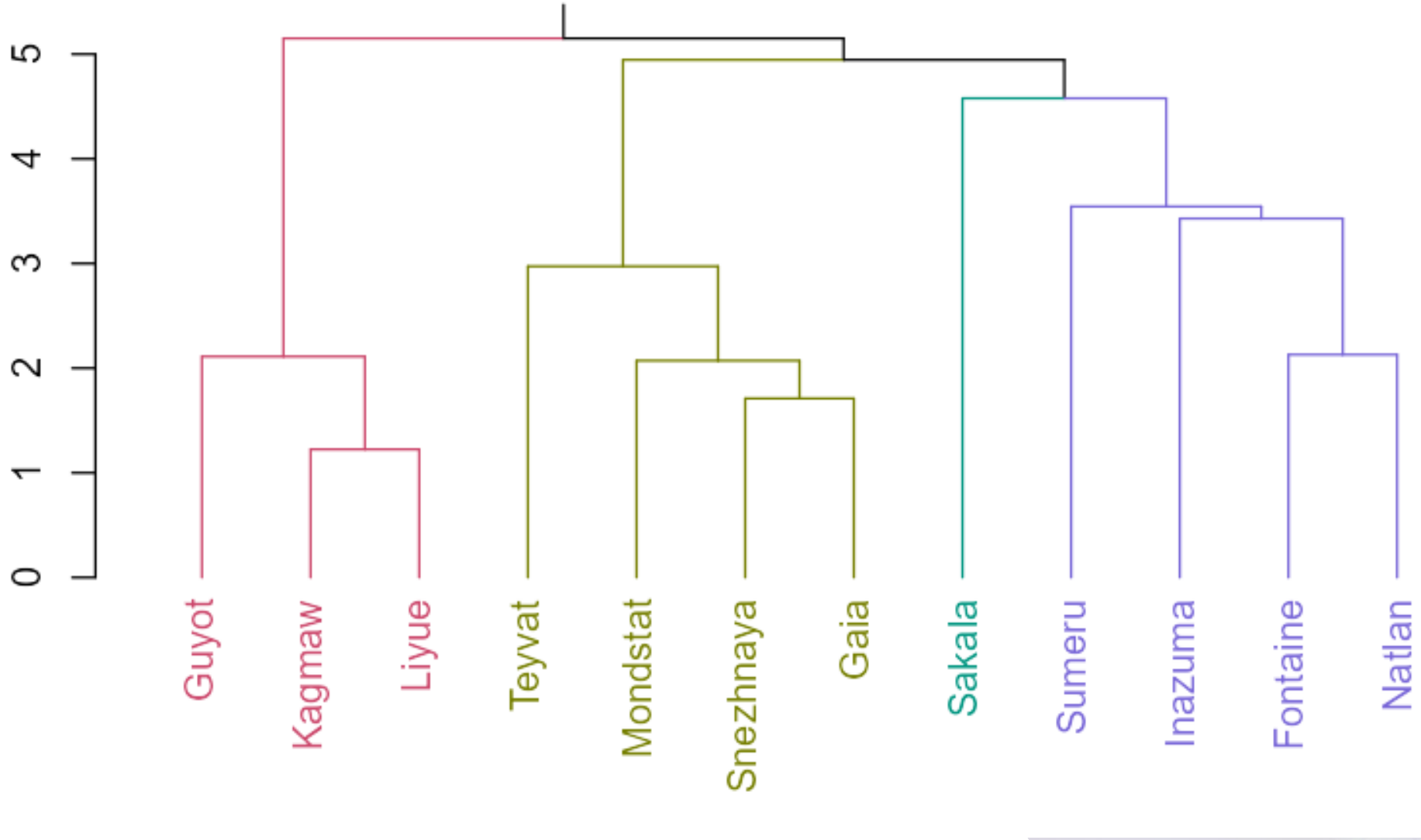
Bahan Diskusi

- Apakah ukuran setiap peubah yang digunakan sama?
- Perhatikan bahwa kita telah menggerombolkan data menggunakan ukuran jarak Mahalanobis tanpa membakukan nilai setiap peubah.
- Apakah hasilnya akan berbeda jika kita membakukan nilai tersebut?
- Bagaimana jika kita menggunakan ukuran jarak Euclidean?

Penggerombolan Pautan Lengkap



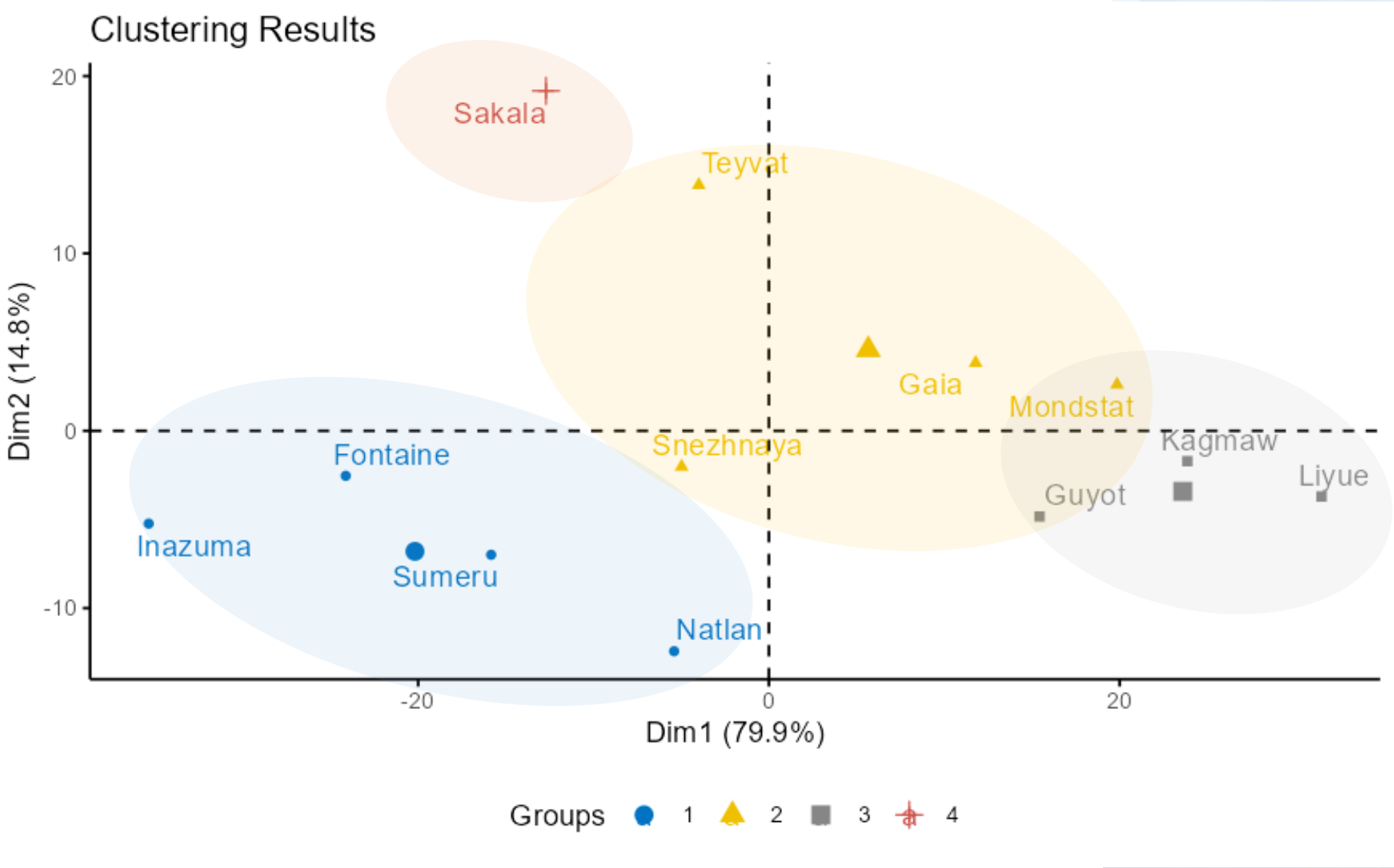
Penggerombolan Metode Ward



Evaluasi Penggerombolan

Metode	# Gerombol	CH	DB	ASW
Pautan Tunggal	2	1.354	0.784	0.23
Pautan Lengkap	3	2.210	1.176	0.16
Metode Ward	4	8.953	0.780	0.17

Hasil Penggerombolan



Karakteristik Gerombol

Rata-rata peubah setiap gerombol:

Gerombol	Pendapatan	Pinjaman	Dana.Hibah	Konsumsi	Produk
1	60.3	5.8	7.5	59.0	37.3
2	79.8	6.9	7.3	79.5	42.3
3	85.0	7.8	9.0	86.0	61.3
4	56.0	7.1	6.0	86.0	29.0

Diskusikan: Bagaimana karakteristik masing-masing gerombol yang terbentuk?

Referensi

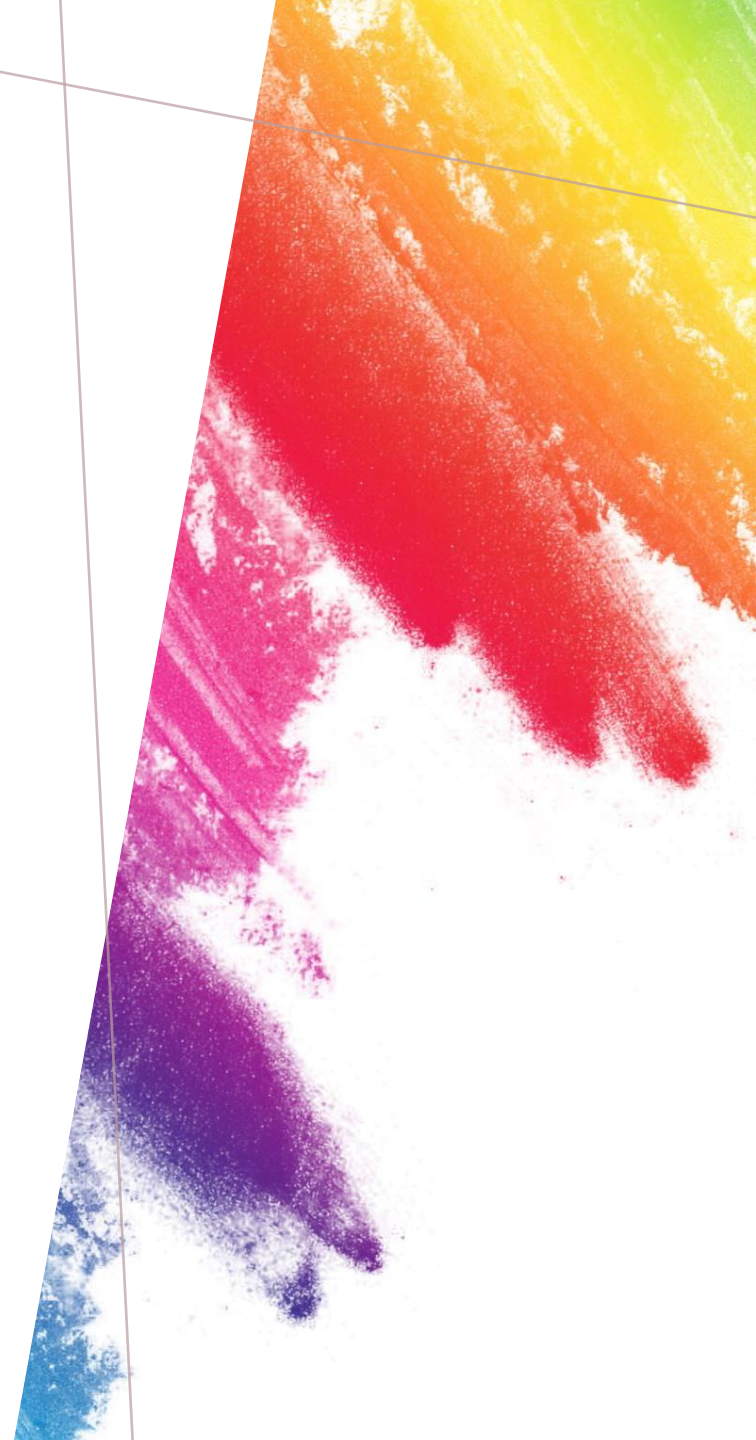
- Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.). (2015). *Handbook of cluster analysis*. CRC press.
- Kassambara, A. (2017). *Practical guide to cluster analysis in R: Unsupervised machine learning* (Vol. 1). Sthda.

Kuliah 9 | Teknik Pembelajaran Mesin

rahmaanisa@apps.ipb.ac.id

METODE PENGGEROMBOLAN DAN EVALUASINYA

[PART 2]



OUTLINE

- Penggerombolan Tak Berhirarki
- Metode K-Means
- Metode K-Medoids
- Ilustrasi

PROSEDUR PENGGEROMBOLAN

- Penggerombolan berhirarki
  - Aglomeratif (dimulai dari *n* gerombol menjadi 1 gerombol)
  - Divisif (dimulai dari 1 gerombol menjadi *n* gerombol)
  - Banyaknya gerombol ditentukan berdasarkan dendogram

- Penggerombolan non-hirarki
  - Banyaknya gerombol yang ingin dibentuk sudah diketahui sejak awal

CLUSTERING PROCEDURES

- **Hierarchical procedures**
  - **Agglomerative** (start from *n* clusters to get to *1* cluster)
  - **Divisive** (start from *1* cluster to get to *n* clusters)

- **Non hierarchical procedures → partitioning algorithms**
  - **K-means clustering**
  - **K-medoids clustering**

NON-HIERARCHICAL CLUSTERING: BASIC CONCEPT

- Partitioning method: Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid or medoid of cluster C_i)

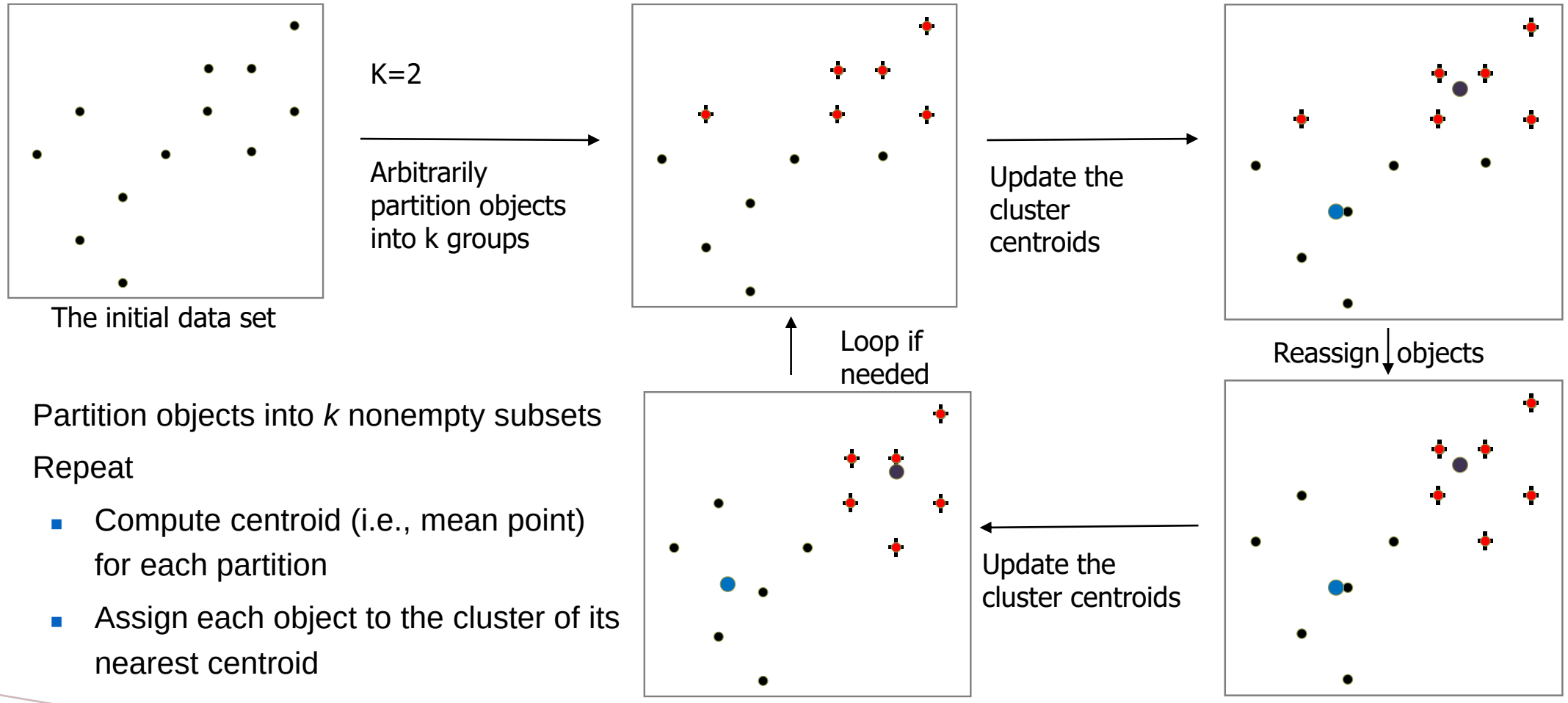
$$E = \Sigma_{i=1}^{k} \Sigma_{p \in C_i} (p - c_i)^2$$

- Given *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: *k-means* and *k-medoids* algorithms
  - *k-means* (MacQueen'67, Lloyd'57/'82): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

THE K-MEANS CLUSTERING METHOD

- Given *k*, the *k-means* algorithm is implemented in four steps:
  - Partition objects into *k* nonempty subsets
  - Compute seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., *mean point*, of the cluster)
  - Assign each object to the cluster with the nearest seed point
  - Go back to Step 2, stop when the assignment does not change

AN EXAMPLE OF K-MEANS CLUSTERING



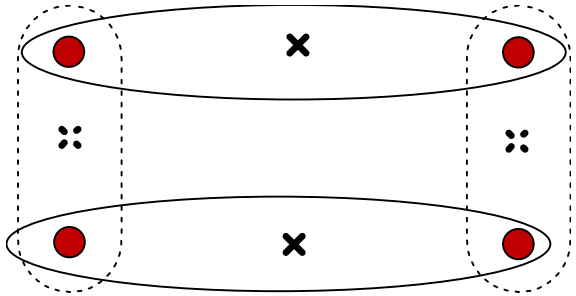
- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

COMMENTS ON THE K-MEANS METHOD

- Strength: *Efficient*: *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.
    - Comparing: PAM: $O(k(n-k)^2)$, CLARA: $O(ks^2 + k(n-k))$
- Comment: Often terminates at a *local optimal*.
- Weakness
    - Applicable only to objects in a continuous n-dimensional space
        - Using the k-modes method for categorical data
        - In comparison, k-medoids can be applied to a wide range of data
    - Need to specify *k,* the *number* of clusters, in advance (there are ways to automatically determine the best k (see Hastie et al., 2009)
    - Sensitive to noisy data and *outliers*
    - Not suitable to discover clusters with *non-convex shapes*

VARIATIONS OF THE K-MEANS METHOD

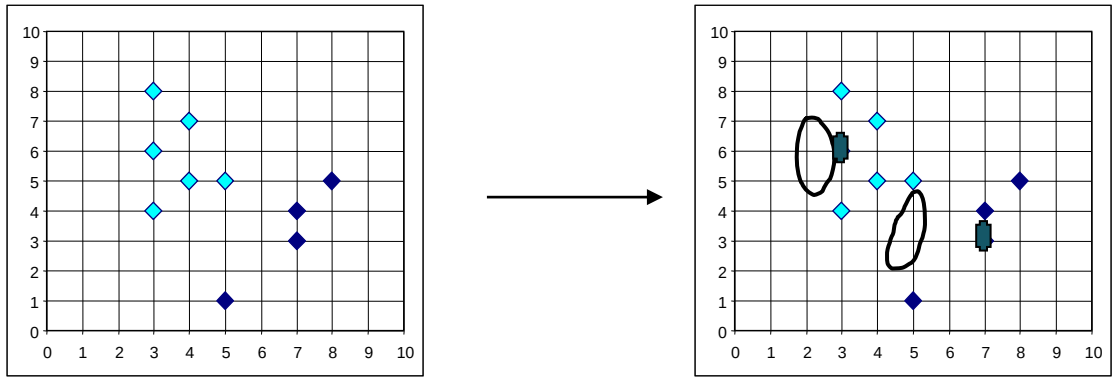
- Most of the variants of the *k-means* which differ in
  - Selection of the initial *k* means
  - Dissimilarity calculations
  - Strategies to calculate cluster means



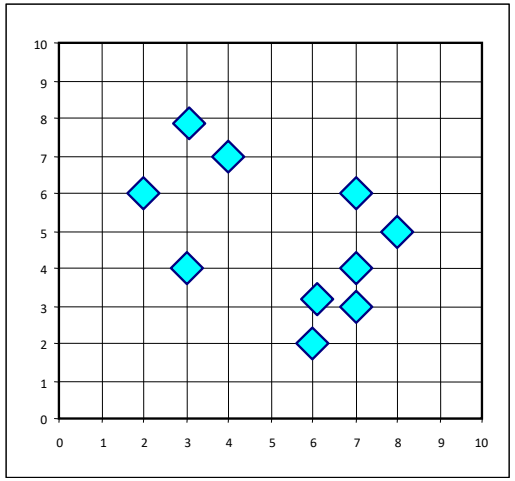
- Handling categorical data: *k-modes*
  - Replacing means of clusters with modes
  - Using new dissimilarity measures to deal with categorical objects
  - Using a frequency-based method to update modes of clusters
  - A mixture of categorical and numerical data: *k-prototype* method

WHAT IS THE PROBLEM OF THE K-MEANS METHOD?

- The k-means algorithm is sensitive to outliers !
  - Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster

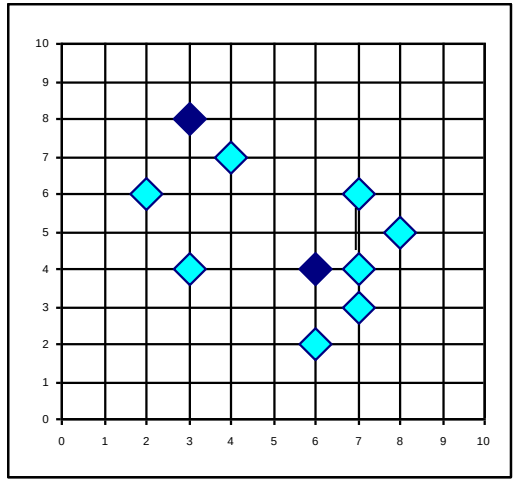


PAM: A TYPICAL K-MEDOIDS ALGORITHM



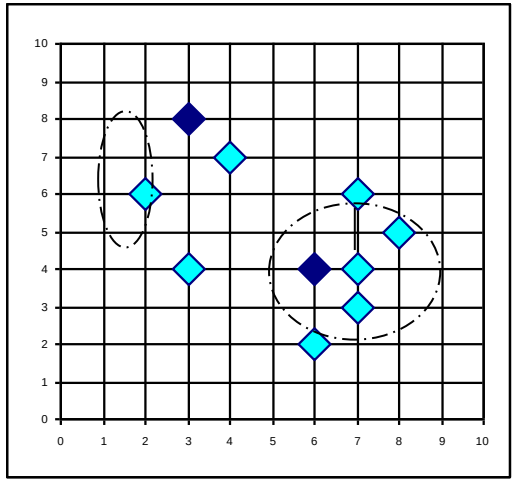
K=2

Arbitrary choose k object as initial medoids

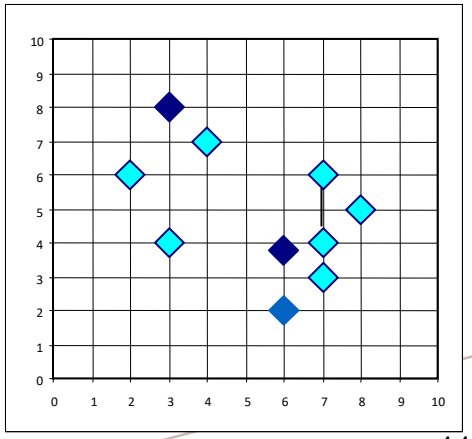


Assign each remaining object to nearest medoids

Total Cost = 20

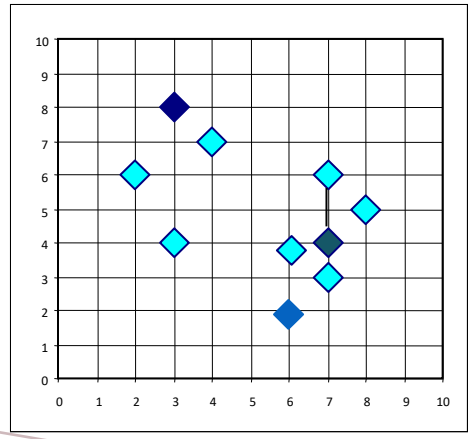


Randomly select a nonmedoid object, O_{ramdom}



Compute total cost of swapping

Total Cost = 26



Swapping O and O_{ramdom}

If quality is improved.

Do loop

Until no change

THE K-MEDOID CLUSTERING METHOD

- *K-Medoids* Clustering: Find *representative* objects (medoids) in clusters
  - *PAM* (Partitioning Around Medoids, Kaufmann & Rousseeuw 1987)
    - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering
    - *PAM* works effectively for small data sets, but does not scale well for large data sets (due to the computational complexity)
- Efficiency improvement on PAM
  - *CLARA* (Kaufmann & Rousseeuw, 1990): PAM on samples
  - *CLARANS* (Ng & Han, 1994): Randomized re-sampling

Other Crisp Clustering Methods

- See Hennig et al. (2015) chapter 4.4.

ILUSTRASI

ILUSTRASI (1) : DATA

Sebagai ilustrasi, akan digunakan data fiktif 12 kota, dengan lima peubah sebagai berikut:

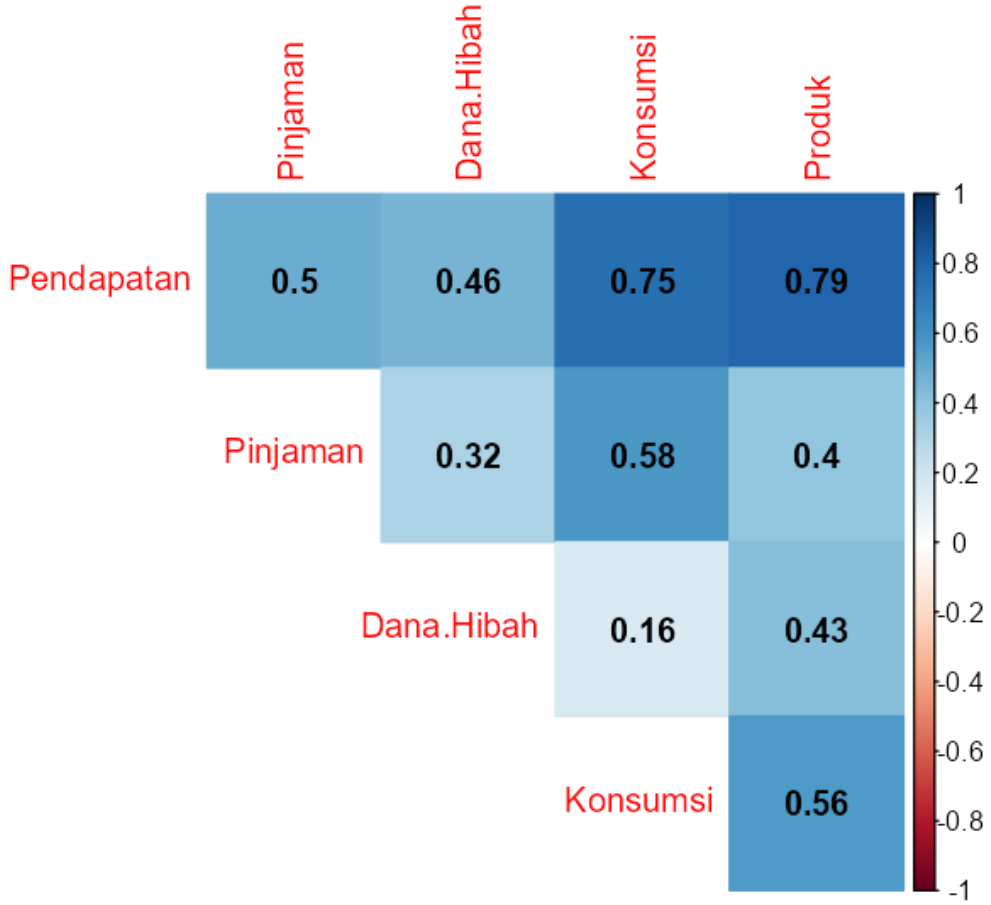
1) Pendapatan kota (trilyun Rupiah)
2) Pinjaman pemerintah kota (milyar Rupiah)
3) Dana hibah yang dimiliki kota (milyar Rupiah)
4) Total pengeluaran konsumsi (milyar Rupiah)
5) Banyaknya komoditas produk lokal unggulan

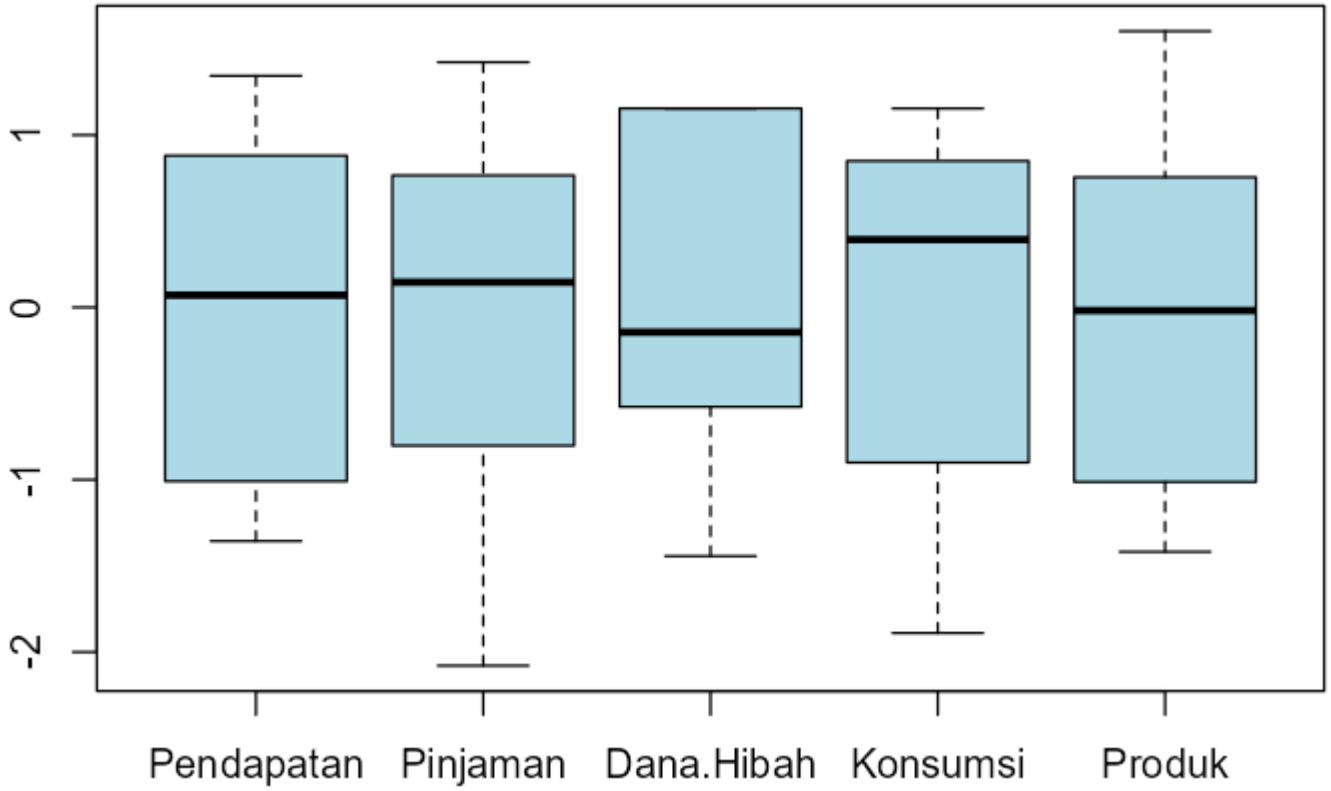
Sumber: https://audhiaprilliant.github.io/assets/docs/Definition%20and%20Procedures%20of%20Cluster%20Analysis.pdf

DATA

Kota	Pendapatan	Pinjaman	Dana Hibah	Konsumsi	Produk
Inazuma	55	5.6	9	50	25
Sumeru	61	8	7	62	41
Fontaine	58	3.9	7	60	32
Natlan	67	5.5	7	64	51
Snezhnaya	71	5.7	6	70	42
Teyvat	76	7.6	8	80	29
Guyot	81	8.7	9	80	57
Sakala	56	7.1	6	86	29
Gaia	84	7.6	7	82	46
Mondstat	88	6.5	8	86	52
Kagmaw	84	6.8	9	88	61
Liyue	90	8	9	90	66

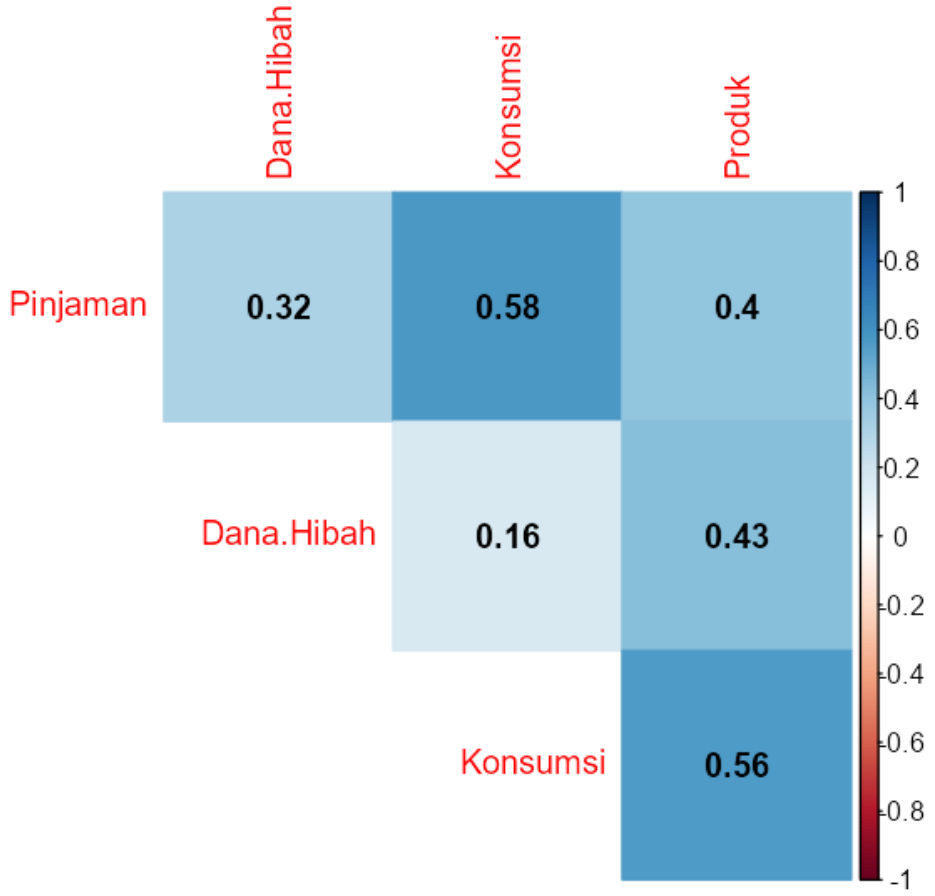
EKSPLORASI DATA



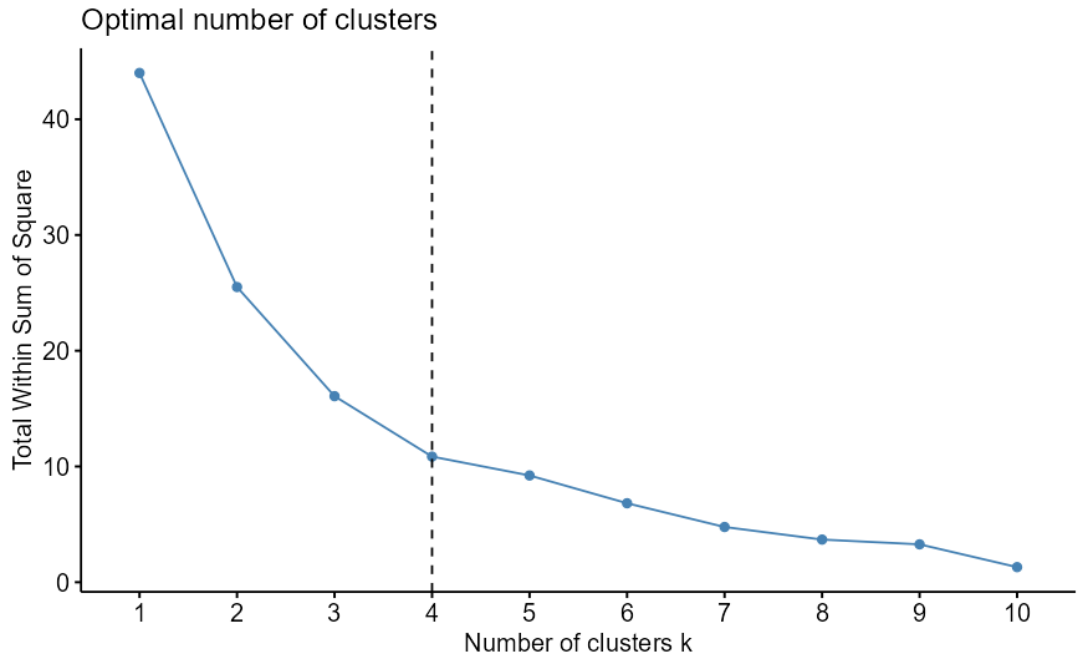


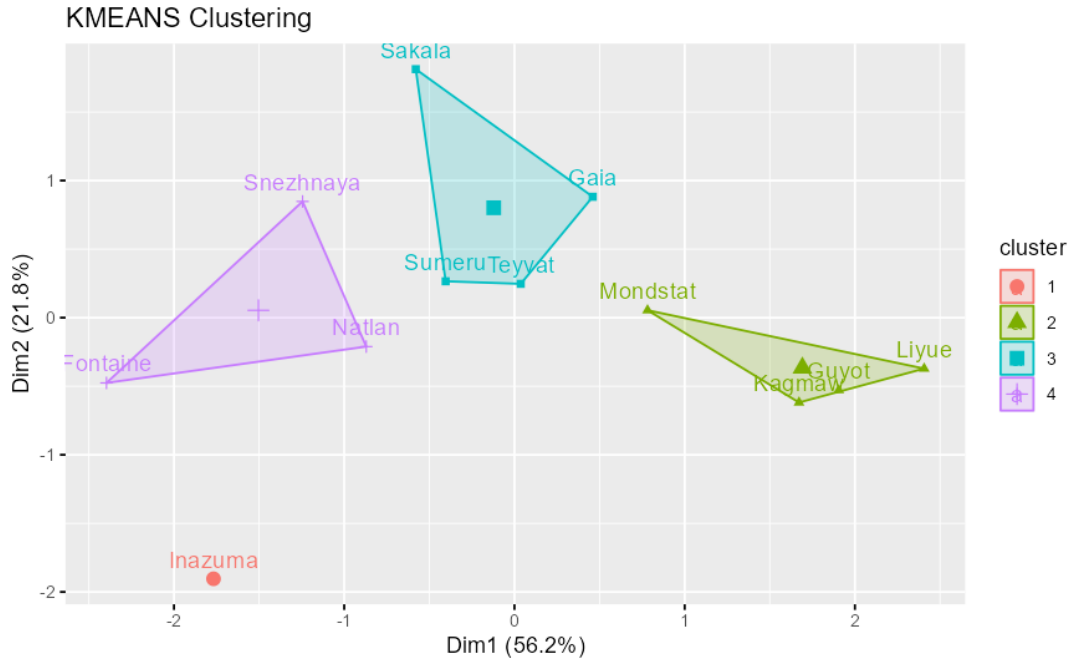
SELEKSI PEUBAH

1) ~~Pendapatan kota (trilyun Rupiah)~~
2) Pinjaman pemerintah kota (milyar Rupiah)
3) Dana hibah yang dimiliki kota (milyar Rupiah)
4) Total pengeluaran konsumsi (milyar Rupiah)
5) Banyaknya komoditas produk lokal unggulan

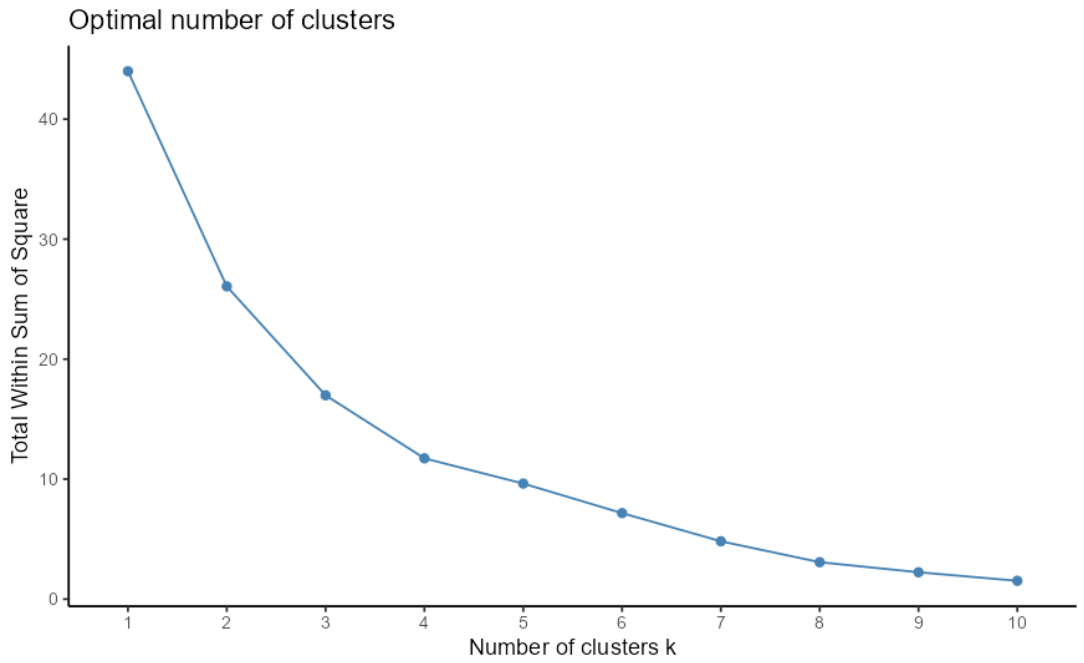


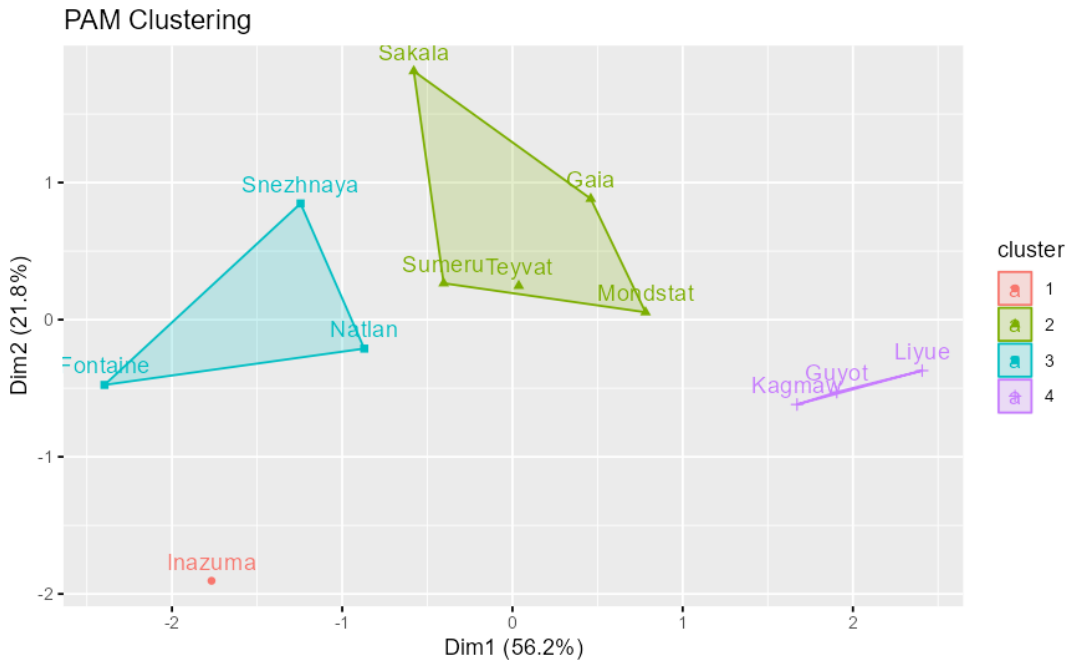
PENGGEROMBOLAN K-MEANS





PENGGEROMBOLAN K-MEDOIDS





EVALUASI PENGGEROMBOLAN

Metode	CH	DB	ASW
K-Means	8.291	1.019	0.296
K-Medoids	6.482	1.027	0.265
	(Terbesar)	(Terkecil)	(Terbesar)

KARAKTERISTIK GEROMBOL

Berikut ini adalah rataan peubah setiap gerombol:

Gerombol	Pinjaman	Dana Hibah	Konsumsi	Produk
1	5.60	9.00	50.00	25.00
2	7.50	8.75	86.00	59.00
3	7.58	7.00	77.50	36.25
4	5.03	6.67	64.67	41.67

Program R tersedia di: https://rpubs.com/r_anisa/kmeans-kmedoids

LATIHAN: DATA PELANGGAN

Data berukuran 18 kolom dan 8636 baris dengan ringkasan sebagai berikut:

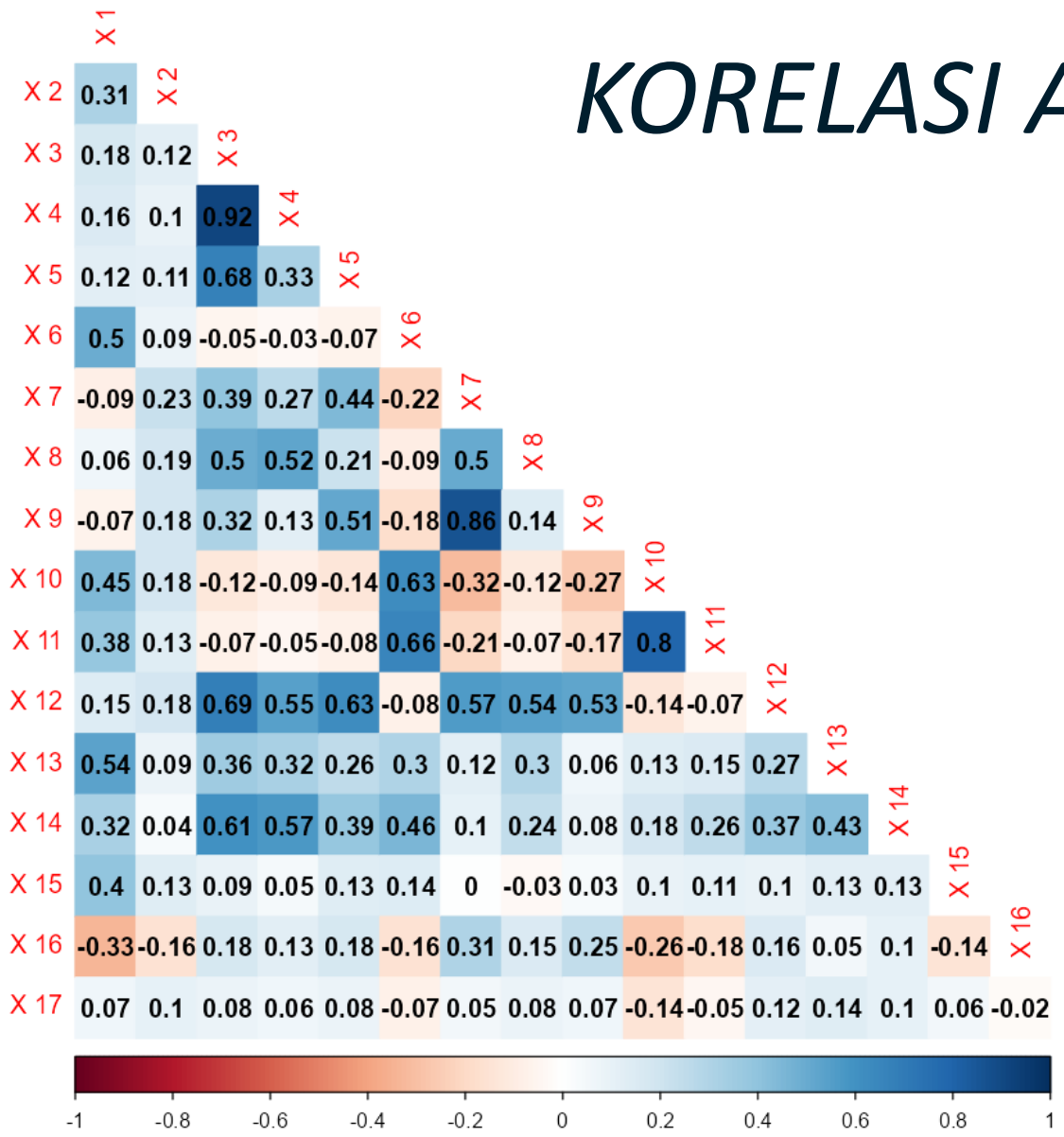
Peubah	Tipe	Nilai
CUST_ID	<chr>	"C10001", "C10002", "C10003", "C10005", "C10006", "C10007", "C10008", "…
BALANCE	<dbl>	40.90075, 3202.46742, 2495.14886, 817.71434, 1809.82875, 627.26081, 182…
BALANCE_FREQUENCY	<dbl>	0.818182, 0.909091, 1.000000, 1.000000, 1.000000, 1.000000, 1.000000, 1…
PURCHASES	<dbl>	95.40, 0.00, 773.17, 16.00, 1333.28, 7091.01, 436.20, 861.49, 1281.60, …
ONEOFF_PURCHASES	<dbl>	0.00, 0.00, 773.17, 16.00, 0.00, 6402.63, 0.00, 661.49, 1281.60, 0.00, …
INSTALLMENTS_PURCHASES	<dbl>	95.40, 0.00, 0.00, 0.00, 1333.28, 688.38, 436.20, 200.00, 0.00, 920.12,…
CASH_ADVANCE	<dbl>	0.00000, 6442.94548, 0.00000, 0.00000, 0.00000, 0.00000, 0.00000, 0.000…
PURCHASES_FREQUENCY	<dbl>	0.166667, 0.000000, 1.000000, 0.083333, 0.666667, 1.000000, 1.000000, 0…
ONEOFF_PURCHASES_FREQUENCY	<dbl>	0.000000, 0.000000, 1.000000, 0.083333, 0.000000, 1.000000, 0.000000, 0…
PURCHASES_INSTALLMENTS_FREQUENCY	<dbl>	0.083333, 0.000000, 0.000000, 0.000000, 0.583333, 1.000000, 1.000000, 0…
CASH_ADVANCE_FREQUENCY	<dbl>	0.000000, 0.250000, 0.000000, 0.000000, 0.000000, 0.000000, 0.000000, 0…
CASH_ADVANCE_TRX	<int>	0, 4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 4, 3, 0, 0, 0, 0, 6, 0, 13, 4…
PURCHASES_TRX	<int>	2, 0, 12, 1, 8, 64, 12, 5, 3, 12, 6, 26, 26, 0, 11, 0, 8, 9, 12, 8, 92,…
CREDIT_LIMIT	<dbl>	1000, 7000, 7500, 1200, 1800, 13500, 2300, 7000, 11000, 1200, 2000, 300…
PAYMENTS	<dbl>	201.8021, 4103.0326, 622.0667, 678.3348, 1400.0578, 6354.3143, 679.0651…
MINIMUM_PAYMENTS	<dbl>	139.50979, 1072.34022, 627.28479, 244.79124, 2407.24604, 198.06589, 532…
PRC_FULL_PAYMENT	<dbl>	0.000000, 0.222222, 0.000000, 0.000000, 0.000000, 1.000000, 0.000000, 0…
TENURE	<int>	12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 12, 8, 12, 12, …

PEUBAH PENGGEROMBOLAN

Peubah	Keterangan
X1	BALANCE
X2	BALANCE_FREQUENCY
X3	PURCHASES
X4	ONEOFF_PURCHASES
X5	INSTALLMENTS_PURCHASES
X6	CASH_ADVANCE
X7	PURCHASES_FREQUENCY
X8	ONEOFF_PURCHASES_FREQUENCY
X9	PURCHASES_INSTALLMENTS_FREQUENCY

Peubah	Keterangan
X10	CASH_ADVANCE_FREQUENCY
X11	CASH_ADVANCE_TRX
X12	PURCHASES_TRX
X13	CREDIT_LIMIT
X14	PAYMENTS
X15	MINIMUM_PAYMENTS
X16	PRC_FULL_PAYMENT
X17	TENURE

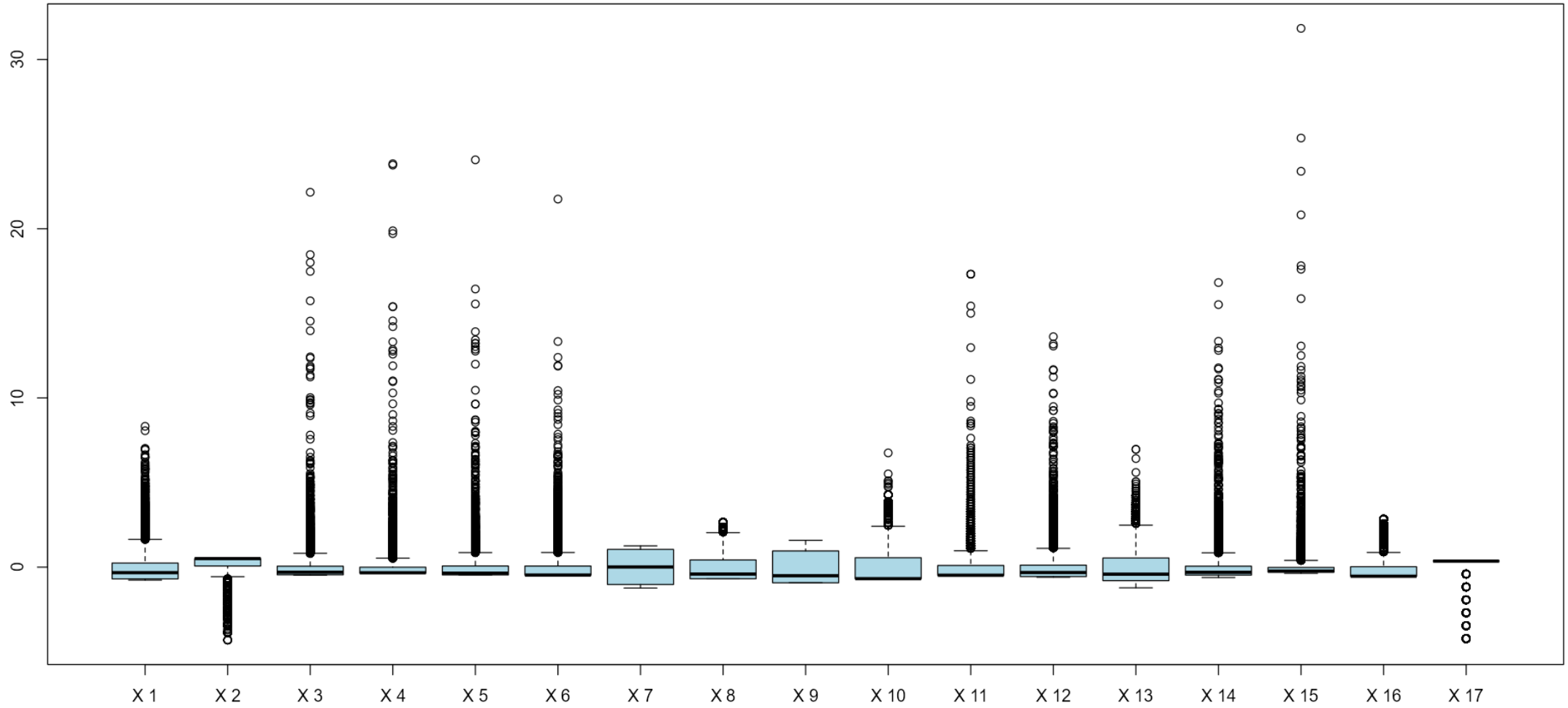
KORELASI ANTAR PEUBAH



Terdapat beberapa peubah yang memiliki korelasi linier cukup tinggi:

- X3 dan X4 → r=0.92
- X7 dan X9 → r=0.86
- X10 dan X11 → r=0.80

EKSPLORASI DATA



DISKUSI KELOMPOK

- Berdasarkan hasil eksplorasi data pelanggan, lakukan prosedur analisis gerombol yang menurut Anda paling tepat.
- Tuliskan ringkasan dari hasil analisis yang telah Anda lakukan, sesuai dengan hasil diskusi kelompok Anda.
- Diskusikan hasil yang Anda peroleh di depan kelas.

REFERENCES

- Hennig, C., Meila, M., Murtagh, F., & Rocci, R. (Eds.). (2015). *Handbook of cluster analysis*. CRC press.
- Kassambara, A. (2017). *Practical guide to cluster analysis in R: Unsupervised machine learning* (Vol. 1). Sthda.
- Other relevant sources.

Reduksi Dimensi dengan Analisis Komponen Utama

Kuliah 10 | STA1382

rahmaanisa@apps.ipb.ac.id

Outline

- Reduksi dimensi
- Analisis Komponen Utama
- Ilustrasi

Reduksi Dimensi

Basic Idea

Student	Height (cm)	Weight (kg)
A	180	80
B	175	75
C	170	60
D	160	80
E	150	45
F	155	48
G	160	60
H	162	59
I	165	60
J	150	80

Univariate

- ✓ Who's the tallest?
- ✓ Who's the shortest?
- ✓ Who's the most weight?

Bivariate – among objects

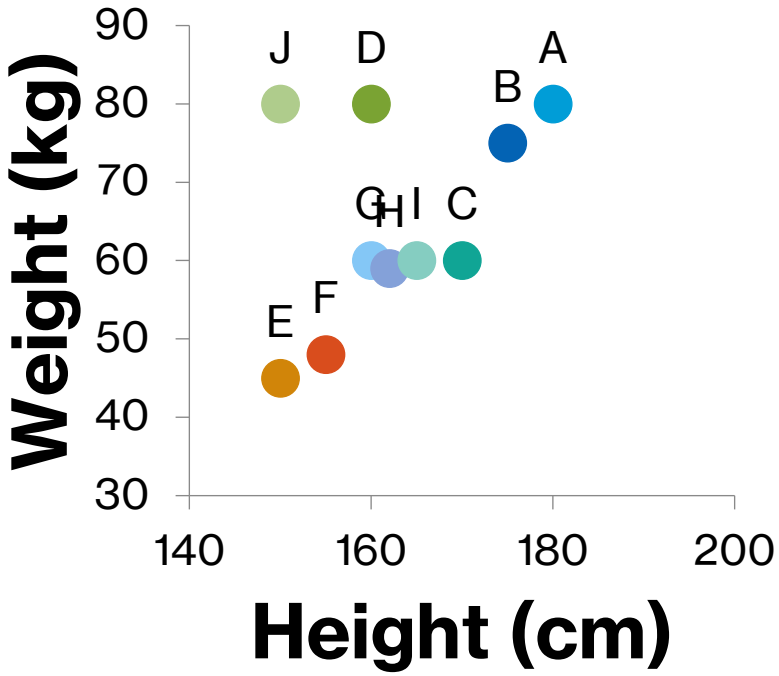
- ✓ Who has similar posture with student B?
- ✓ If we want to divide students into groups having similar posture:
  - ✓ how many groups we have?
  - ✓ How do we allocate students into that group?

Bivariate – among variables

- ✓ How close correlation between height and weight?

Basic Idea

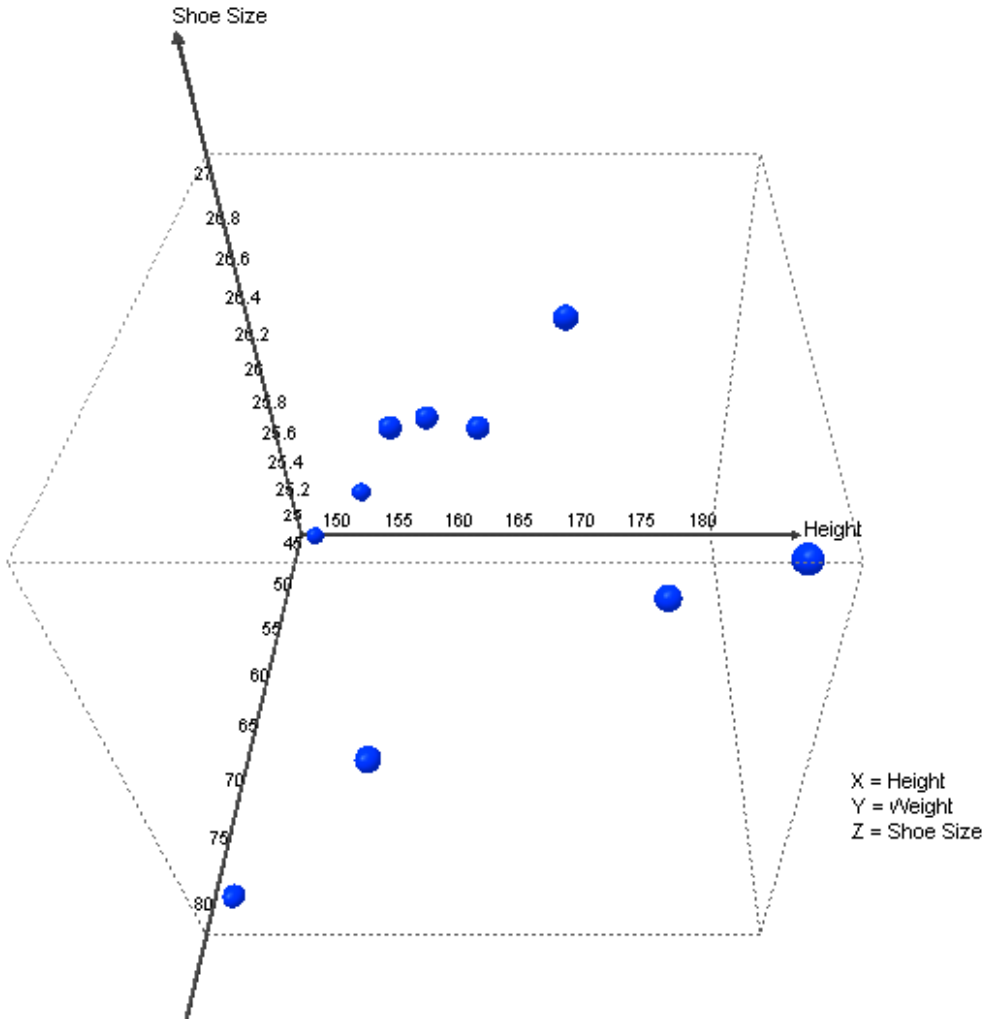
Student	Height (cm)	Weight (kg)
A	180	80
B	175	75
C	170	60
D	160	80
E	150	45
F	155	48
G	160	60
H	162	59
I	165	60
J	150	80



- ✓ Visual representation is more informative
- ✓ Variable act as dimension
- ✓ Similarity among objects is easier to detect
- ✓ Relative position of object to variable is easier to obtained

A little more complex data set

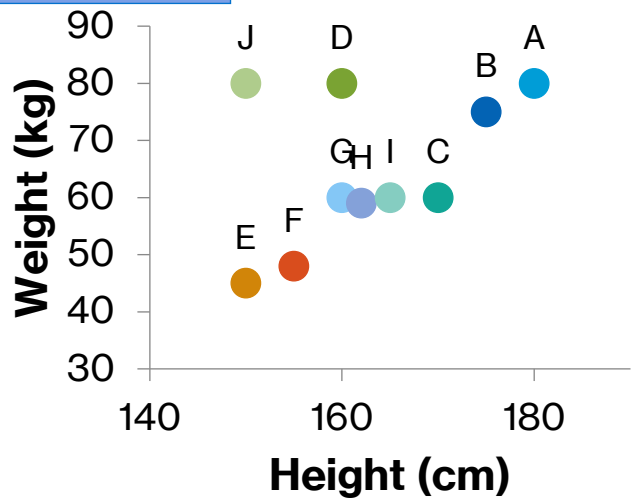
Student	Height (cm)	Weight (kg)	Shoe Size (cm)
A	180	80	27
B	175	75	26.5
C	170	60	27
D	160	80	26
E	150	45	25
F	155	48	25.5
G	160	60	26.5
H	162	59	26.5
I	165	60	26.5
J	150	80	25



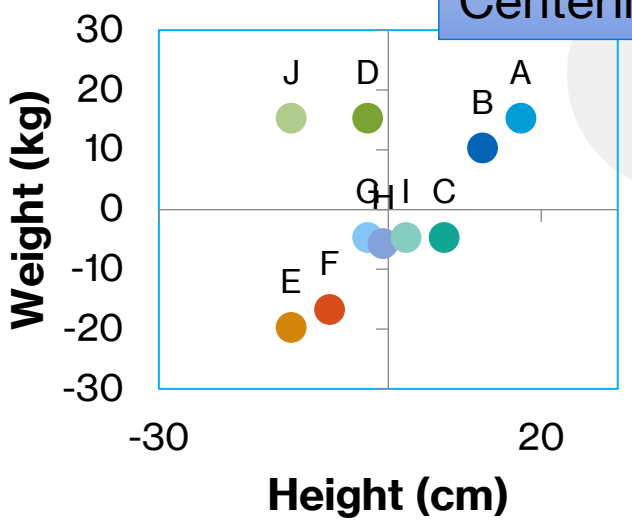
- ✓ Adding one more variable means adding one more dimension in the plot
- ✓ Difficult to observe the plot
- ✓ What should we do?

Rotating the coordinates (1)

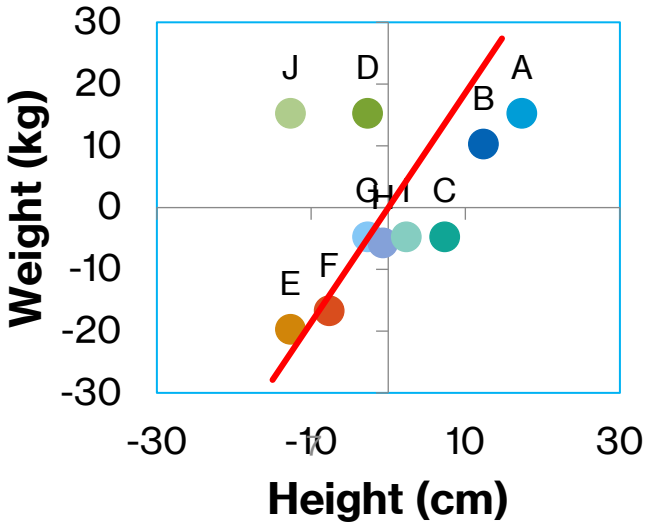
1 Original plot



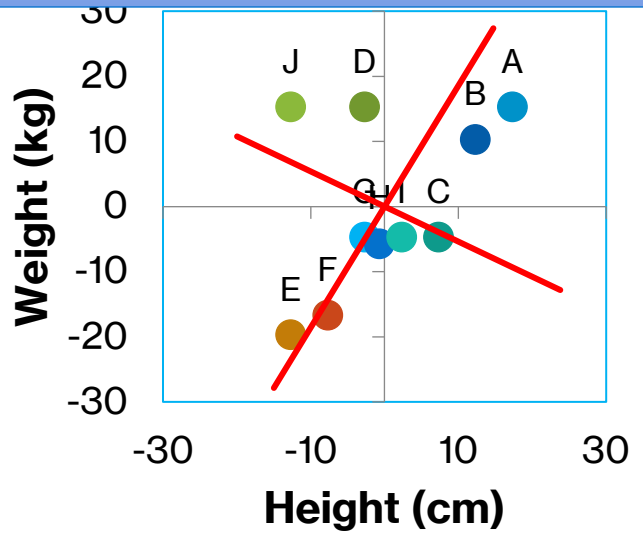
2 Centering the plot



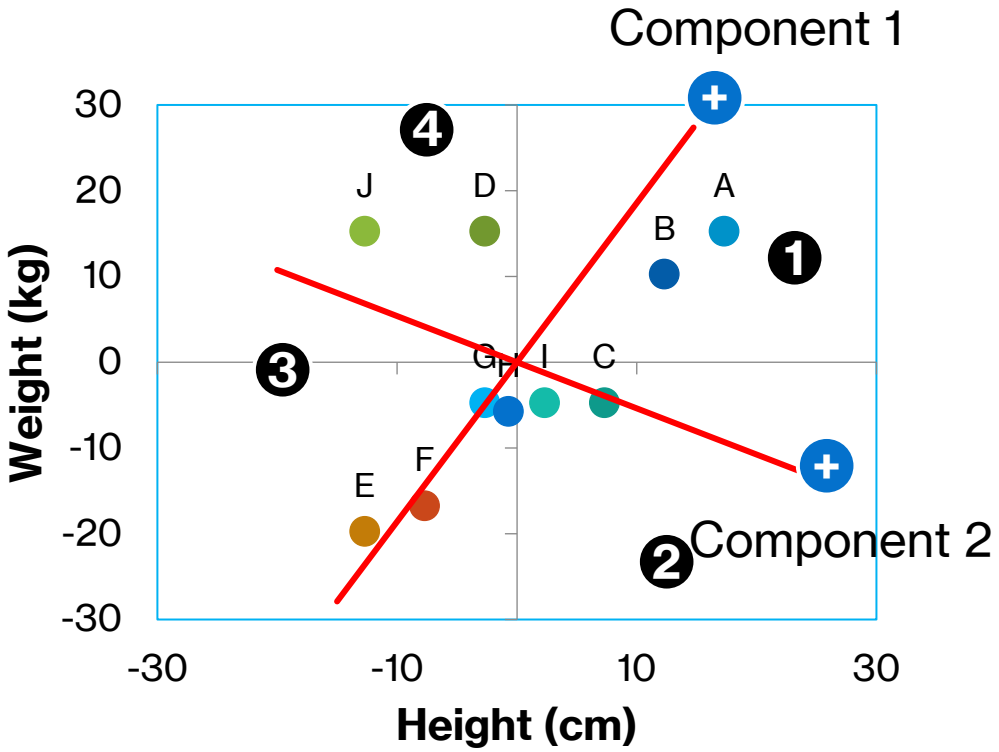
3 1st axis, maximizing the variance

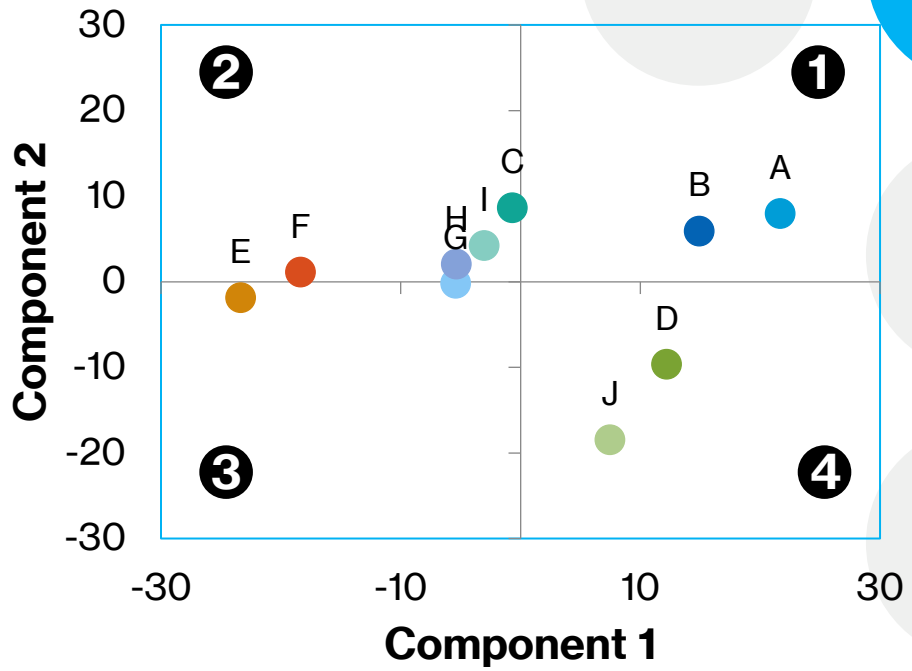


4 2nd axis, maximizing the residual variance, perpendicular to previous axis



Rotating the coordinates (2)





Covariance matrix

	Height	Weight
Height	100.68	56.23
Weight	56.23	174.9

Eigen analysis

		Comp 1	Comp 2	
Eigen value		205.16	70.41	Variance of the comp.
Eigen vector	Height	0.474	0.881	Rotation matrix
	Weight	0.881	-0.474	

Rotating the coordinates (3)

Student	Height (cm)	Weight (kg)
A	180	80
B	175	75
C	170	60
D	160	80
E	150	45
F	155	48
G	160	60
H	162	59
I	165	60
J	150	80

1.Centering
2.Rotating

Student	Comp 1	Comp 2
A	21.67	7.98
B	14.90	5.95
C	-0.68	8.66
D	12.19	-9.63
E	-23.37	-1.85
F	-18.35	1.13
G	-5.42	-0.15
H	-5.35	2.08
I	-3.05	4.25
J	7.45	-18.43

Scores of the components

Rotation matrix

	Comp 1	Comp 2
Height	0.474	0.881
Weight	0.881	-0.474

Comp 1 = 0.474*height + 0.881*weight
Comp 2 = 0.881*height – 0.474*weight

Principal Component Analysis

Student	Height (cm)	Weight (kg)	Shoe Size (cm)
A	180	80	27
B	175	75	26.5
C	170	60	27
D	160	80	26
E	150	45	25
F	155	48	25.5
G	160	60	26.5
H	162	59	26.5
I	165	60	26.5
J	150	80	25

Singular values	Eigen values	Cumulative % of Eigenvalues
42.98	1847.66	74.35
25.23	636.56	99.96
1.00	1.00	100.00

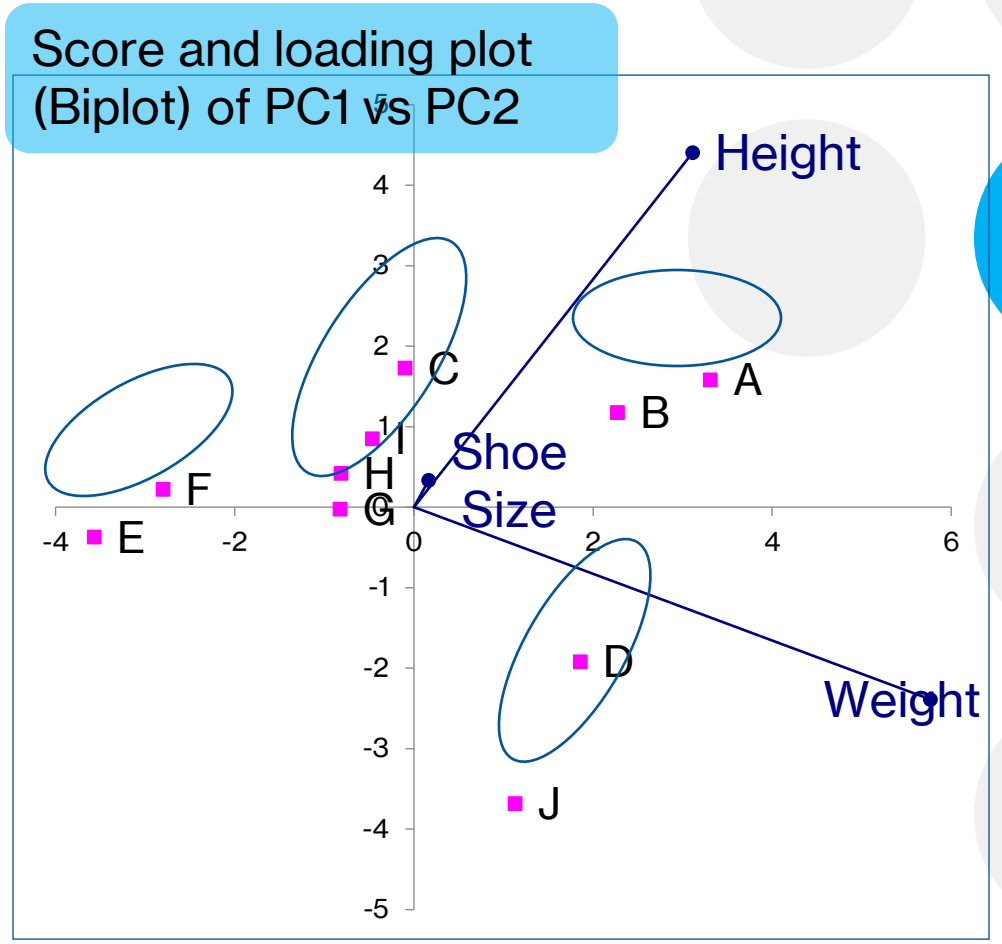
Variables	Eigen vectors		
	PC1	PC2	PC3
Height	0.475	**0.877**	0.071
Weight	**0.880**	-0.475	-0.010
Shoe Size	0.025	0.067	**-0.997**

Students	Score component		
	PC1	PC2	PC3
A	0.505	0.316	0.232
B	0.347	0.235	0.424
C	-0.015	0.345	-0.286
D	0.283	-0.382	-0.187
E	-0.544	-0.074	0.434
F	-0.427	0.045	0.261
G	-0.126	-0.004	-0.495
H	-0.124	0.084	-0.344
I	-0.071	0.169	-0.141
J	0.172	-0.733	0.102

Biplot

Student	Height (cm)	Weight (kg)	Shoe Size (cm)
A	180	80	27
B	175	75	26.5
C	170	60	27
D	160	80	26
E	150	45	25
F	155	48	25.5
G	160	60	26.5
H	162	59	26.5
I	165	60	26.5
J	150	80	25

Variable	s	Correlation		
		Height	Weight	Shoe Size
Height	10.03	1		
Weight	13.22	0.424	1	
Shoe Size	0.75	0.955	0.359	1



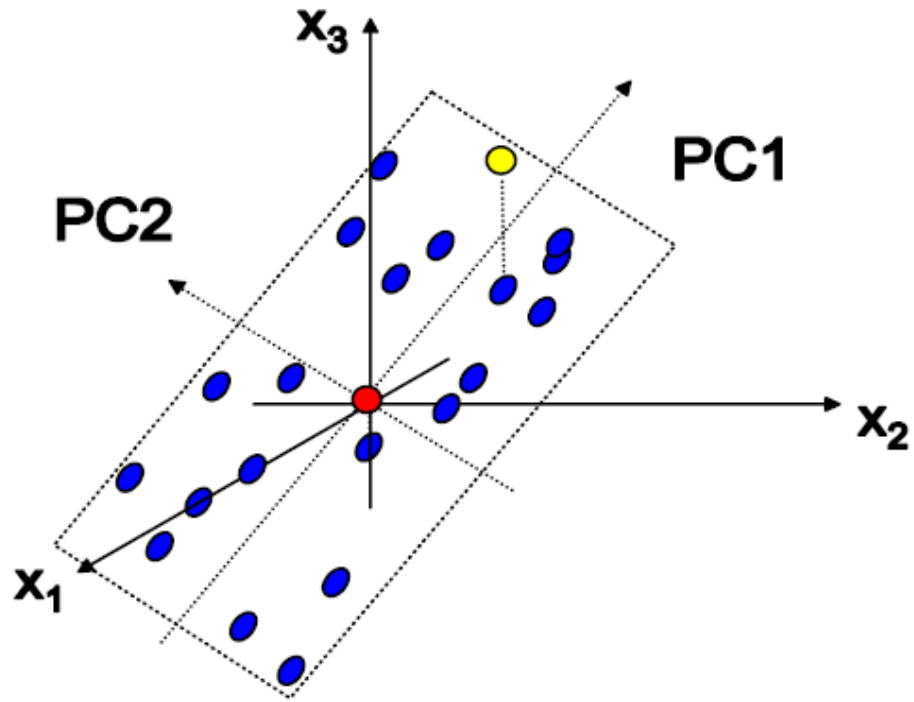
- Circles showing cluster among students
- Magnitude of variable line represents its variance
- Cosines value of angle between two variables shows their correlation
- Relative position of objects to variables showing value of those objects on the variables

Analisis Komponen Utama

What is a Projection?

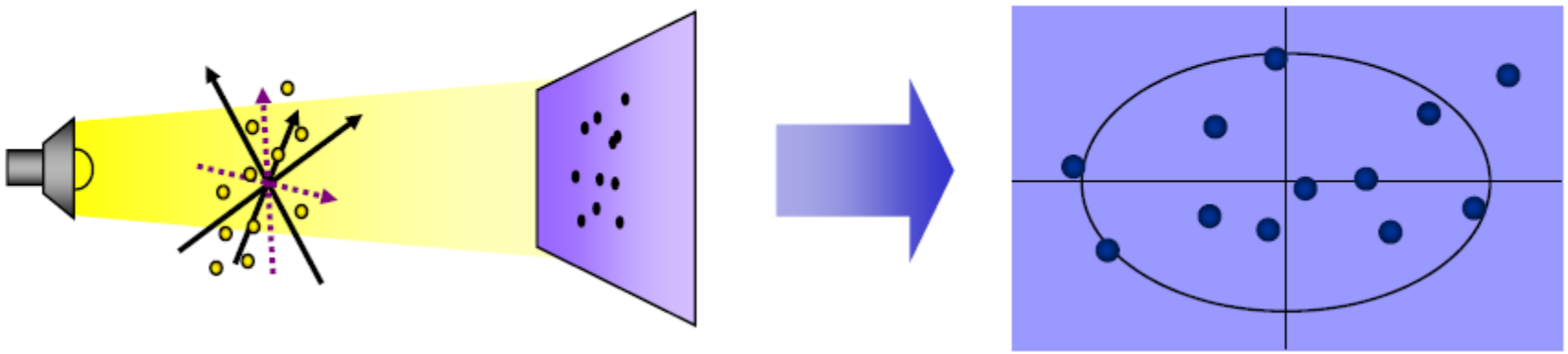
➔ Reduction of dimensionality, model in latent variables!

- Algebraically
  - – Summarizes the information in the observations as a few new (latent) variables
- Geometrically
  - – The swarm of points in a K dimensional space (K = number of variables) is approximated by a (hyper)plane and the points are projected on that plane.

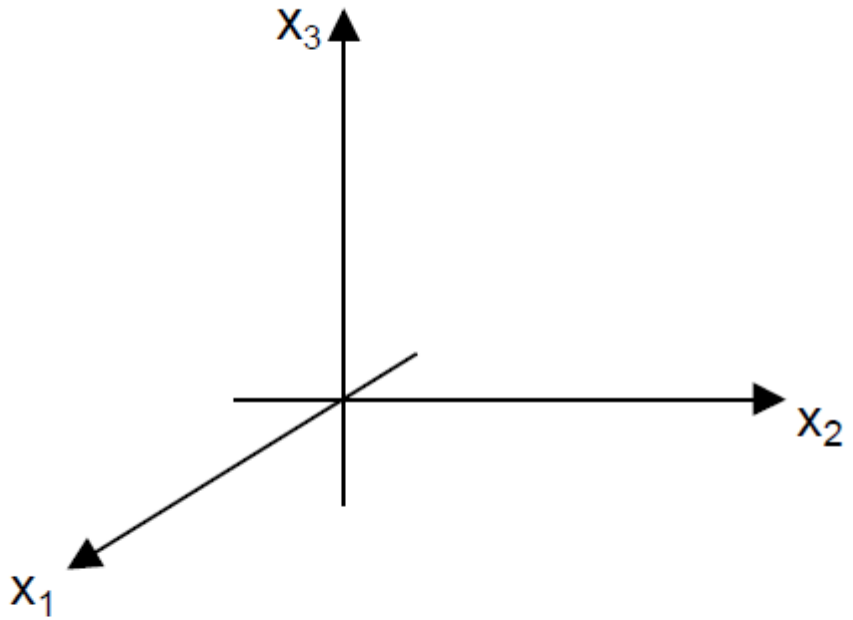


What is a Projection?

- Variables form axes in a multidimensional space
- An observation in multidimensional space is a point
- Project points onto a plane

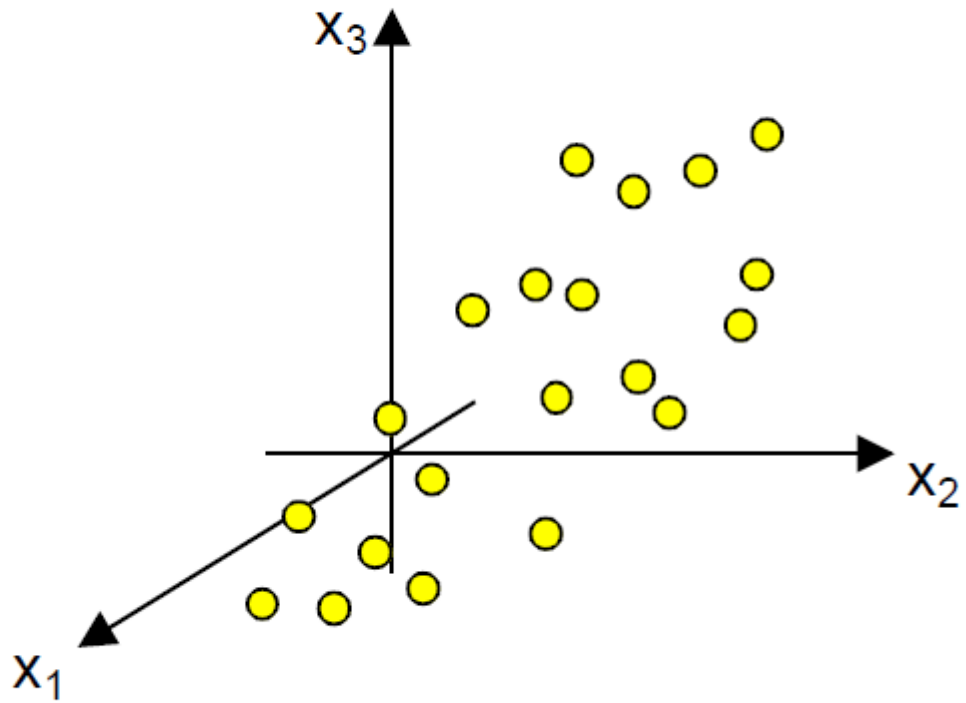


PCA -- Geometric Interpretation, 1



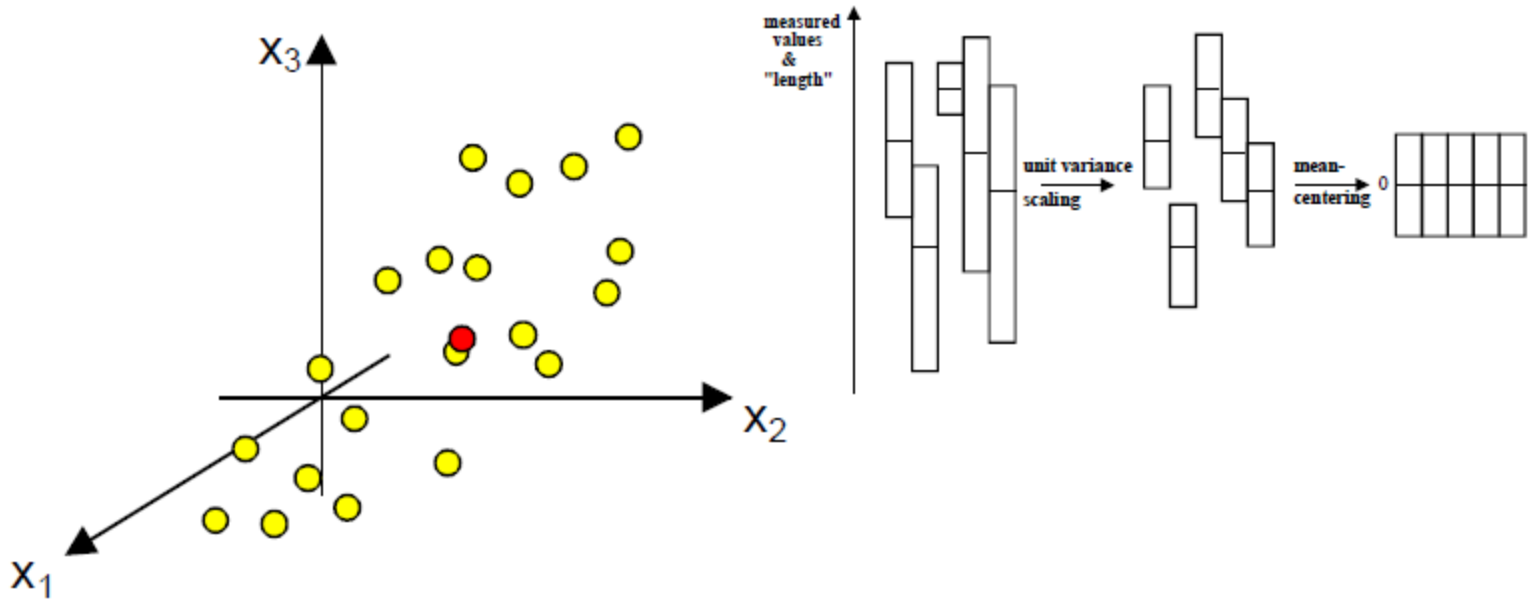
- For the matrix X we construct a space with K dimensions (we see, however, only three of these)
- Each variable has one co-ordinate axis with the length determined by scaling, usually unit variance

PCA -- Geometric Interpretation, 2



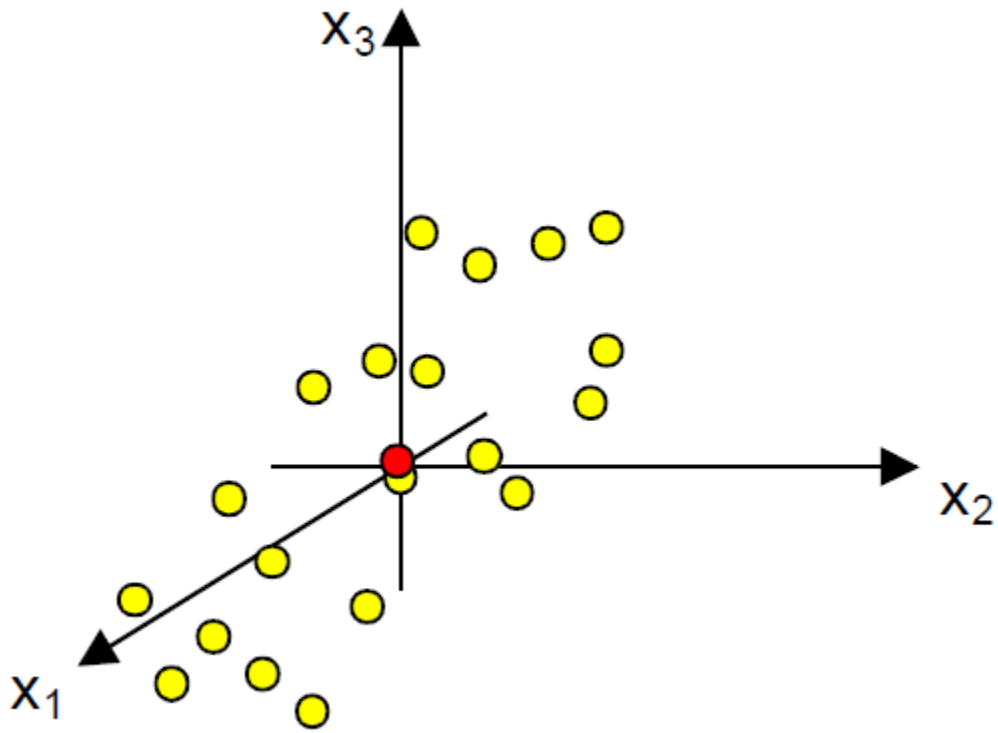
- Each observation is represented by one point in K-space
- Hence, the data matrix X is a swarm of points in this space

PCA -- Geometric Interpretation, 3



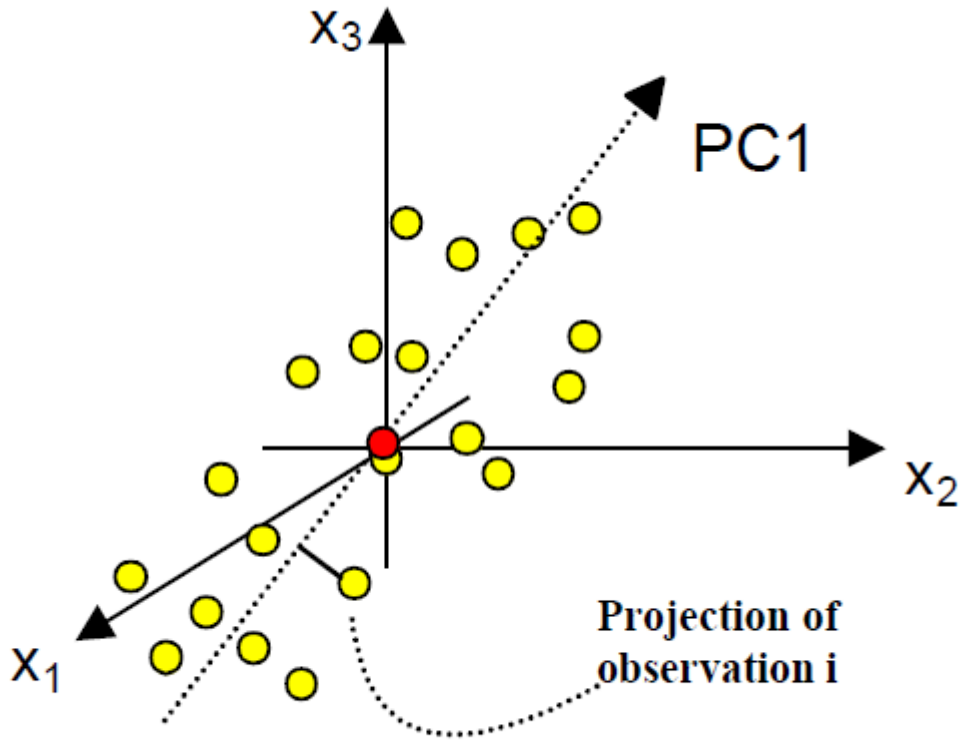
• First we calculate the average of each variable. The vector of averages is a point in K-space. The average is subtracted.

PCA -- Geometric Interpretation, 4



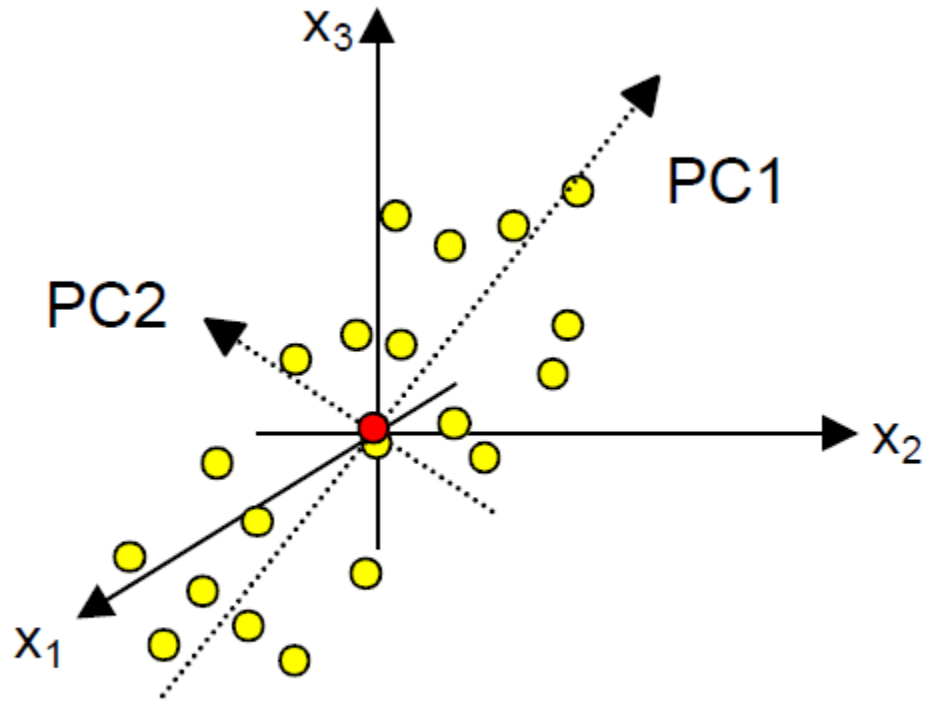
- The mean-centering procedure corresponds to moving the co-ordinate system

PCA -- Geometric Interpretation, 5



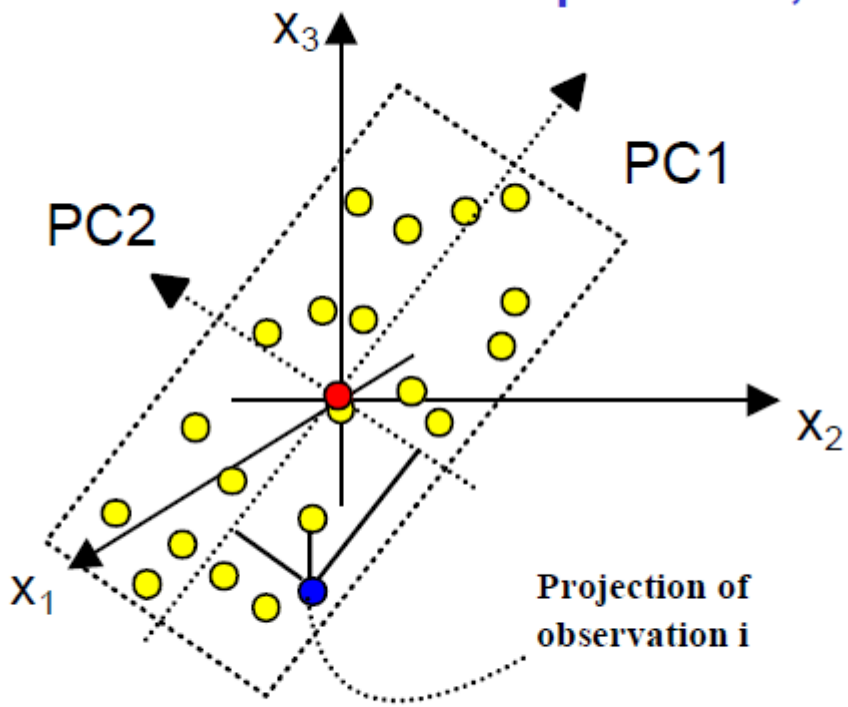
• The first principal component is the line in X-space that best approximates the data (least squares). The line goes through the average point.

PCA -- Geometric Interpretation, 6



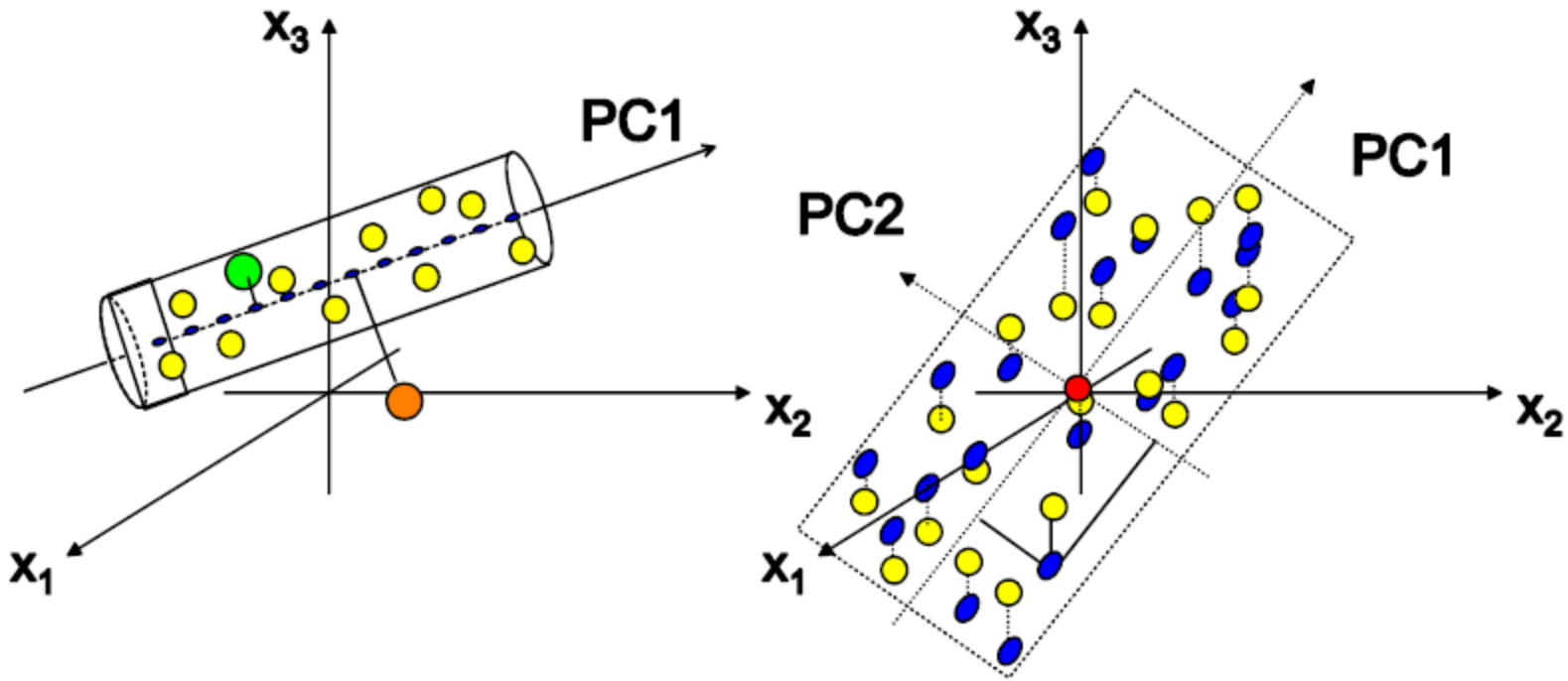
• The second PC is represented by a line in X-space orthogonal to the first PC, also passing through the average point. The second PC improves the approximation of X as much as possible.

PCA -- Geometric Interpretation, 7

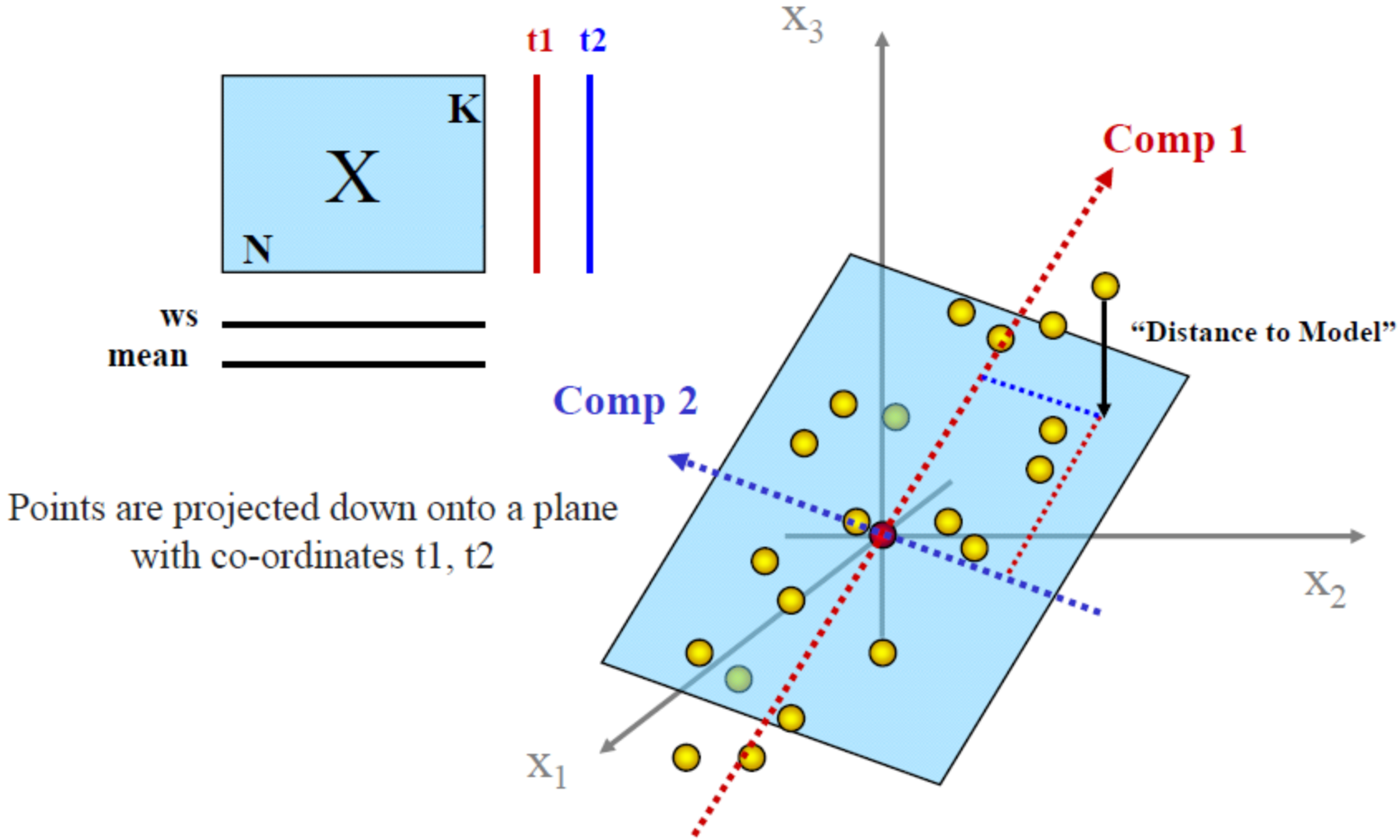


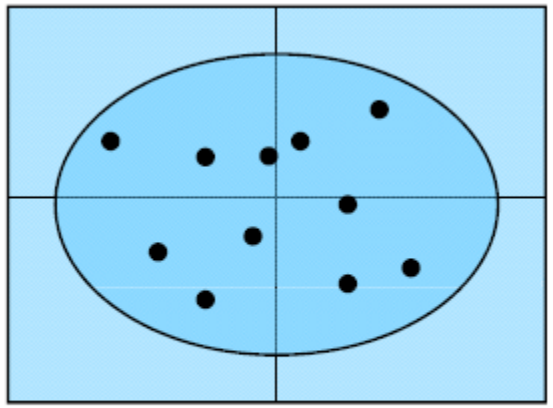
• The two principal components form a plane in the X-space. This plane is a window into the multidimensional space, which can be visualised graphically.

PCA -- Geometric Interpretation, 8

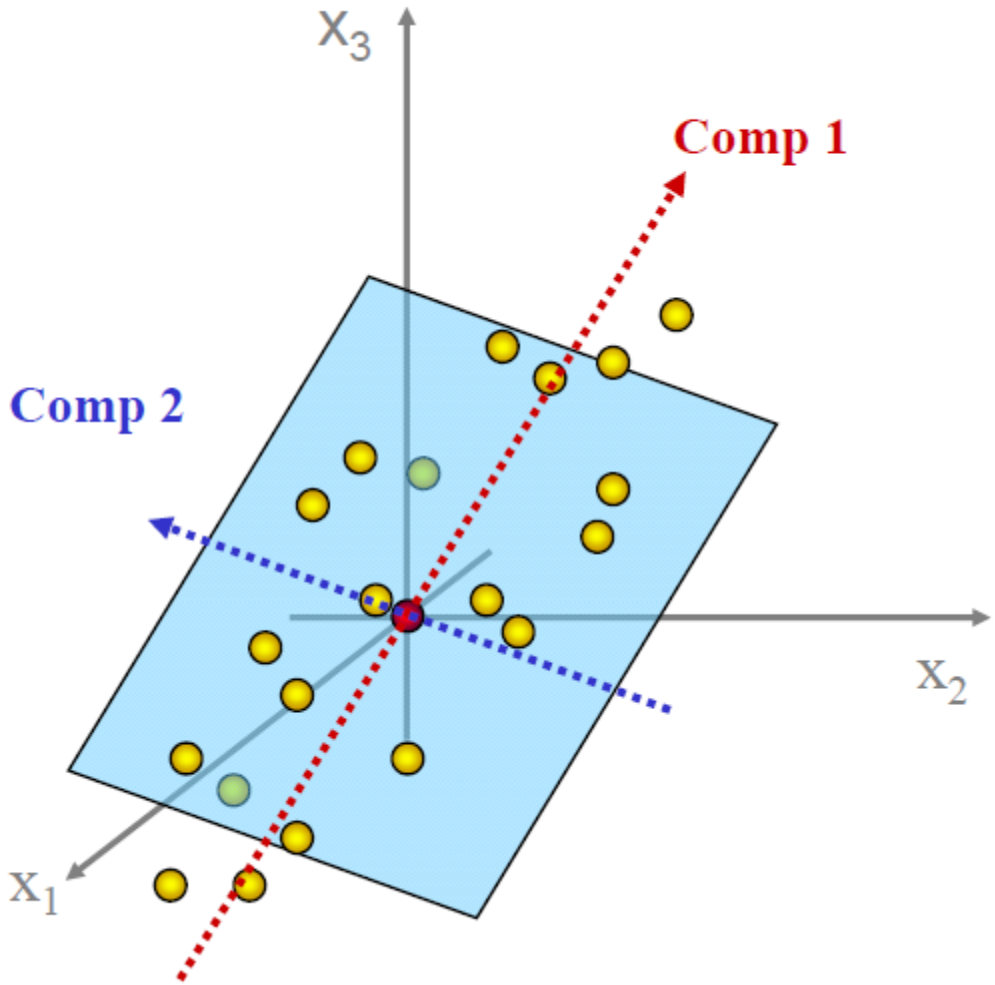


- Yellow points are the observed values. Blue points are their approximations. Projected locations on the model (line, plane, or hyperplane) are given by the *scores* (t).

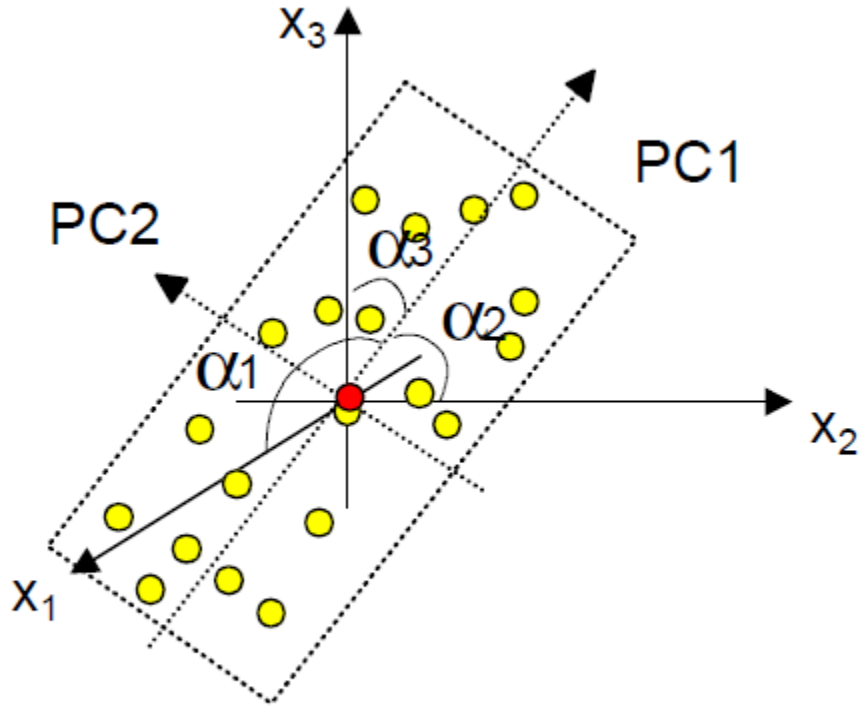

t1
t2
K
X
N
ws
mean
Points are projected down onto a plane
with co-ordinates t1, t2
X3
Comp 1
"Distance to Model"
Comp 2
X2
X1



Plane is then extracted for viewing on computer screen



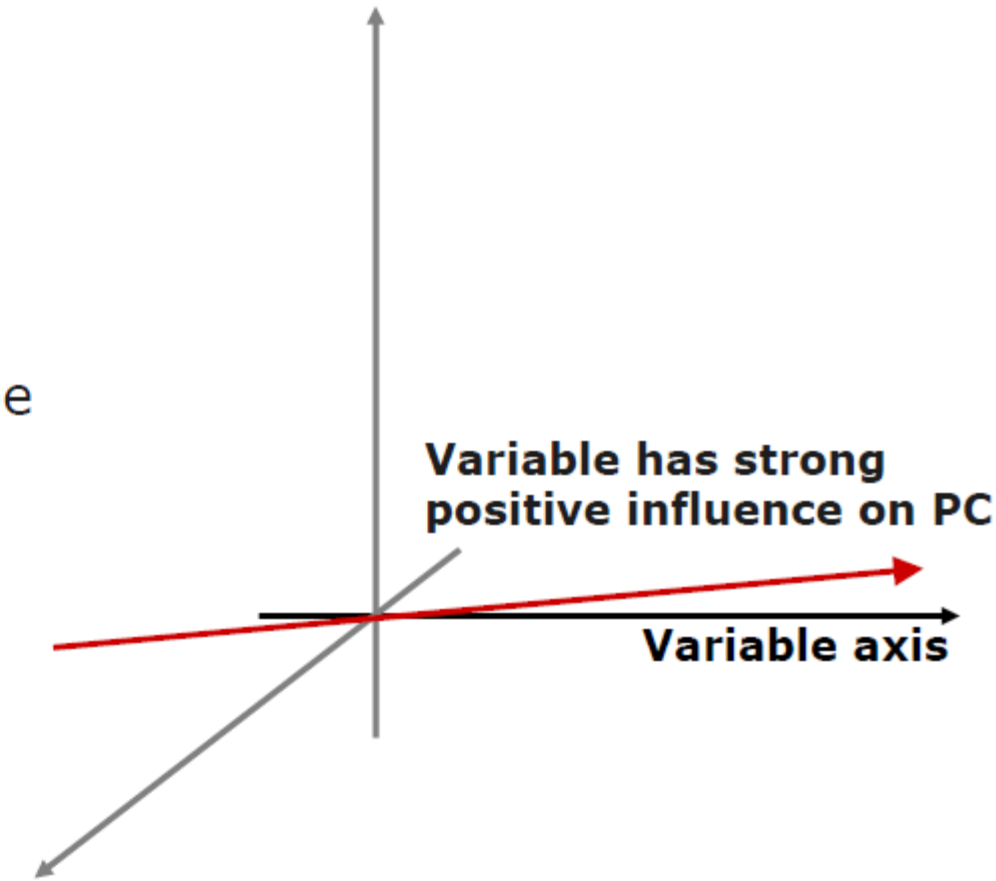
PCA -- Geometric Interpretation, 9



• The direction of, for example, PC1 (p_1) is given by the cosine of the angles α_1, α_2 and α_3. These values indicate how the variables x_1, x_2 and x_3 "load" into PC1. Hence they are called loadings.

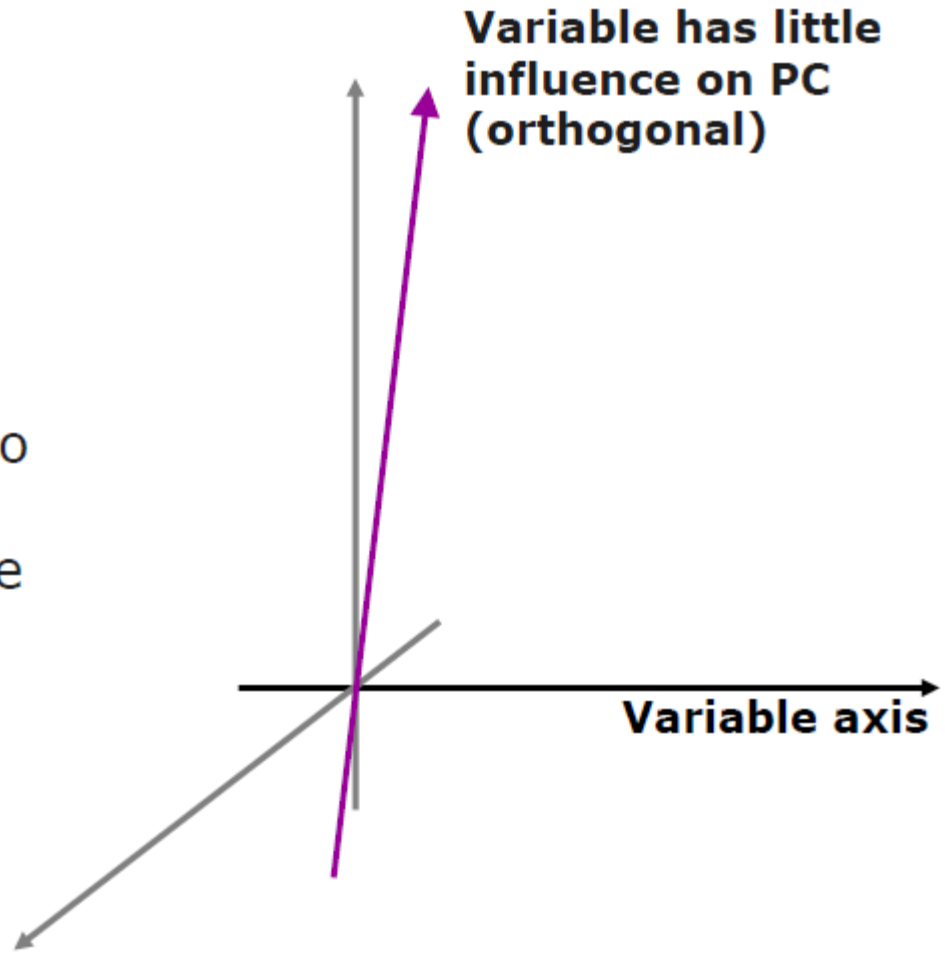
Positive Loading

If component lines up with variable axis the loading will be close to 1 showing strong influence => cos(0) = 1



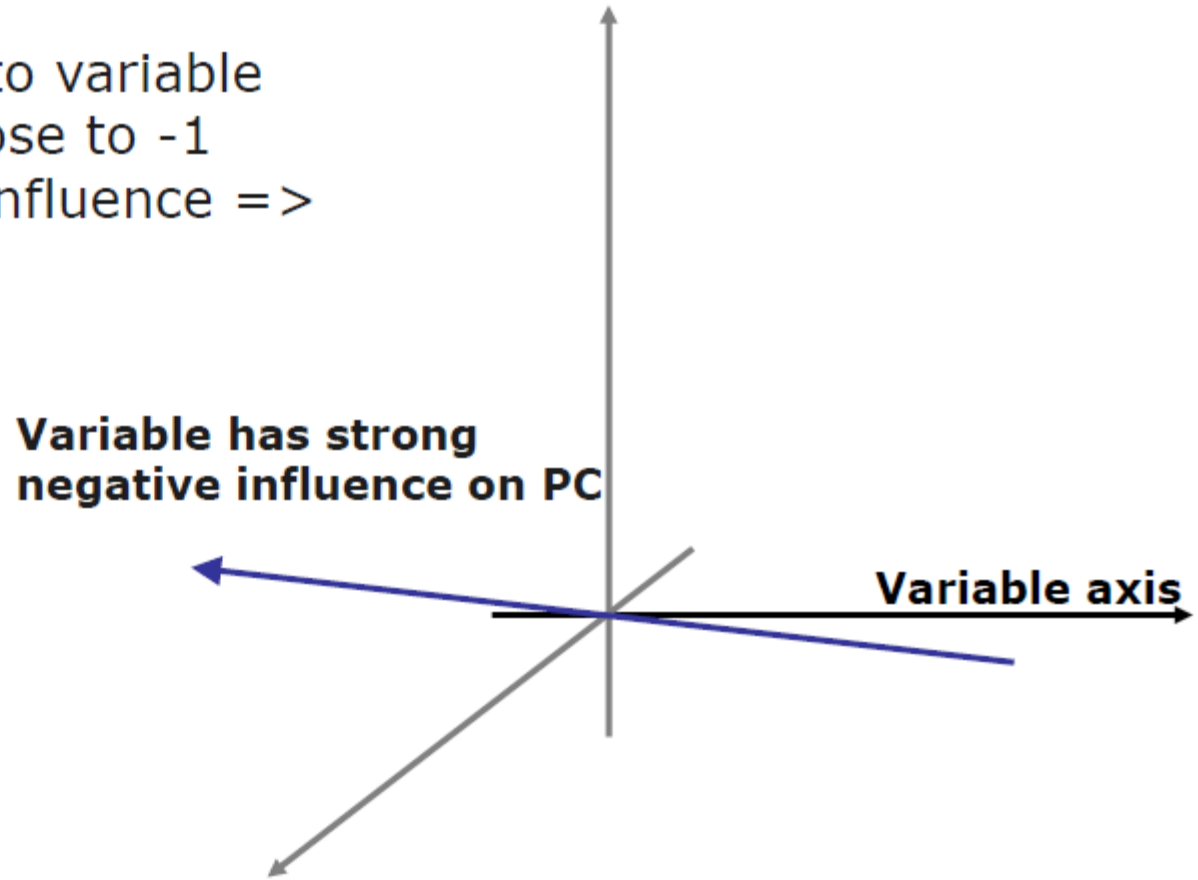
Zero Loading

If component is at right angles to variable axis the loading will be close to 0 showing little influence
=> cos(90) = 0



Negative Loading

If component is opposite to variable axis the loading will be close to -1 showing strong negative influence => cos(180) = -1

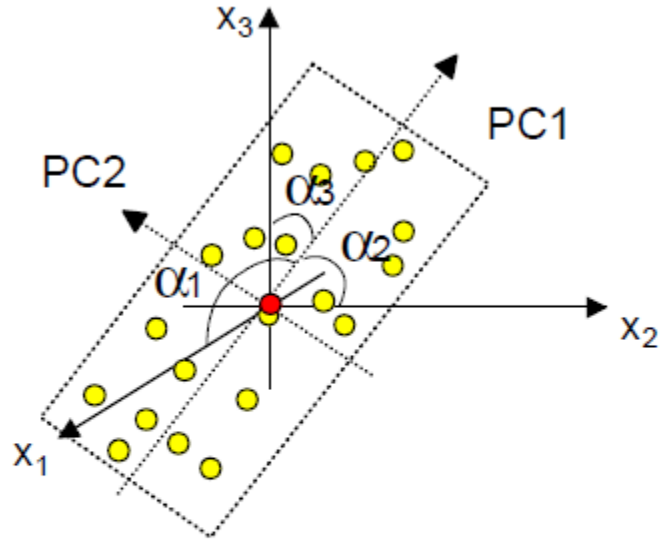


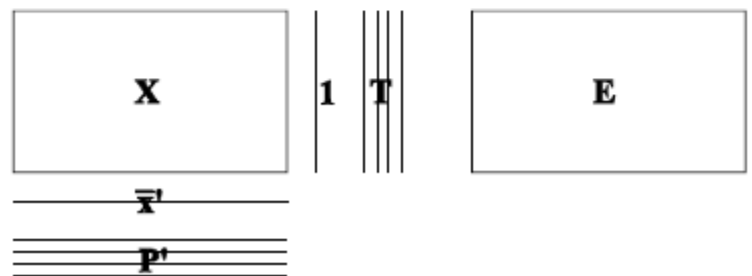
PCA, overview of a data table (data set)

- X is modelled as

$$\mathbf{X} = \mathbf{1} * \bar{\mathbf{x}}' + \mathbf{T} * \mathbf{P}' + \mathbf{E}$$

- Each PC (score vector) is associated with a loading vector
- **Scores, (t)** are co-ordinates in the (hyper)-plane (columns in **T**)
- **Loadings, (p)** define the orientation of the (hyper)-plane (rows in **P'**)
- **DModX,** is the distance between the observations and the model plane (residual row SD)



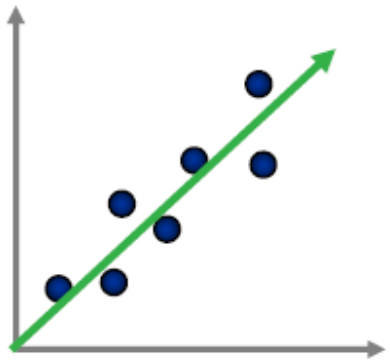


Principal Components Analysis

- **Data visualisation and simplification**
  - Information resides in the ***correlation structure*** of the data
  - Mathematical principle of projection to lower dimensionality

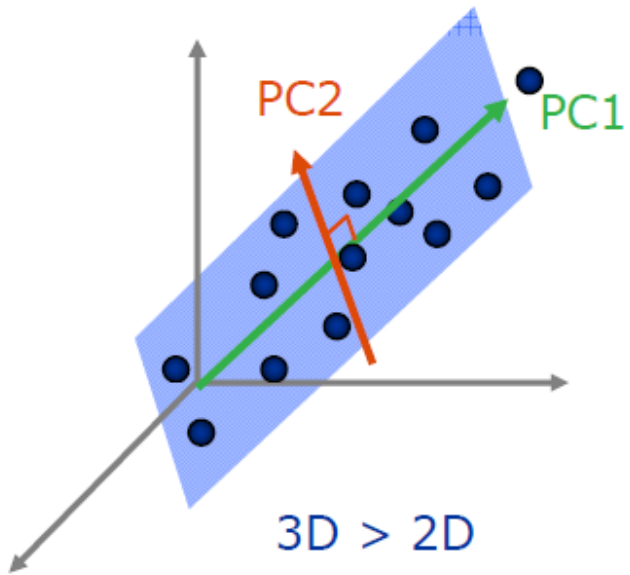
2 Variables

V1	V2
1	1.3
2	2.3
3	2.7
4	3.9
…	…



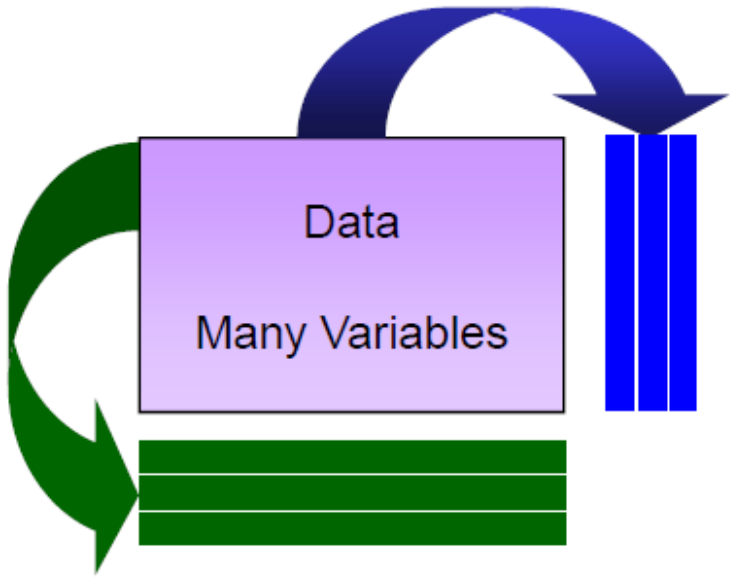
Many Variables

V1	V2	V3	Vn
1	1.3	0.4	
2	2.3	1.2	
3	2.7	2.1	
4	3.9	4.6	
…	…	…	…



PCA Simplifies Data

- PCA breaks down a large table of data into two smaller ones
- Plots of scores and loadings turn data into pictures
- Correlations among **observations** and **variables** are easily seen



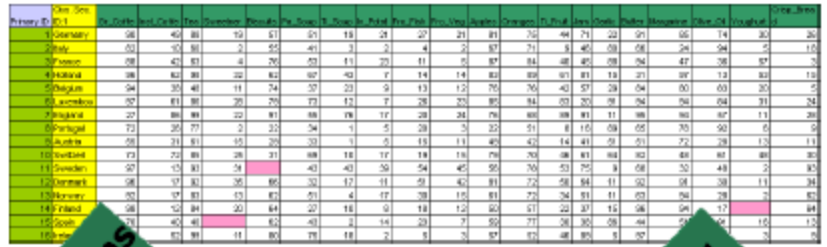
SCORES

- Summarise the observations
- Separates signal from noise
- Observe patterns, trends, clusters

LOADINGS

- Summarise the variables
- Explain the position of observations in scores plot

PCA Converts Tables to Pictures

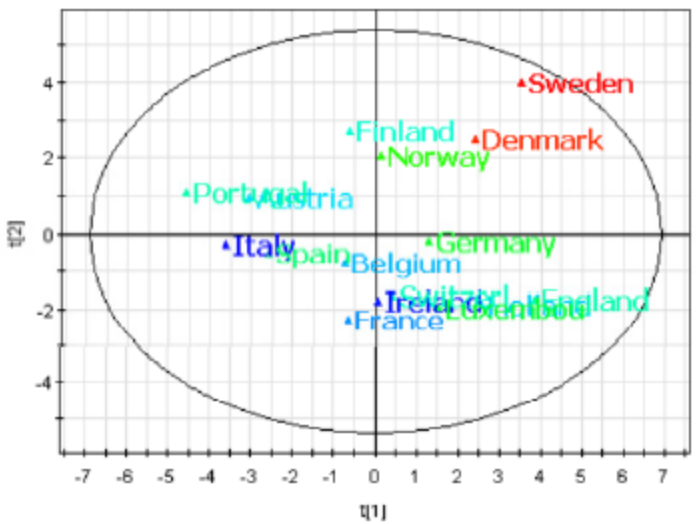


Observations

Variables

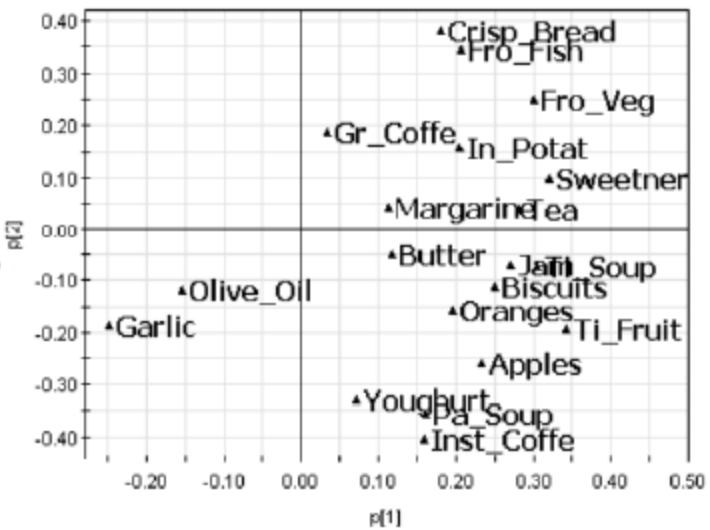
PCA converts table into two interpretable plots:

Foods.M1 t[1]/t[2]
Colored according to value in variable Fro_Fish



Interpretation

1 (PCA-X), Foods PCA
p[1]/p[2]



Scores plot relates to **observations**

Loadings plot relates to **variables**

PCA Example

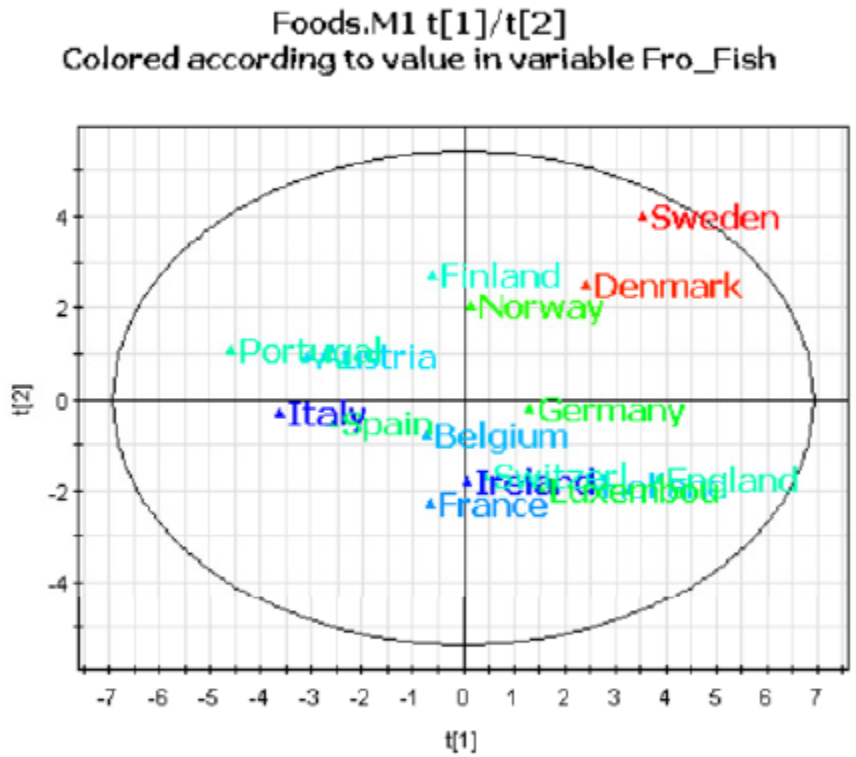
Problem: To investigate patterns of food consumption in Western Europe, the percentage usage of 20 common food products was obtained for 16 countries

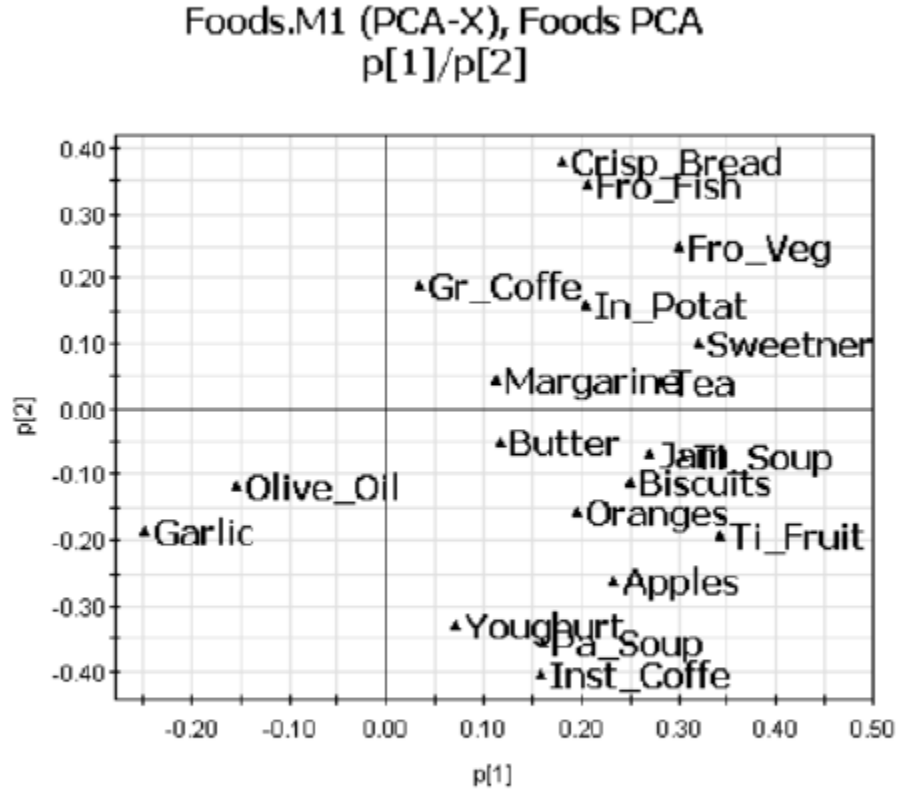
Perform a multivariate analysis (PCA) to overview data

Food consumption patterns for 16 European countries (part of the data).

COUNTRY	Grain coffee	Instant coffee	Tea	Sweet-ner	Bis-cuits	Pa soup	Ti soup	In potat	Fro fish	Fro veg	Fresh apple	Fresh orange	Ti fruit	Jam	Garlic	Butter	Marg-arine
Germany	90	49	88	19	57	51	19	21	27	21	81	75	44	71	22	91	85
Italy	82	10	60	2	55	41	3	2	4	2	67	71	9	46	80	66	24
France	88	42	63	4	76	53	11	23	11	5	87	84	40	45	88	94	47
Holland	96	62	98	32	62	67	43	7	14	14	83	89	61	81	15	31	97
Belgium	94	38	48	11	74	37	23	9	13	12	76	76	42	57	29	84	80
Luxembou	97	61	86	28	79	73	12	7	26	23	85	94	83	20	91	94	94
England	27	86	99	22	91	55	76	17	20	24	76	68	89	91	11	95	94
Portugal	72	26	77	2	22	34	1	5	20	3	22	51	8	16	89	65	78
Austria	55	31	61	15	29	33	1	5	15	11	49	42	14	41	51	51	72
Switzerl	73	72	85	25	31	69	10	17	19	15	79	70	46	61	64	82	48
Sweden	97	13	93	31		43	43	39	54	45	56	78	53	75	9	68	32
Denmark	96	17	92	35	66	32	17	11	51	42	81	72	50	64	11	92	91
Norway	92	17	83	13	62	51	4	17	30	15	61	72	34	51	11	63	94
Finland	98	12	84	20	64	27	10	8	18	12	50	57	22	37	15	96	94
Spain	70	40	40		62	43	2	14	23	7	59	77	30	38	86	44	51
Ireland	30	52	99	11	80	75	18	2	5	3	57	52	46	89	5	97	25

General PCA Example - Foods



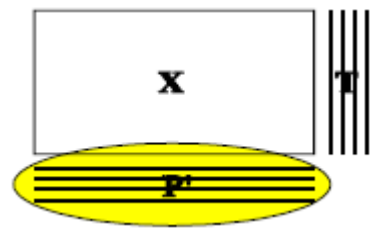


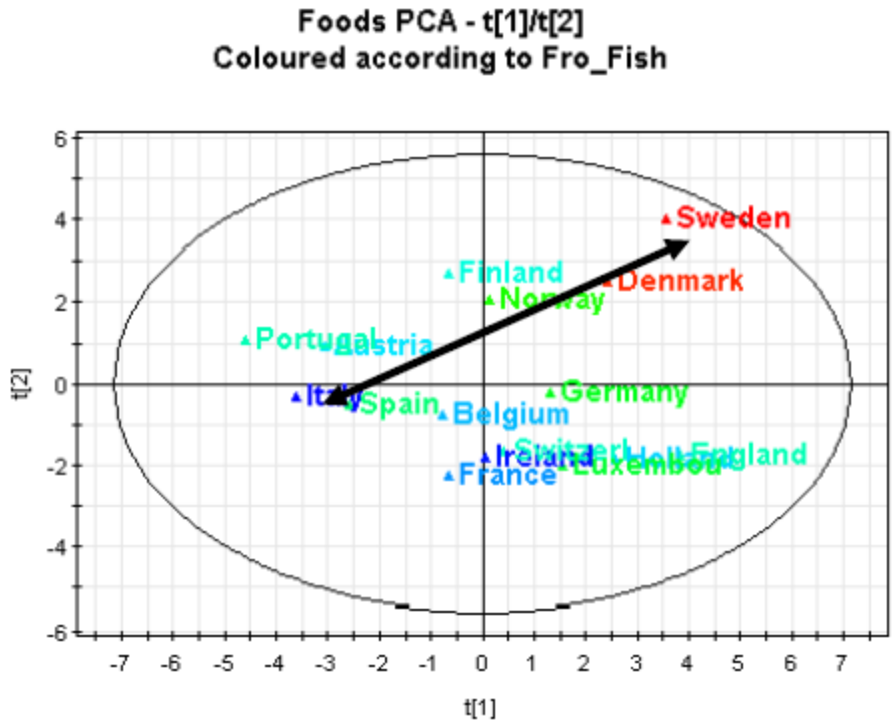
Observations

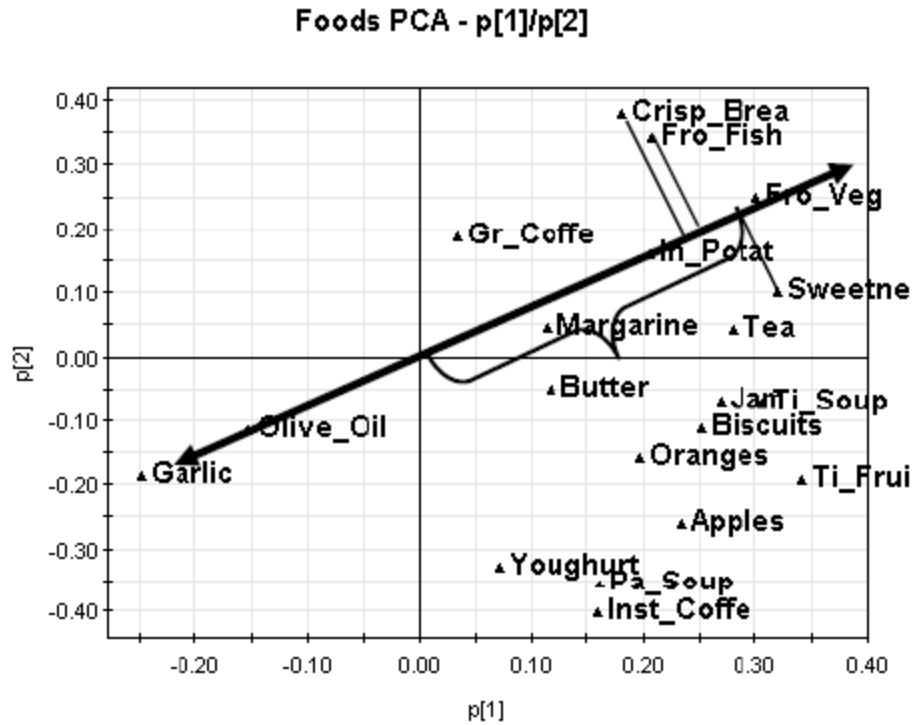
Scores plot

Variables

Loadings plot



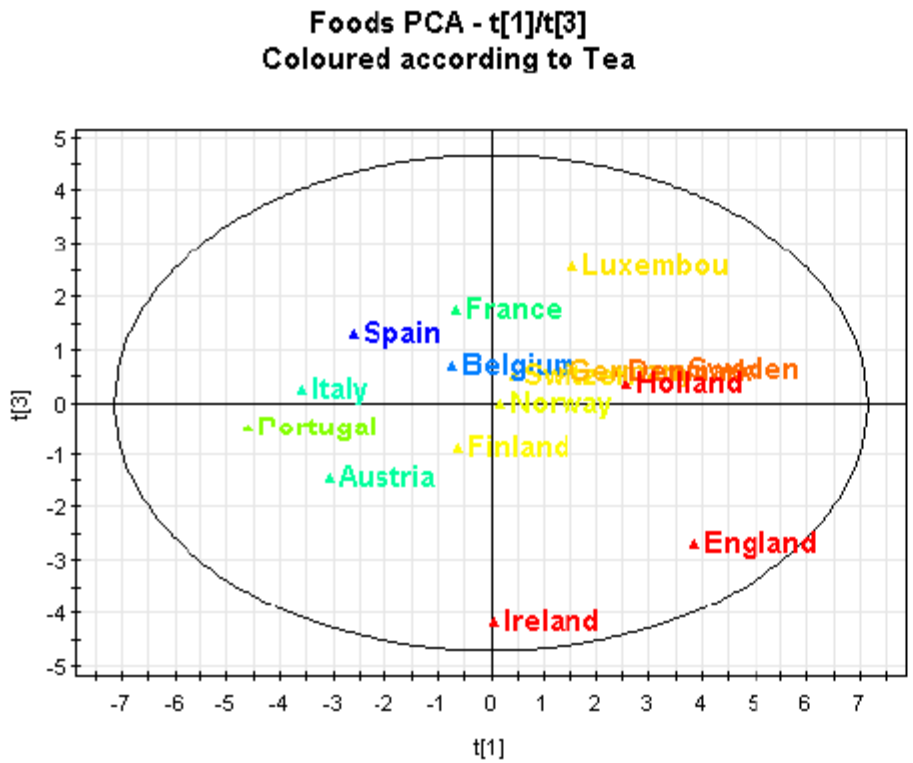


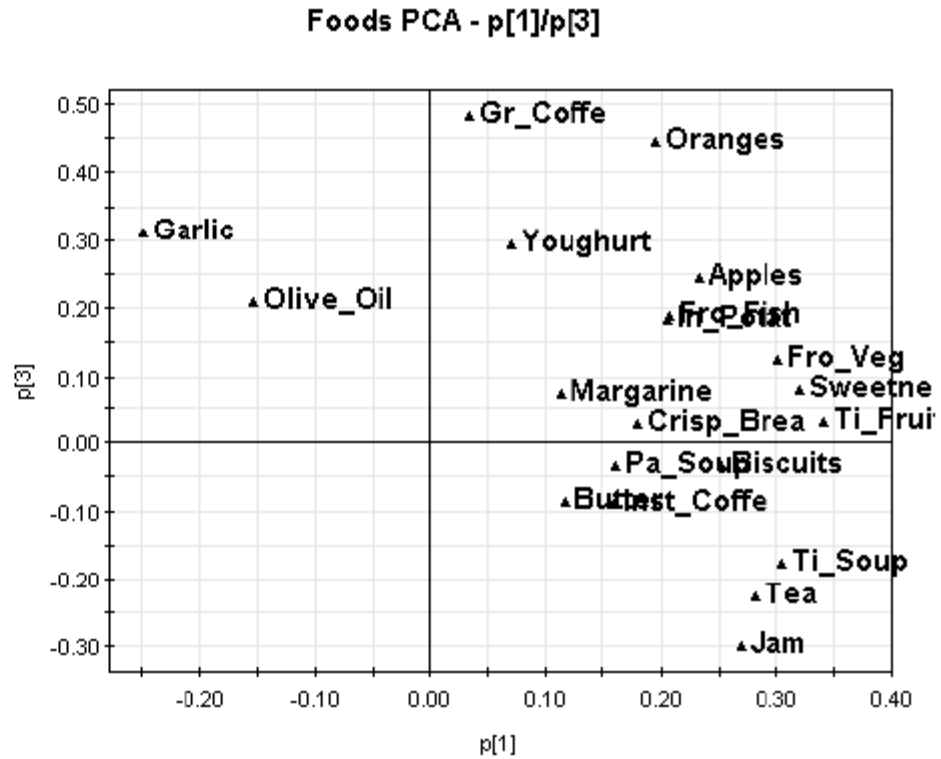


Observations

Variables

Why are Italy and Spain different from Sweden and Denmark?





Observations

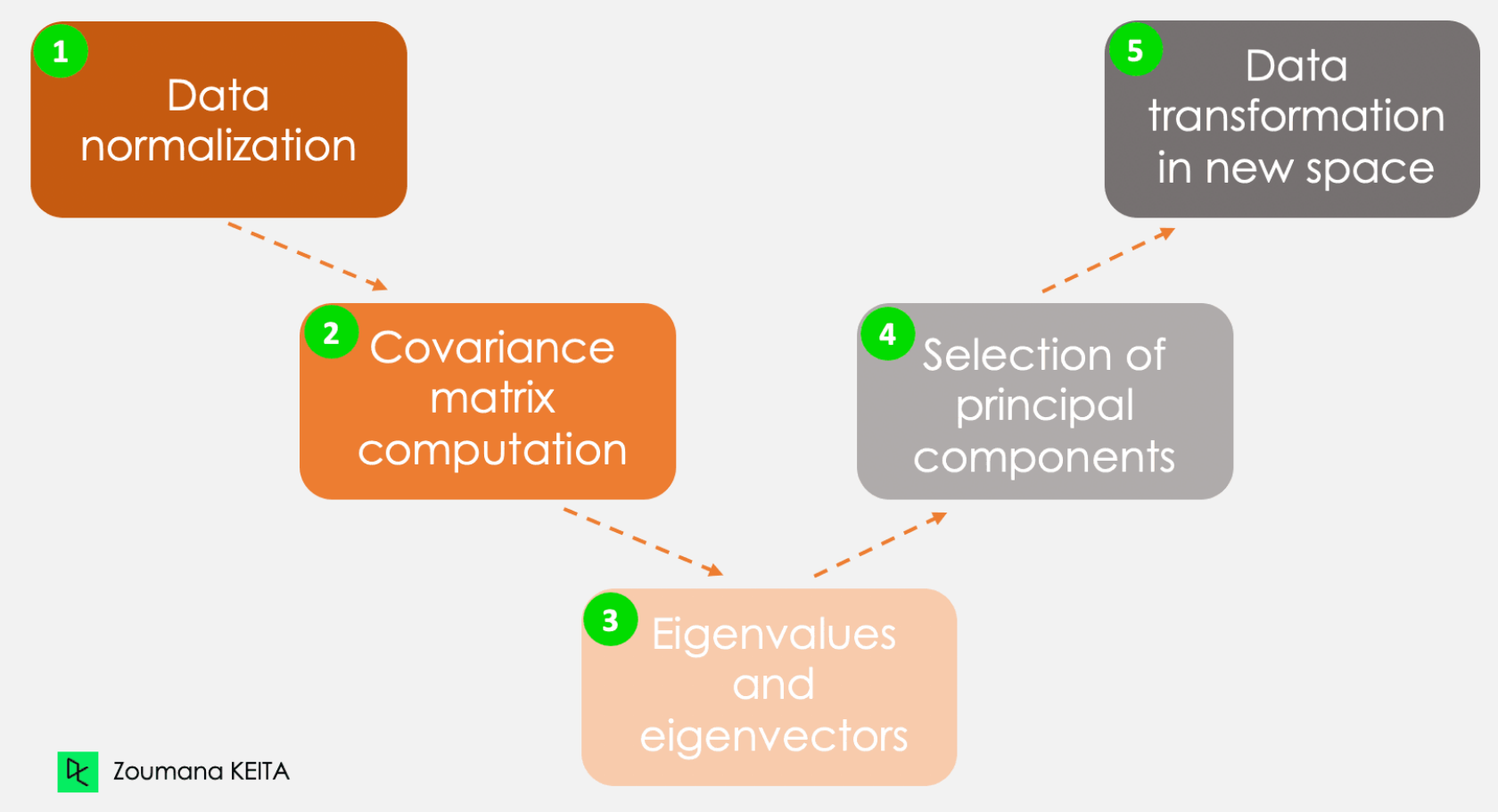
Variables

In the third component Ireland and England are different from the other countries

Applications of Principal Component Analysis

- Finance → stock prices
- Image processing → image recognition
- Healthcare → MRI scan
- Security → biometric system

Summary



https://www.datacamp.com/tutorial/pca-analysis-r

Illustration

Protein Data

- The protein data set is a real-valued multivariate data set describing the average protein consumption by citizens of 25 European countries.
- For each country, there are ten columns. The first eight correspond to the different types of proteins. The last one corresponds to the total value of the average values of proteins.

Source: https://www.datacamp.com/tutorial/pca-analysis-r

Protein Data

```
$ Country                 : chr  "Albania" "Austria" "Belgium" "Bulgaria" ...
$ Red_Meat                : int  10 9 14 8 10 11 8 10 18 10 ...
$ White_Meat              : int  1 14 9 6 11 11 12 5 10 3 ...
$ Eggs                    : int  1 4 4 2 3 4 4 3 3 3 ...
$ Milk                    : int  9 20 18 8 13 25 11 34 20 18 ...
$ Fish                    : int  0 2 5 1 2 10 5 6 6 6 ...
$ Cereals                 : int  42 28 27 57 34 22 25 26 28 42 ...
$ Starchy_Foods           : int  1 4 6 1 5 5 7 5 5 2 ...
$ Pulses_nuts_oilseeds: int  6 1 2 4 1 1 1 1 2 8 ...
$ Fruits_Vegetables       : int  2 4 4 4 4 2 4 1 7 7 ...
```

Protein Data

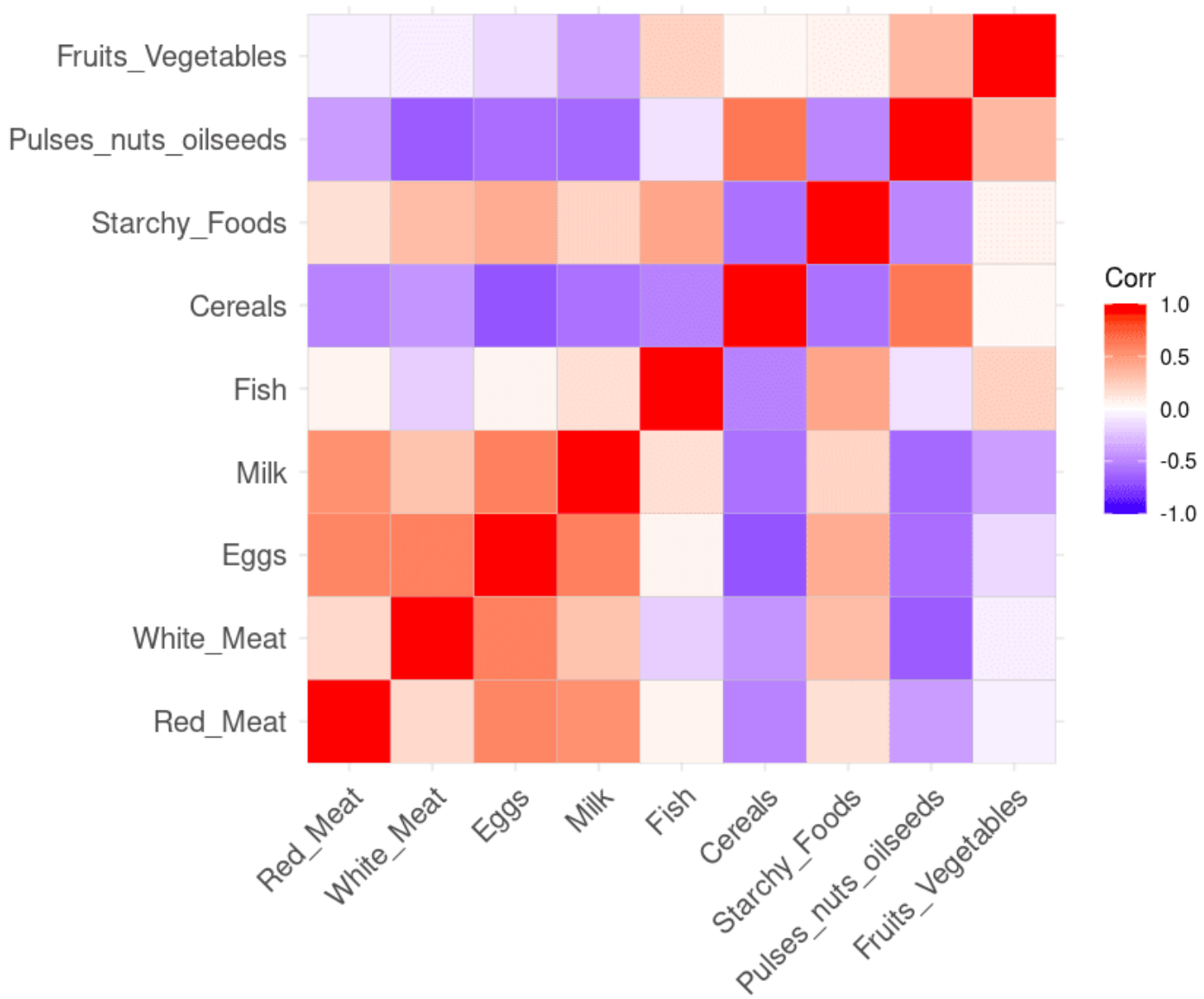
	Red_Meat	White_Meat	Eggs	Milk	Fish	Cereals
1	10	1	1	9	0	42
2	9	14	4	20	2	28
3	14	9	4	18	5	27
4	8	6	2	8	1	57
5	10	11	3	13	2	34
6	11	11	4	25	10	22

Data Normalization

A matrix: 6 × 9 of type dbl

Red_Meat	White_Meat	Eggs	Milk	Fish	Cereals
0.05876425	-1.8498883	-1.86538958	-1.1665829	-1.2333048	0.8791769
-0.23505701	1.6253354	0.82507616	0.3832253	-0.6569941	-0.3923599
1.23404931	0.2887109	0.82507616	0.1014420	0.2074718	-0.4831840
-0.52887828	-0.5132638	-0.96856767	-1.3074746	-0.9451495	2.2415378
0.05876425	0.8233607	-0.07174575	-0.6030163	-0.6569941	0.1525844
0.35258552	0.8233607	0.82507616	1.0876836	1.6482484	-0.9373043

Correlation Matrix



Applying PCA

```
Importance of components:
                          Comp.1     Comp.2     Comp.3      Comp.4      Comp.5
Standard deviation     1.306476 0.5182247 0.3610639 0.27323842 0.14161042
Proportion of Variance 0.768188 0.1208652 0.0586723 0.03360071 0.00902517
Cumulative Proportion  0.768188 0.8890531 0.9477255 0.98132616 0.99035133
                           Comp.6       Comp.7       Comp.8 Comp.9
Standard deviation     0.114308469 0.081225268 0.042129867      0
Proportion of Variance 0.005880602 0.002969253 0.000798813      0
Cumulative Proportion  0.996231934 0.999201187 1.000000000      1
```

Focus here

Applying PCA

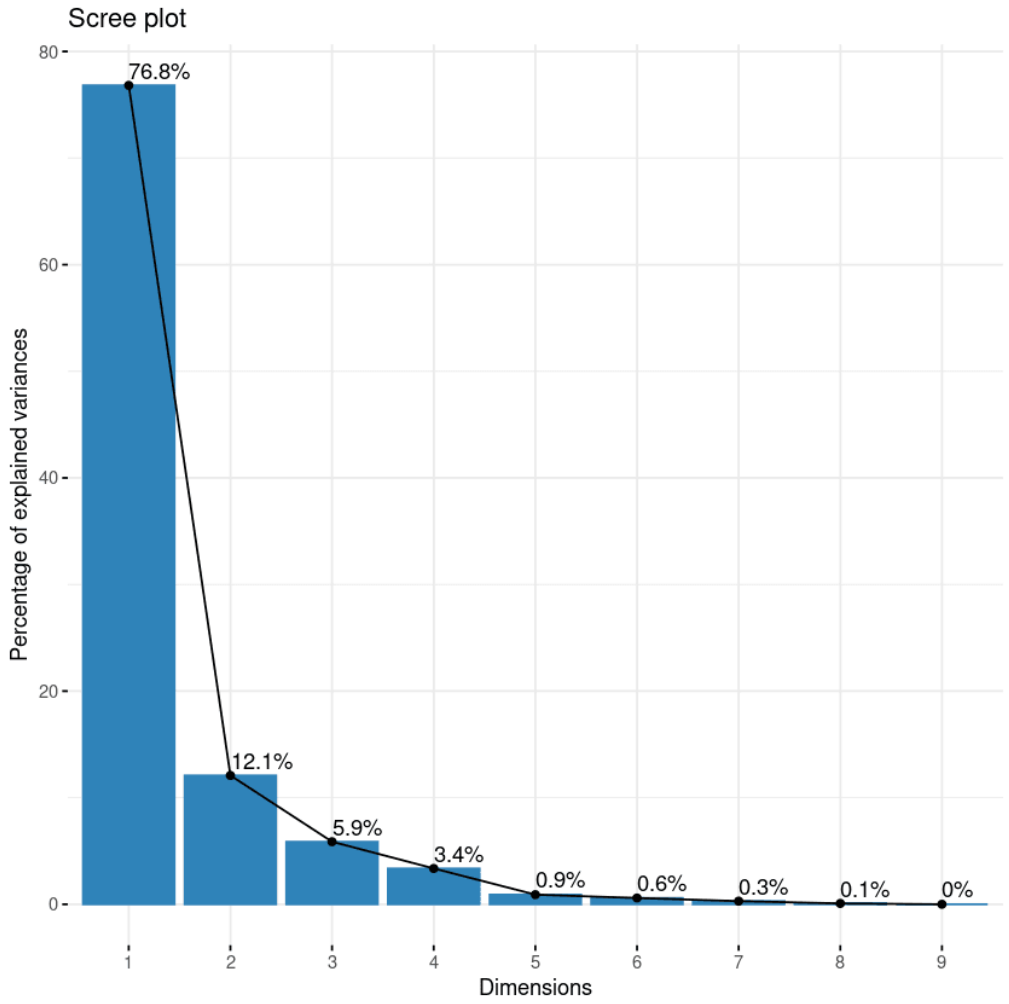
- Each component explains a percentage of the total variance in the data set. In the **Cumulative Proportion** section, the first principal component explains almost 77% of the total variance. This implies that almost two-thirds of the data in the set of 9 variables can be represented by just the first principal component. The second one explains 12.08% of the total variance.
- The cumulative proportion of Comp.1 and Comp.2 explains nearly 89% of the total variance. This means that the first two principal components can accurately represent the data.

Loading Vectors

A matrix: 9 × 2 of type dbl

	Comp.1	Comp.2
Red_Meat	0.2993407	0.10651363
White_Meat	0.3193363	0.22312711
Eggs	0.4134492	0.11960563
Milk	0.3837089	0.15175036
Fish	0.1137789	-0.68417466
Cereals	-0.4246411	0.28573531
Starchy_Foods	0.2807581	-0.40862948
Pulses_nuts_oilseeds	-0.4375650	-0.07772646
Fruits_Vegetables	-0.1633793	-0.42281956

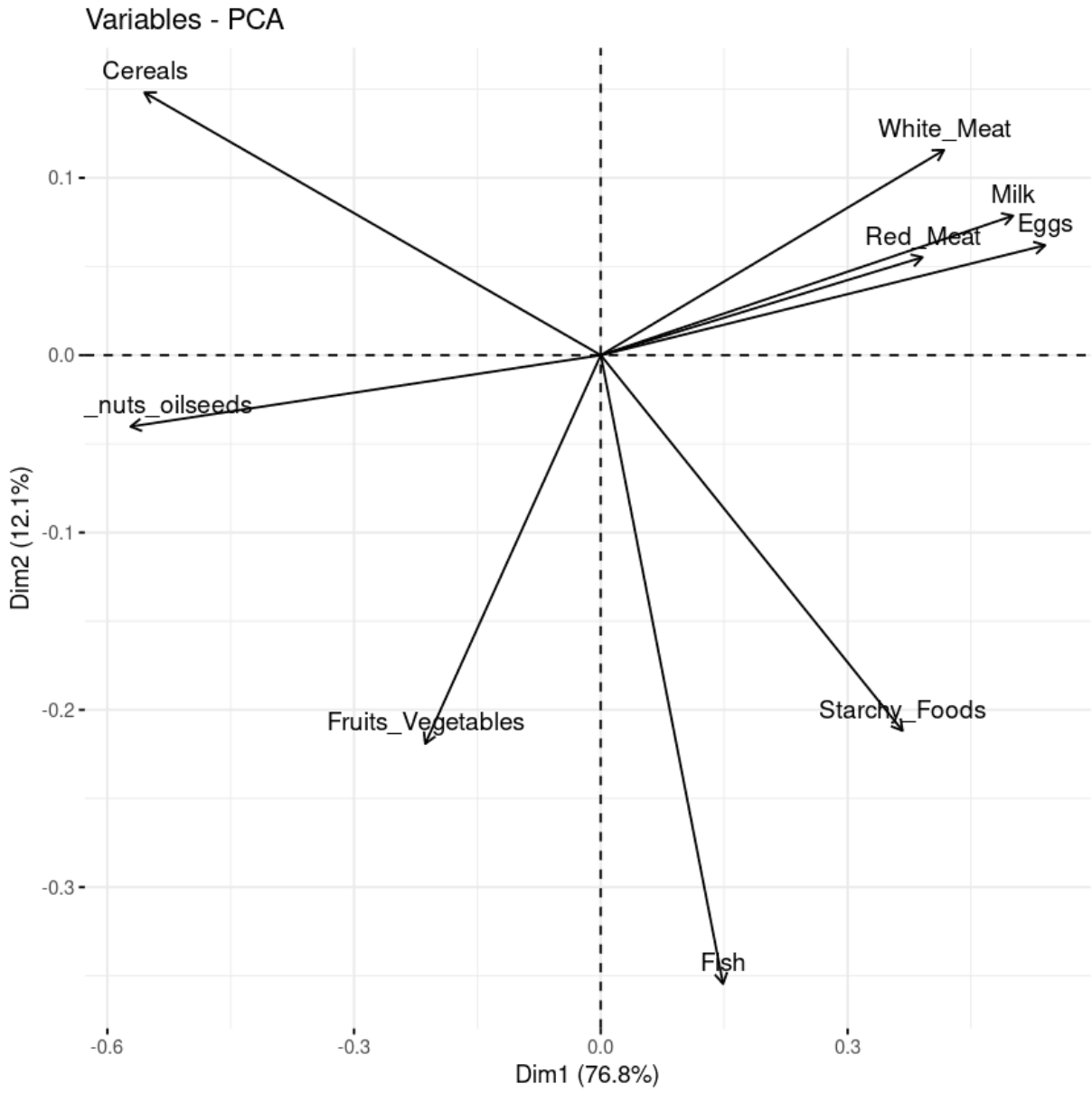
Scree Plot



- This plot shows the eigenvalues in a downward curve, from highest to lowest.
- The first two components can be considered to be the most significant since they contain almost 89% of the total information of the data.

Biplot of the attributes

- All the variables that are grouped together are positively correlated to each other, and that is the case for instance for white/red meat, milk, and eggs have a positive correlation to each.
- Eggs, milk, and white meat have higher magnitude compared to red meat, and hence are well represented compared to red meat.
- Variables that are negatively correlated are displayed to the opposite sides of the biplot's origin.



Exercise

- Use the R functions listed on the datacamp web (link is provided) to apply PCA on the customer data in the link below:
- https://raw.githubusercontent.com/raoy/data/master/Customer%20Data%20v.2.csv

References

- https://www.datacamp.com/tutorial/pca-analysis-r
- Other relevant resources

Association Rules

Lecture 12

Teknik Pembelajaran Mesin (STA1382)

rahmaanisa@app.ipb.ac.id

Outline

- Understanding association rule
- The Apriori algorithm
- Example – identifying frequently purchased groceries

The R code in this lecture is provided on the reference book and rpubs (https://rpubs.com/r_anisa/apriori)

Understanding Association Rule

Association Rules

- Typical rule:
  - {peanut butter, jelly} → {bread}
- Association rules are learned from subsets of **itemsets**
- The preceding rule was identified from the set {peanut butter, jelly, bread}

Association Rules

- Association rules are not used for prediction
- It is for unsupervised knowledge discovery in large databases
- Since it is **unsupervised**, there is no need for labeling.
- The program is simply unleashed on a dataset in the hope that interesting associations are found
- The downside: no objective metrics to measure the performance
- It mostly used for **market basket analysis**
- Its also useful for finding pattern

Potential Application

SEARCHING PATTERNS OF DNA AND PROTEIN SEQUENCES

FINDING PATTERN OF PURCHASES

IDENTIFYING COMBINATIONS OF CUSTOMER BEHAVIOR

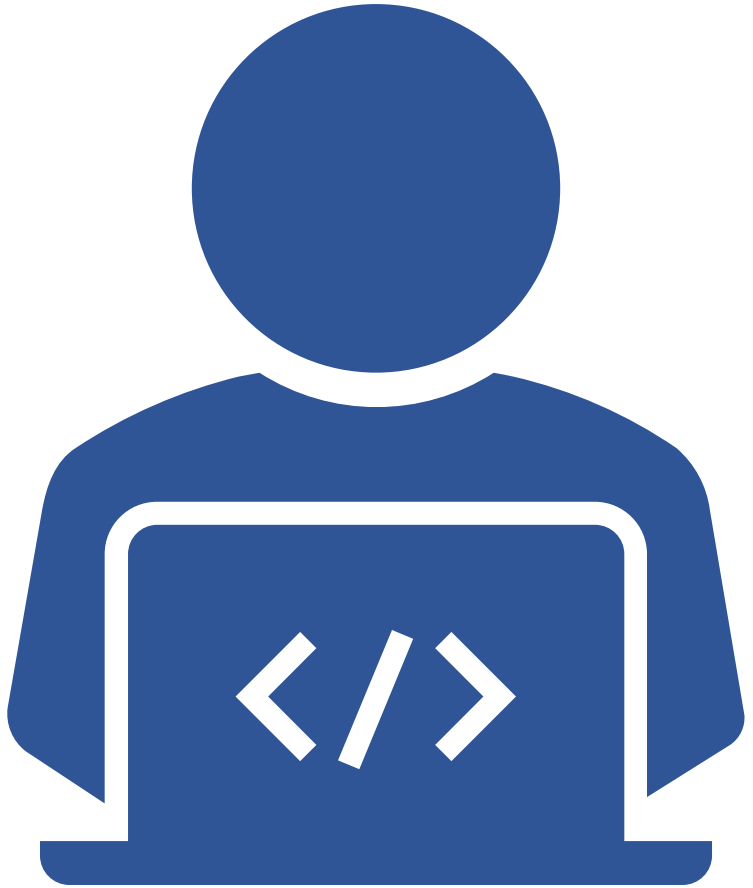
The Apriori Algorithm

Apriori Algorithm

- Transactional datasets is quite complex
  - It is extremely large: both in numbers and features
  - # of potential itemsets grows exponentially with # of features
- The algorithm identify some combinations of items are rarely found
  - i.e {motor oil, lipstick} → very uncommon
  - Ignoring this rare combinations → more manageable size

Apriori Algorithm

- Introduced in 1994 by R. Agrawal & R. Srikant
- The name is derived from the fact that the algorithm utilizes a simple prior **(a prior)** belief about the properties of frequent itemsets



Apriori Algorithm

Strengths

Ideally suited for very large amounts of transactional data

The results are easy to understand

Useful for discovering unexpected knowledge in the database

Weakness

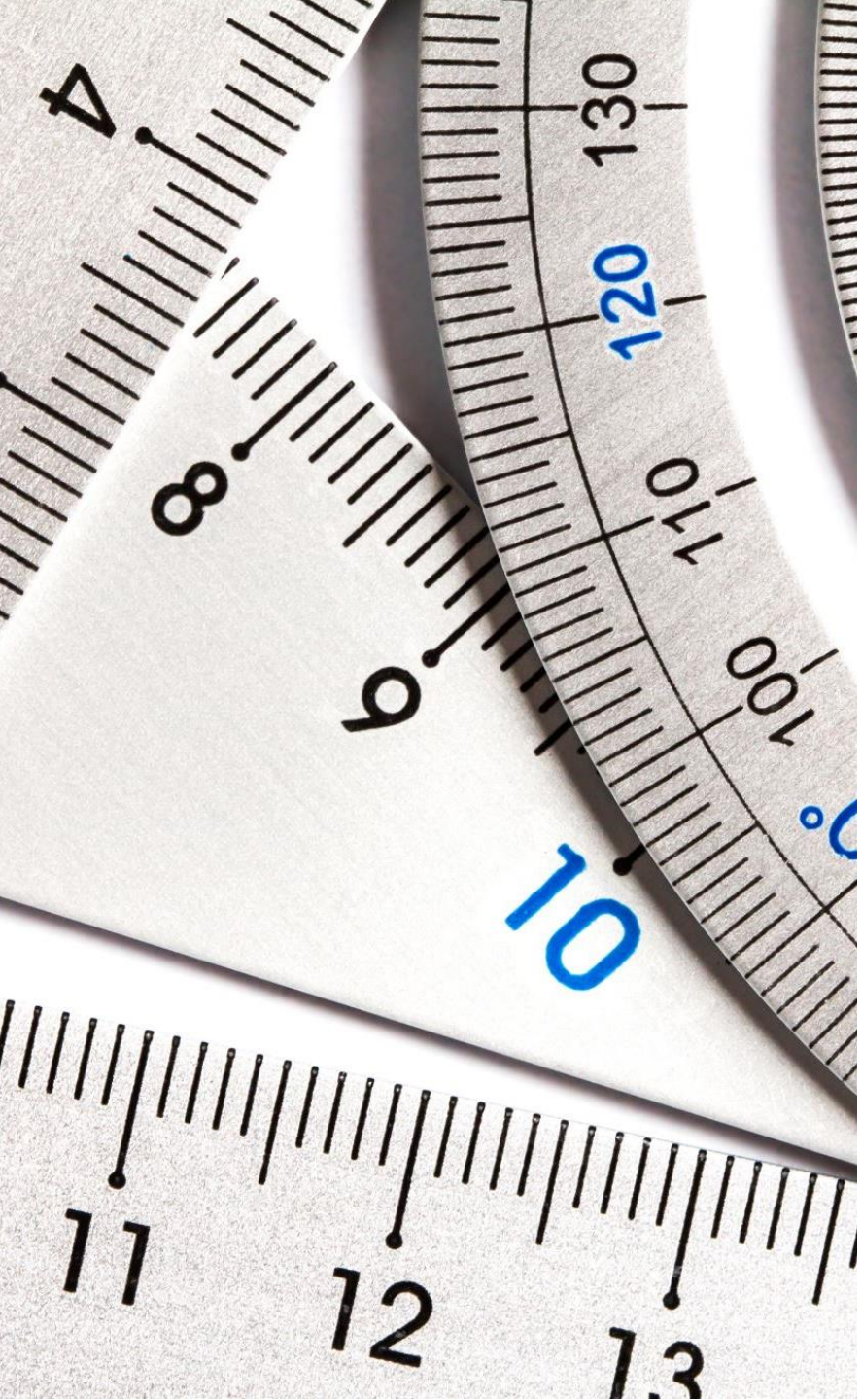
Not very helpful for small datasets

Takes effort to separate the insight from the common sense

Easy to draw spurious conclusions from random patterns

Apriori Property

- All subsets of a frequent itemset must also be frequent
- Hence, in {motor oil, lipstick} itemset, if both are **infrequent**, then any set containing these items can be **excluded** from search.



Measuring Rule Interest

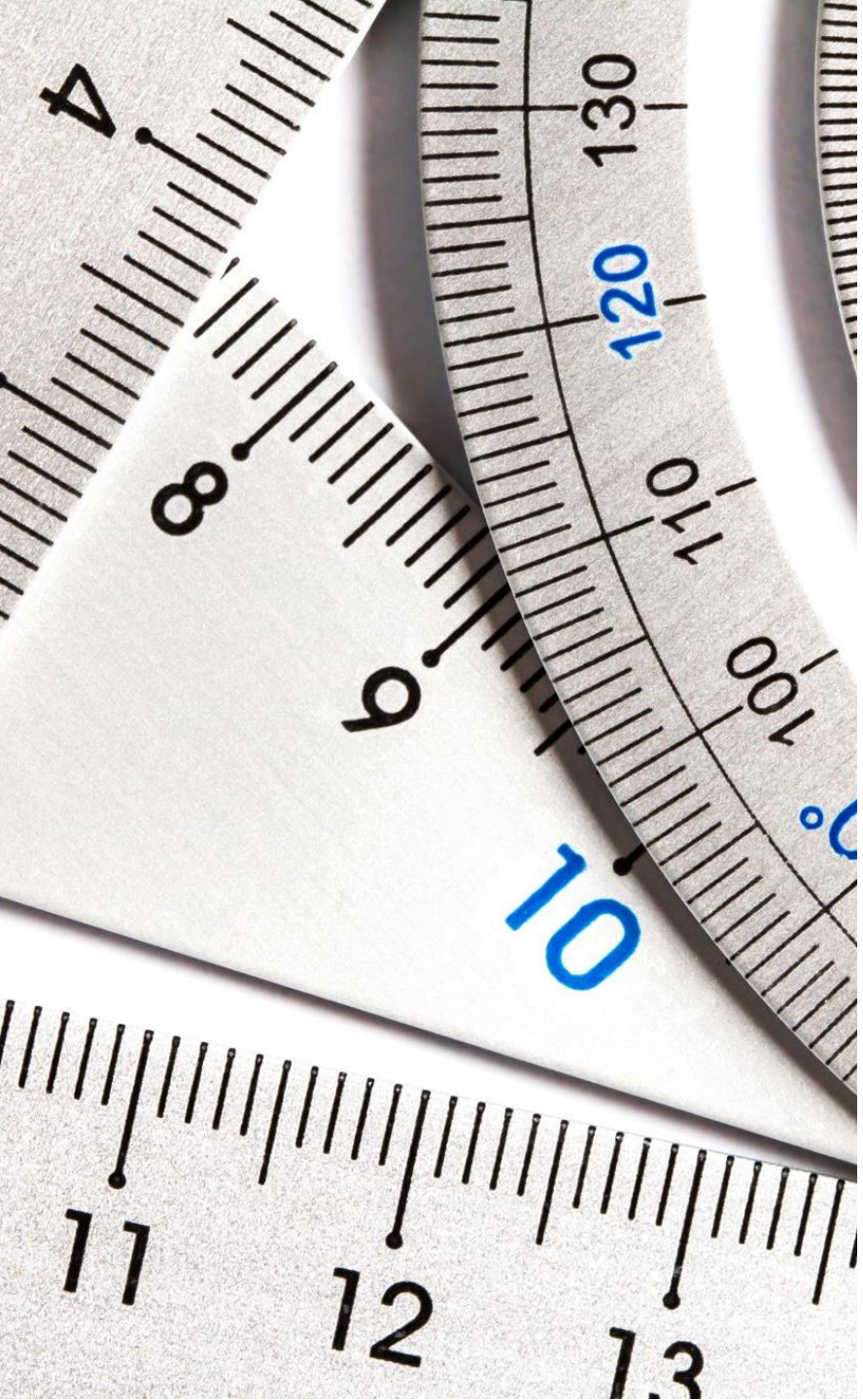
- Support → measures how frequently the itemset occurs in the data

$$support(X) = \frac{count(X)}{N}$$

where N is # of transactions

- Confidence → measure its predictive power or accuracy

$$confidence(X \rightarrow Y) = \frac{support(X, Y)}{support(X)}$$



Measuring Rule Interest

- $support\ (A\ , B) = P(A\ \cap\ B)$
- $confidence(A \rightarrow B) = P(B|A)$

Simple Illustration

Transaction Number	Purchased Items
1	{flowers, get well card, soda}
2	{plush toy bear, flowers, balloons, candy bar}
3	{get well card, candy bar, flowers}
4	{plush toy bear, balloons, soda}
5	{flowers, get well card, soda}

- $Support\{get\ well\ card, flowers\} = \frac{3}{5} = 0.6$
- $Support\ (get\ well\ card\} = \frac{3}{5} = 0.6$
- $Support\ (flowers\} = \frac{4}{5} = 0.8$
- $Support\ (candy\ bar\} = \frac{2}{5} = 0.4$

- $Confidence\{flowers\ \rightarrow get\ well\ card\} = \frac{0.6}{0.8} = 0.75$
- $Confidence\{get\ well\ card\ \rightarrow flowers\} = \frac{0.6}{0.6} = 1$

This means that a purchase involving flowers results is accompanied by a purchase of a get well card 75% of the time, while a purchase of a get well card is associated with flowers 100% of the time.

Building a Set of Rules with Apriori Principle

- Reminders:
  - If $\{A, B\}$ is frequent, then $\{A\}$ and $\{B\}$ both must be frequent
  - If we know that $\{A\}$ does not meet a desired support threshold, there is no reason to consider $\{A, B\}$ or any itemset containing $\{A\}$

- The actual process occurs in TWO phases:
  1) Identifying all itemsets that meet a minimum support threshold
  2) Creating rules from these itemsets that meet a minimum confidence threshold

Example

Identifying frequently purchased groceries with association rules

Step 1: Collecting data

Groceries data set

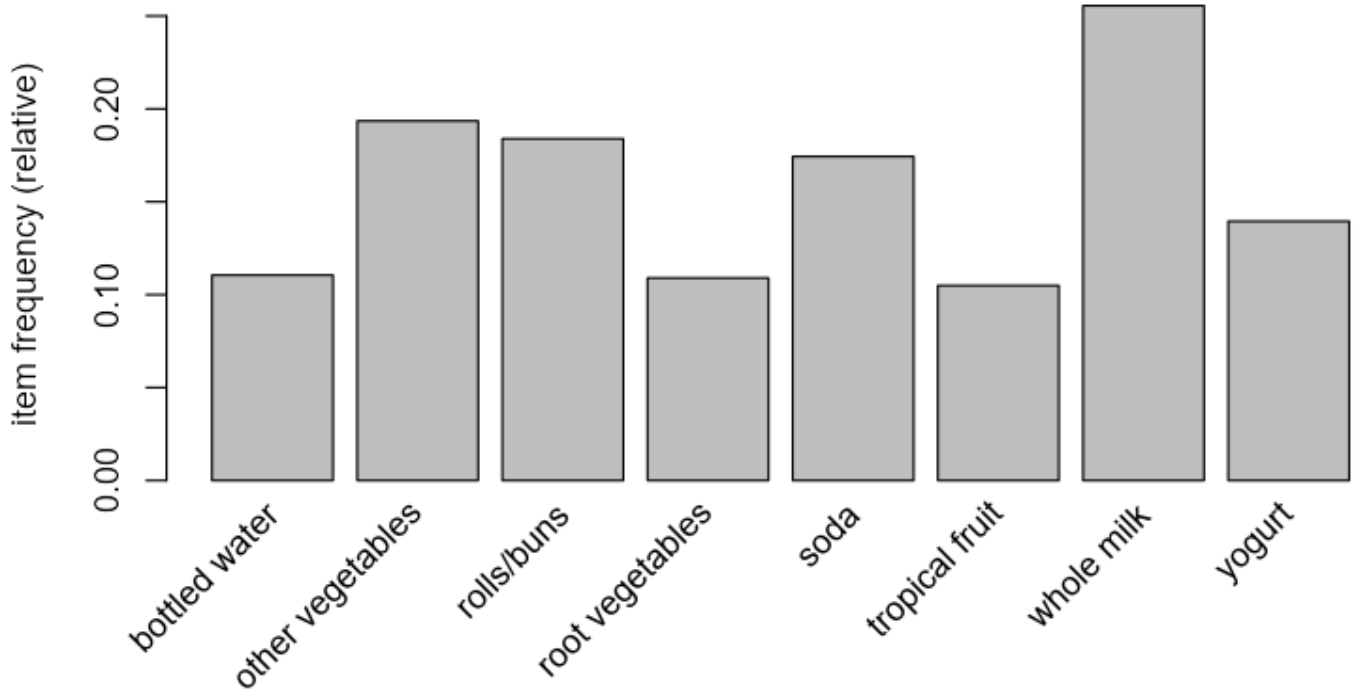
- Purchase data from one month of operation
- Data contain 9,835 transactions (about 327 transactions per day, roughly 30 transactions per hour in a 12 hour business day)
- We will assume that they are not concerned with a specific brands, so all brands names can be removed from the data
- Therefore, the number of groceries were reduced into 169 types, using broad categories such as chicken, frozen meals, margarines, soda, etc.

Step 2: Exploring and preparing the data

transactions

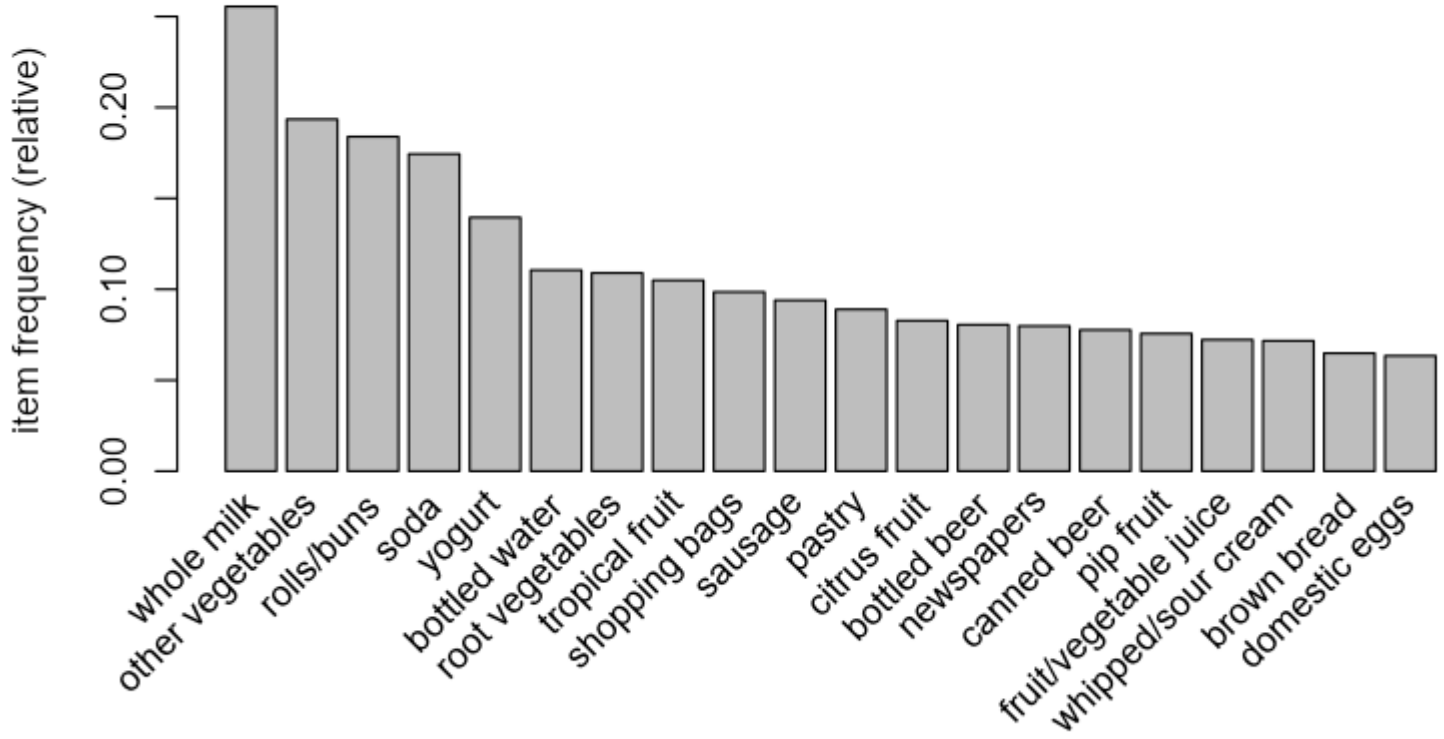
	V1	V2	V3	V4
1	citrus fruit	semi-finished bread	margarine	ready soups
2	tropical fruit	yogurt	coffee	
3	whole milk			
4	pip fruit	yogurt	cream cheese	meat spreads
5	other vegetables	whole milk	condensed milk	long life bakery product
6	whole milk	butter	yogurt	rice
7	abrasive cleaner			
8	rolls/buns			
9	other vegetables	UHT-milk	rolls/buns	bottled beer
10	liquor (appetizer)			
11	pot plants			
12	whole milk	cereals		
13	tropical fruit	other vegetables	white bread	bottled water
14	chocolate			

Step 2: Exploring and preparing the data



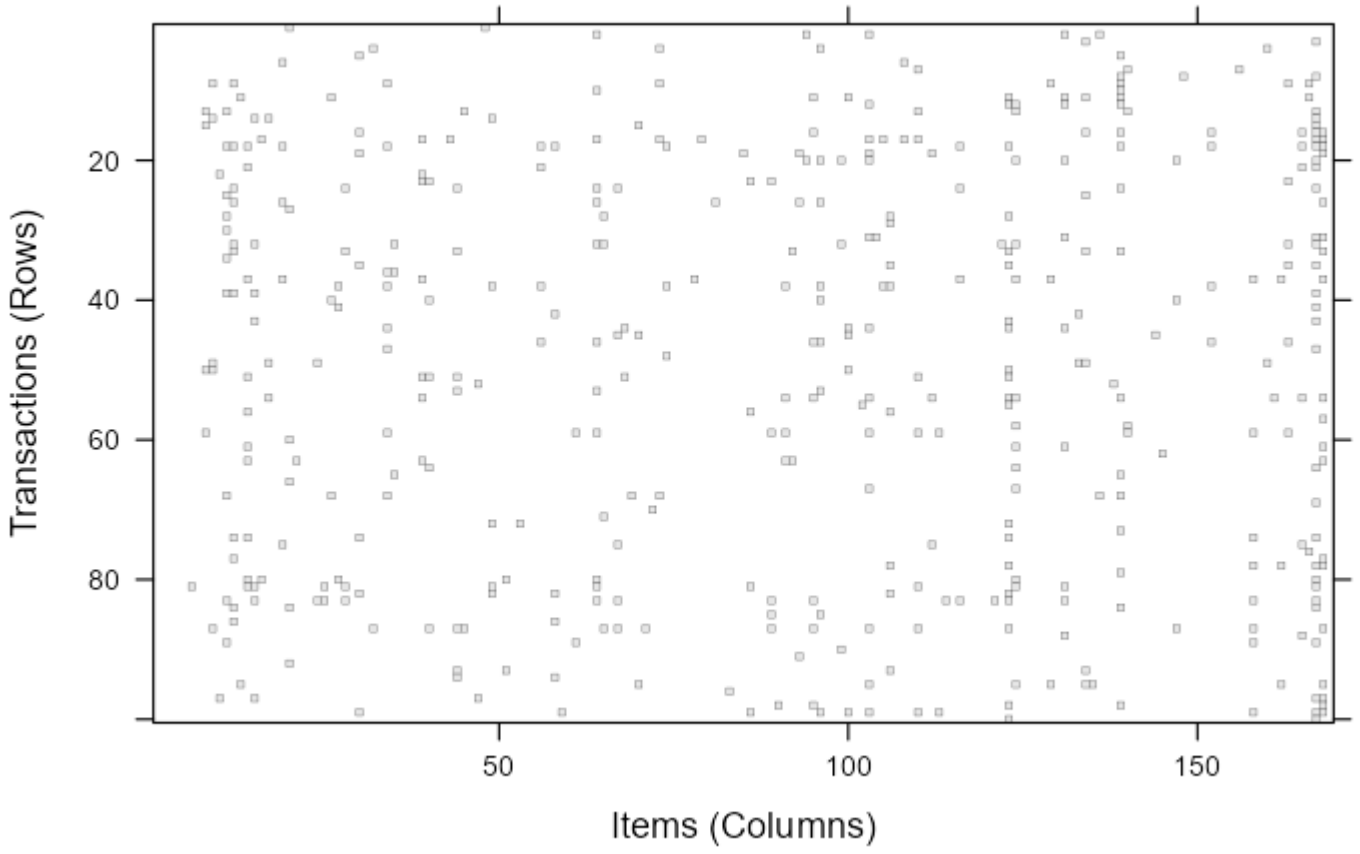
The graph shows the eight items in the groceries data with at least 10% support.

Step 2: Exploring and preparing the data



The top 20 items in the groceries data.

Step 2: Exploring and preparing the data



A few column seem fairly heavily populated, indicating some very popular items at the store.

However, the overall distributions seems fairly random.

Step 3: Training a model on the data

```
groceryrules<-apriori(groceries, parameter=list(support=0.006,
                                                confidence=0.25,
                                                minlen=2))
```

- We set minimum support of 0.006 considering an item is purchased twice a day (60 times out of 9,835, which is equal to 0.006).
- We set a confidence threshold of 0.25, meaning that in order to be included in the results, the rule has to be correct at least 25% of the time.
- In addition, we set minlen=2 to eliminate rules that contain fewer than two items.

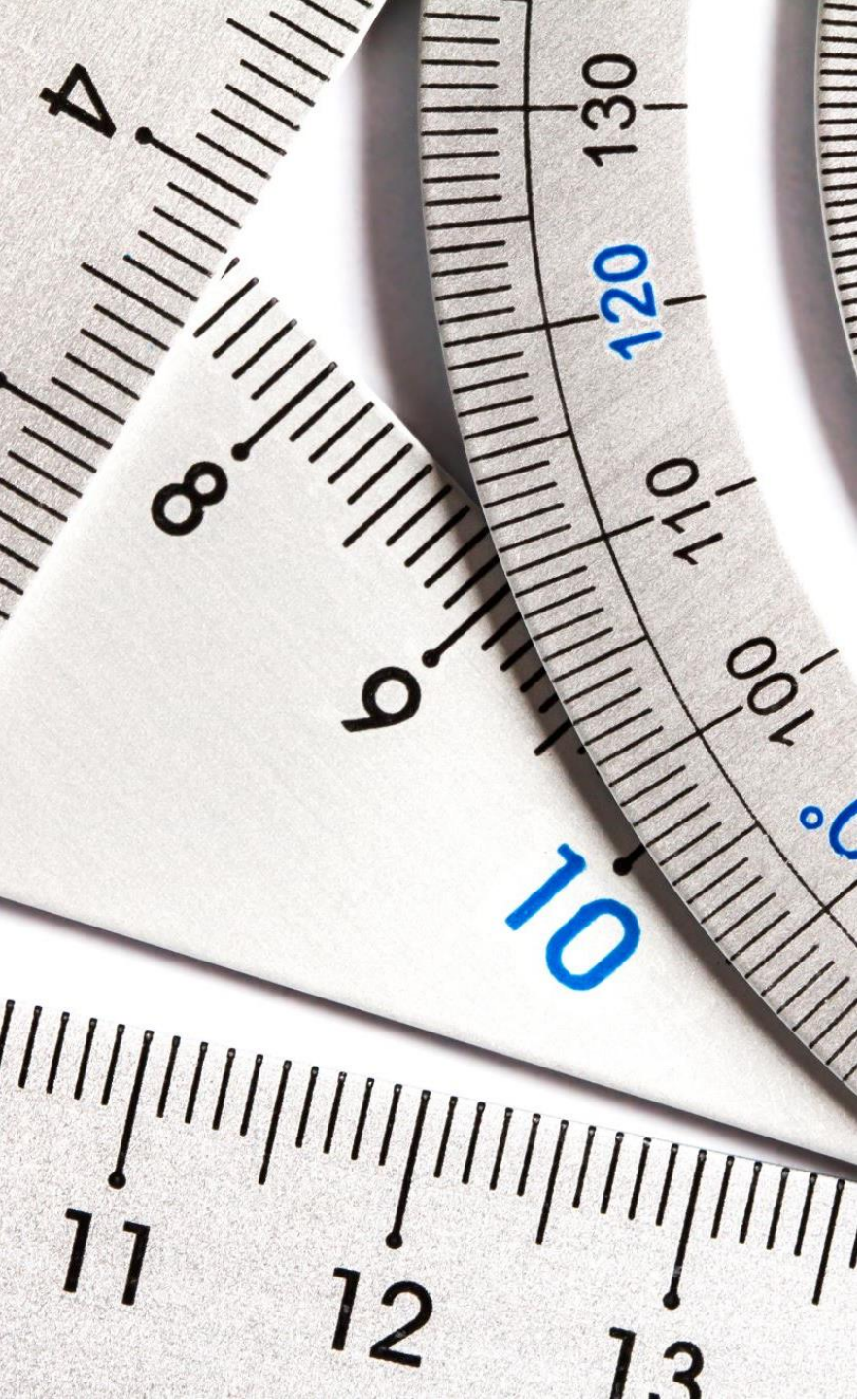
Step 4: Evaluating model performance

```
set of 463 rules

rule length distribution (lhs + rhs):sizes
  2   3   4
150 297  16

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  2.000   2.000   3.000   2.711   3.000   4.000

summary of quality measures:
    support           confidence        coverage             lift            count
 Min.   :0.006101   Min.   :0.2500   Min.   :0.009964   Min.   :0.9932   Min.   : 60.0
 1st Qu.:0.007117   1st Qu.:0.2971   1st Qu.:0.018709   1st Qu.:1.6229   1st Qu.: 70.0
 Median :0.008744   Median :0.3554   Median :0.024809   Median :1.9332   Median : 86.0
 Mean   :0.011539   Mean   :0.3786   Mean   :0.032608   Mean   :2.0351   Mean   :113.5
 3rd Qu.:0.012303   3rd Qu.:0.4495   3rd Qu.:0.035892   3rd Qu.:2.3565   3rd Qu.:121.0
 Max.   :0.074835   Max.   :0.6600   Max.   :0.255516   Max.   :3.9565   Max.   :736.0
```



Other Metric

- Lift : measure of how much likely one item to be purchased relative to its typical purchase rate, given that you know another item has been purchased.

  $$lift(X \rightarrow Y) = \frac{confidence(X \rightarrow Y)}{support\ (Y)}$$

  Note that $lift(X \rightarrow Y) = lift(Y \rightarrow X)$

- Large lift value indicates that a rule is important, and reflects a true connection between the items.

Step 4: Evaluating model performance

Description: df [3 × 8]

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{pot plants}	=>	{whole milk}	0.006914082	0.4000000	0.01728521	1.565460	68
[2]	{pasta}	=>	{whole milk}	0.006100661	0.4054054	0.01504830	1.586614	60
[3]	{herbs}	=>	{root vegetables}	0.007015760	0.4312500	0.01626843	3.956477	69

$\{pot\ plants\} \rightarrow \{whole\ milk\}$

- Confidence and lift are quite high
- Not very useful, because there is no logical reason explaining the purchase of those combination.

Step 4: Evaluating model performance

Type of rules:

- Actionable → provide a clear and useful insight
- Trivial → any rules that are so obvious that they are not worth
- Inexplicable → unclear association

Step 5: Improving model performance

Sorting the Rules by Lift Value

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>
[1]	{herbs}	=>	{root vegetables}	0.007015760	0.4312500	0.01626843	3.956477
[2]	{berries}	=>	{whipped/sour cream}	0.009049314	0.2721713	0.03324860	3.796886
[3]	{other vegetables, tropical fruit, whole milk}	=>	{root vegetables}	0.007015760	0.4107143	0.01708185	3.768074
[4]	{beef, other vegetables}	=>	{root vegetables}	0.007930859	0.4020619	0.01972547	3.688692
[5]	{other vegetables, tropical fruit}	=>	{pip fruit}	0.009456024	0.2634561	0.03589222	3.482649

$\{herbs\} \rightarrow \{root\ vegetables\}$

- $lift = 3.956477 \approx 4$
- People who buy herbs are nearly four times more likely to buy root vegetables than the typical customer.

Step 5: Improving model performance

Taking subsets of association rules

	lhs <chr>		rhs <chr>	support <dbl>	confidence <dbl>	coverage <dbl>	lift <dbl>	count <int>
[1]	{berries}	=>	{whipped/sour cream}	0.009049314	0.2721713	0.0332486	3.796886	89
[2]	{berries}	=>	{yogurt}	0.010574479	0.3180428	0.0332486	2.279848	104
[3]	{berries}	=>	{other vegetables}	0.010269446	0.3088685	0.0332486	1.596280	101
[4]	{berries}	=>	{whole milk}	0.011794611	0.3547401	0.0332486	1.388328	116

- It is reasonable to pair berries with whipped/sour cream, and yogurt in the store.

References

- Lantz, B. (2013). Machine learning with R: learn how to use R to apply powerful machine learning methods and gain an insight into real-world applications. Packt Publishing.

Ensemble Learning

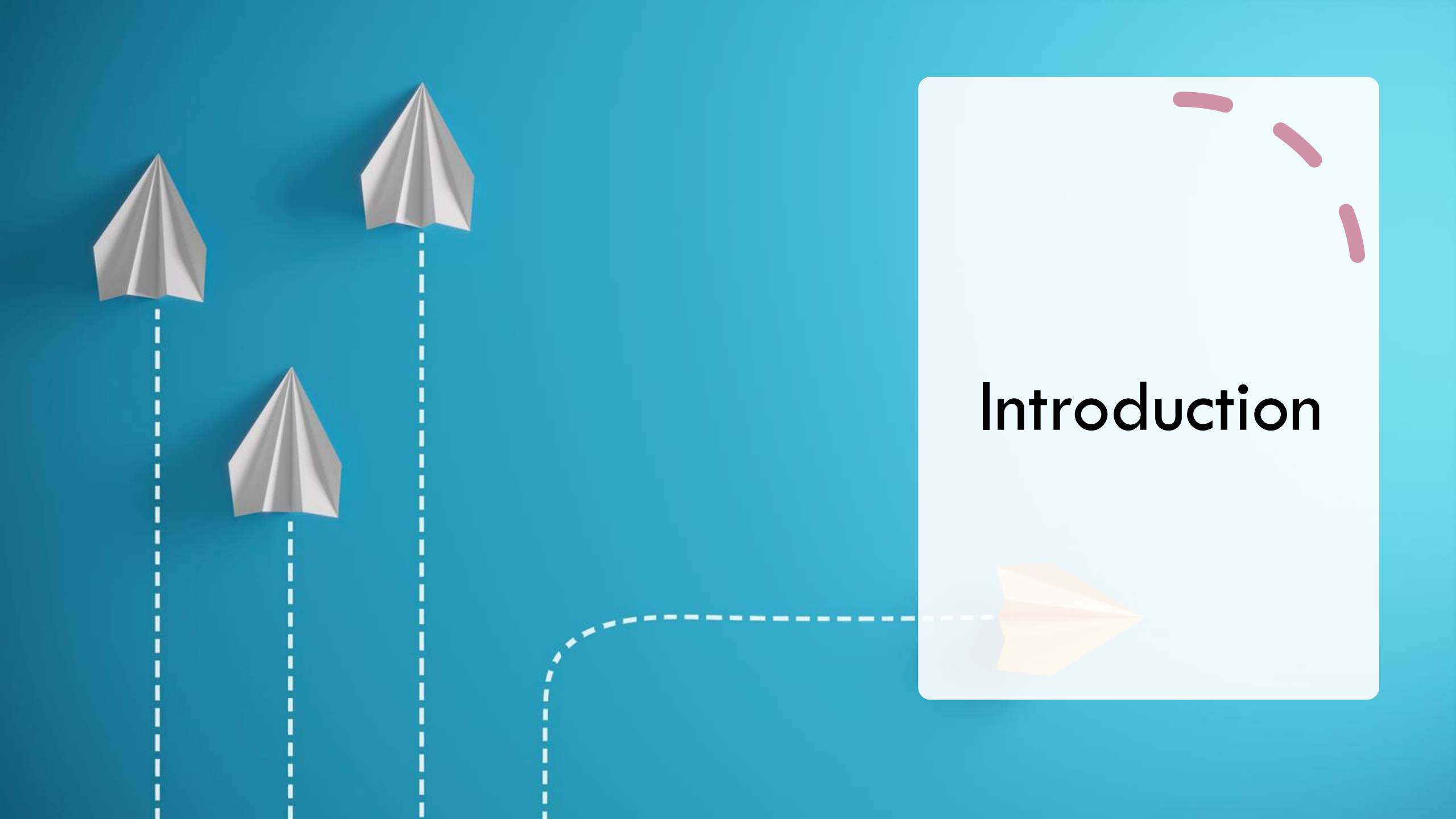
Kuliah 13

Teknik Pembelajaran Mesin (STA1382)

rahmaanisa@apps.ipb.ac.id

Outline

- Introduction
- Bagging & Boosting
- Random Forest



Introduction

Basic Ideas

- Suppose we want to predict default status of customer based on the following predictors:

age	Age in years
ed	Level of education
employ	Years with current employer
address	Years at current address
income	Household income in thousands
debtinc	Debt to income ratio (x100)
creddebt	Credit card debt in thousands
othdebt	Other debt in thousands

- We can use: binary logistic regression (BLR), discriminant analysis (DA), etc

Basic Ideas

Customer	BLR	DA	Actual
1	0	1	0
2	1	0	1
3	0	0	0
4	0	0	0
5	0	0	0
6	1	1	1
7	1	0	0
8	0	1	0
…			

note: 1 → default; 0 → not default

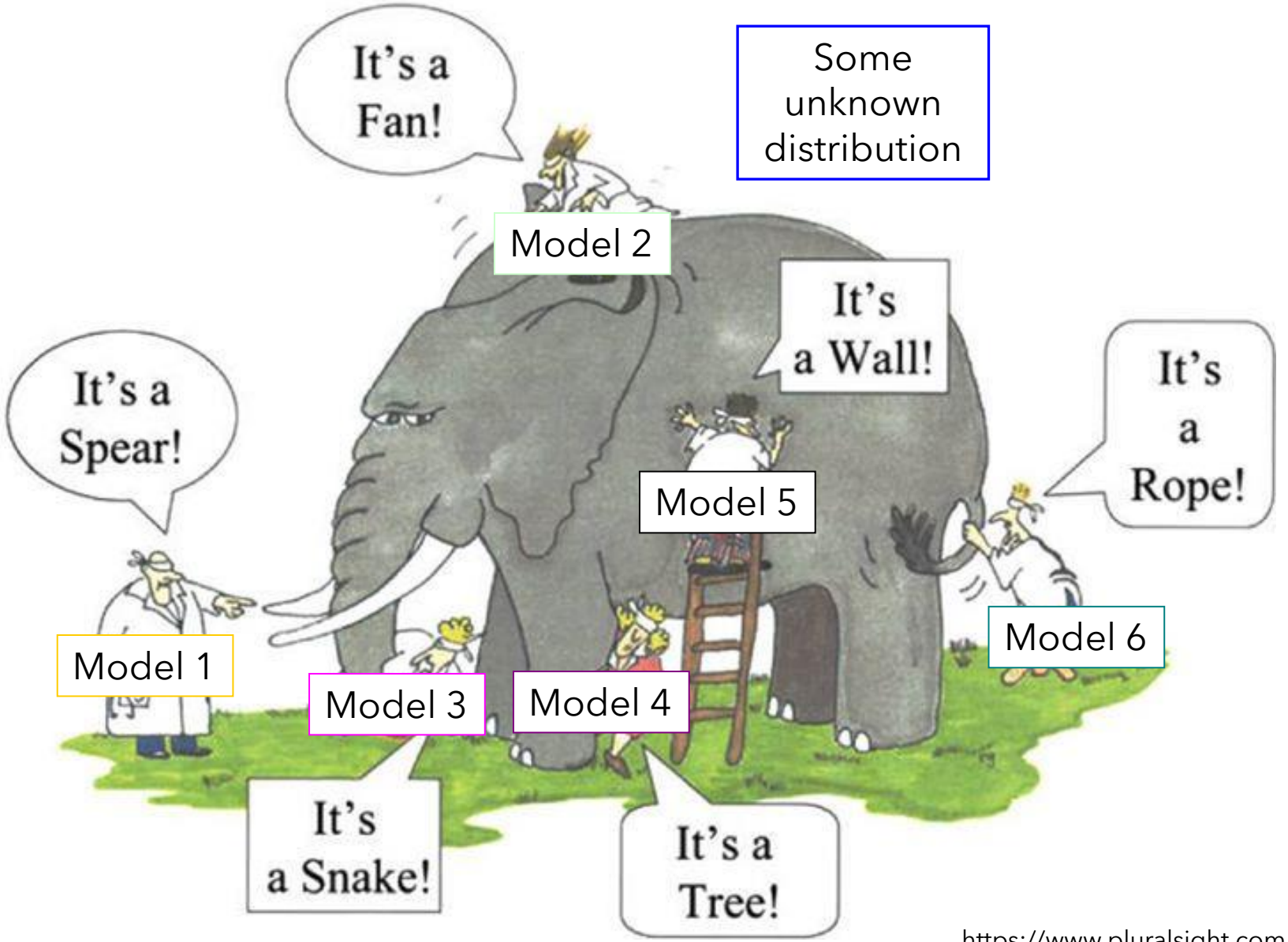
- After modeling, we obtain classification accuracy:
  - BLR: 80%
  - DA: 78%

  which method we choose?
- Can we combine both method so that the accuracy will increase?
  → Ensemble approach

Rationale

- There is no algorithm that is always the most accurate
- Generate a group of base-learners which when combined has higher accuracy
- Each algorithm makes assumptions which might be or not be valid for the problem at hand.
- Different learners use different
  - Algorithms
  - Hyperparameters
  - Representations (Modalities)
  - Training sets
  - Subproblems

Why Ensemble Works?



Ensemble gives the global picture!

https://www.pluralsight.com/guides/ensemble-methods:-bagging-versus-boosting

Why does it work?

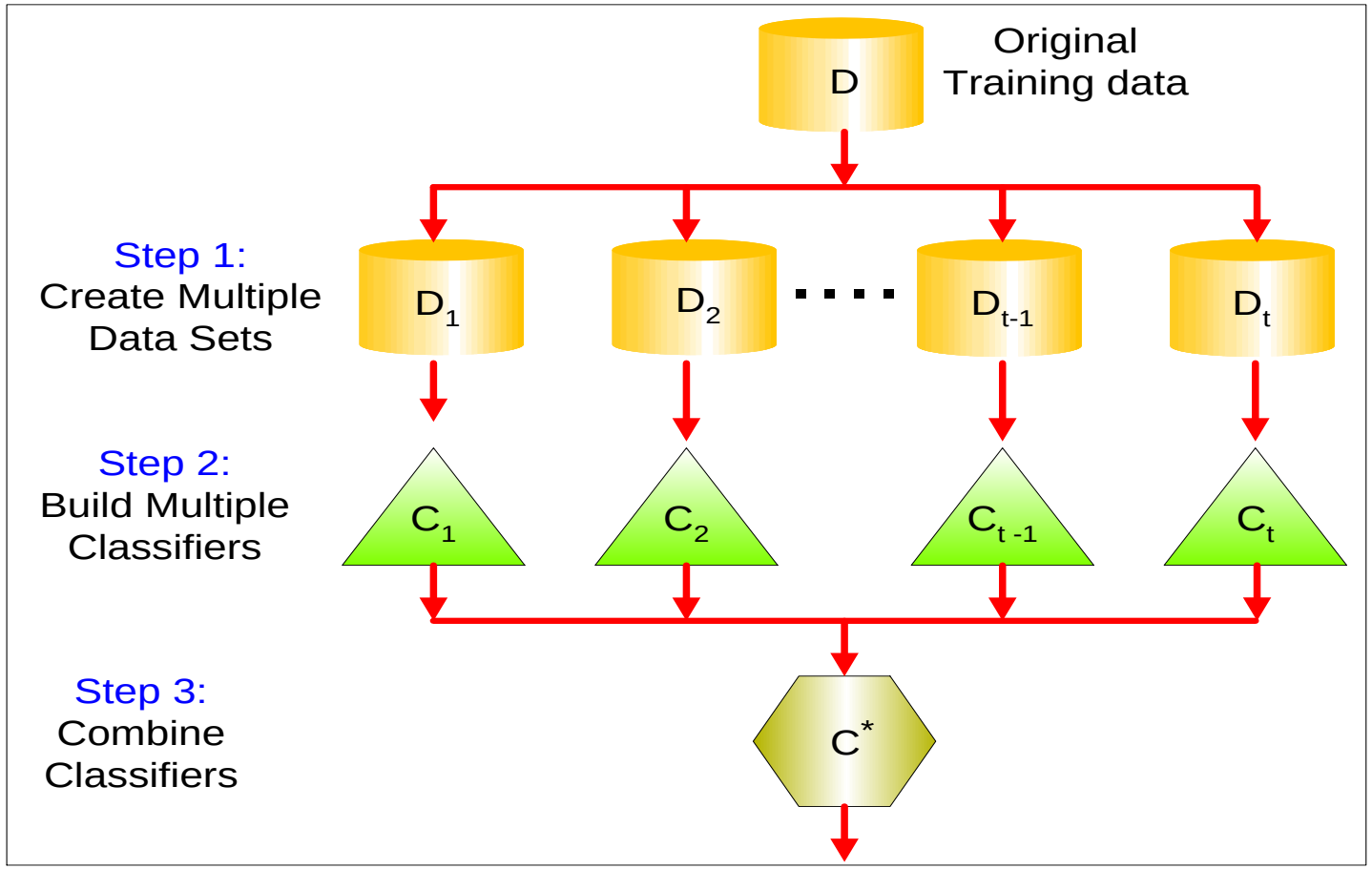
- Suppose there are 25 base classifiers
  - Each classifier has error rate, $\varepsilon = 0.35$
  - Assume classifiers are independent
  - Probability that the ensemble classifier makes a wrong prediction:

$$\sum_{i=13}^{25} \binom{25}{i} \varepsilon^i (1-\varepsilon)^{25-i} = 0.06$$

What is the Main Challenge for Developing Ensemble Models?

- The main challenge is not to obtain highly accurate base models, but rather to obtain base models which make different kinds of errors.
- High accuracies can be accomplished if different base models misclassify different training examples, even if the base classifier accuracy is low.

General Idea



Bagging & Boosting

Bagging

- Bagging stands for Bootstrap Aggregation
- Use bootstrapping to generate *L* training sets and train one base-learner with each
- Use voting (Average or median with regression)
- Unstable algorithms profit from bagging

Bagging

- Sampling with replacement

Original Data	1	2	3	4	5	6	7	8	9	10
Bagging (Round 1)	7	8	10	8	2	5	10	10	5	9
Bagging (Round 2)	1	4	9	1	2	3	2	7	3	2
Bagging (Round 3)	1	8	5	10	5	5	9	6	3	7

- Build classifier on each bootstrap sample

- Each sample has probability $(1 - 1/n)^n$ of being selected

Boosting

- An iterative procedure to adaptively change distribution of training data by focusing more on previously misclassified records
  - Initially, all N records are assigned equal weights
  - Unlike bagging, weights may change at the end of boosting round

Boosting

- Records that are wrongly classified will have their weights increased
- Records that are classified correctly will have their weights decreased

Original Data	1	2	3	4	5	6	7	8	9	10
Boosting (Round 1)	7	3	2	8	7	9	4	10	6	3
Boosting (Round 2)	5	4	9	4	2	5	1	7	4	2
Boosting (Round 3)	4	4	8	10	4	5	4	6	3	4

- Example 4 is hard to classify
- Its weight is increased, therefore it is more likely to be chosen again in subsequent rounds

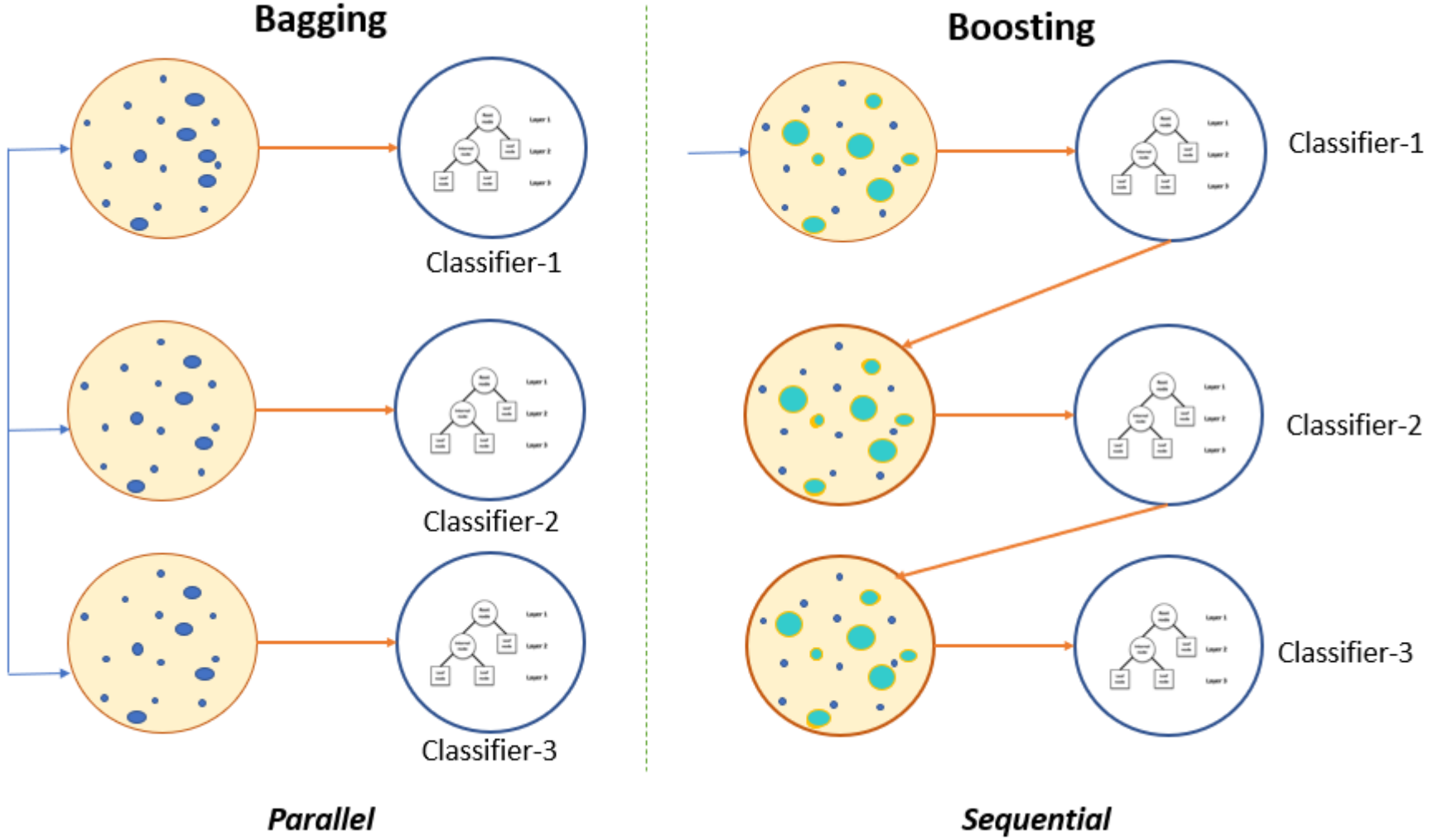
Bagging vs Boosting

- How do the N Learners generated

Single	Bagging	Boosting
		
Complete training set	Random sampling with replacement	Random sampling with repleacement over weighted data

Bagging vs Boosting

- Weighting the data elements

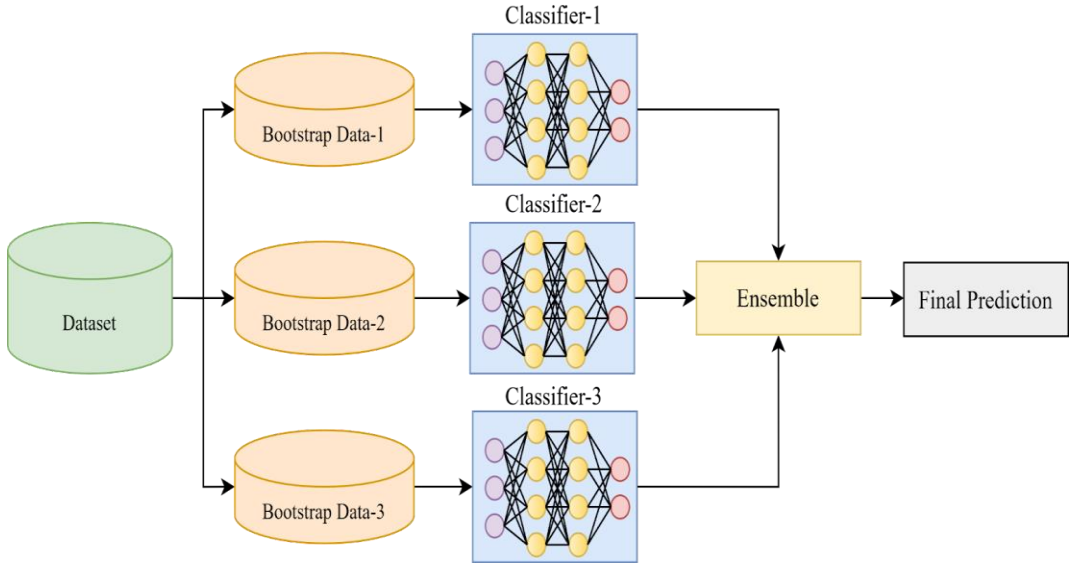


https://www.pluralsight.com/guides/ensemble-methods:-bagging-versus-boosting

Bagging vs Boosting

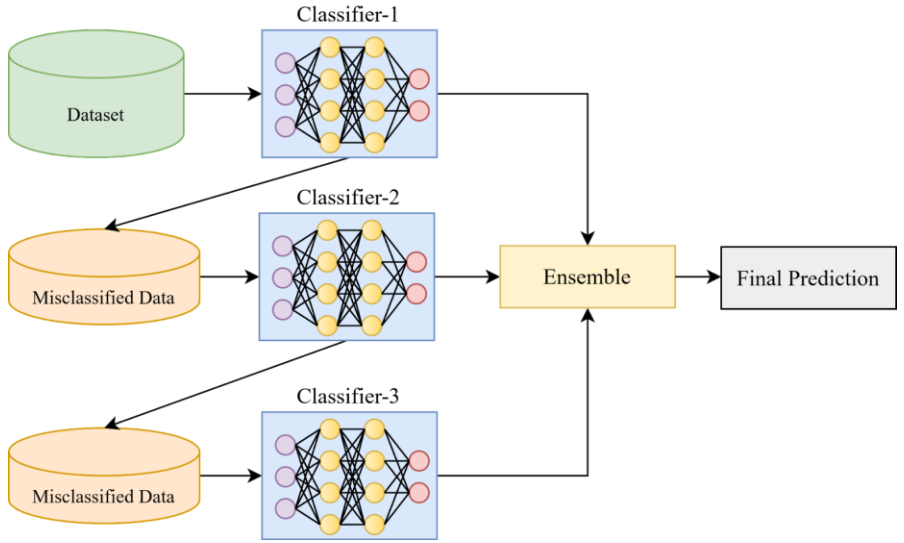
Bagging

- Averaging the responses of the N learners (or majority vote).



Boosting

- Uses weighted average of their estimates; a learner with good a classification result on the training data will be assigned a higher weight than a poor one



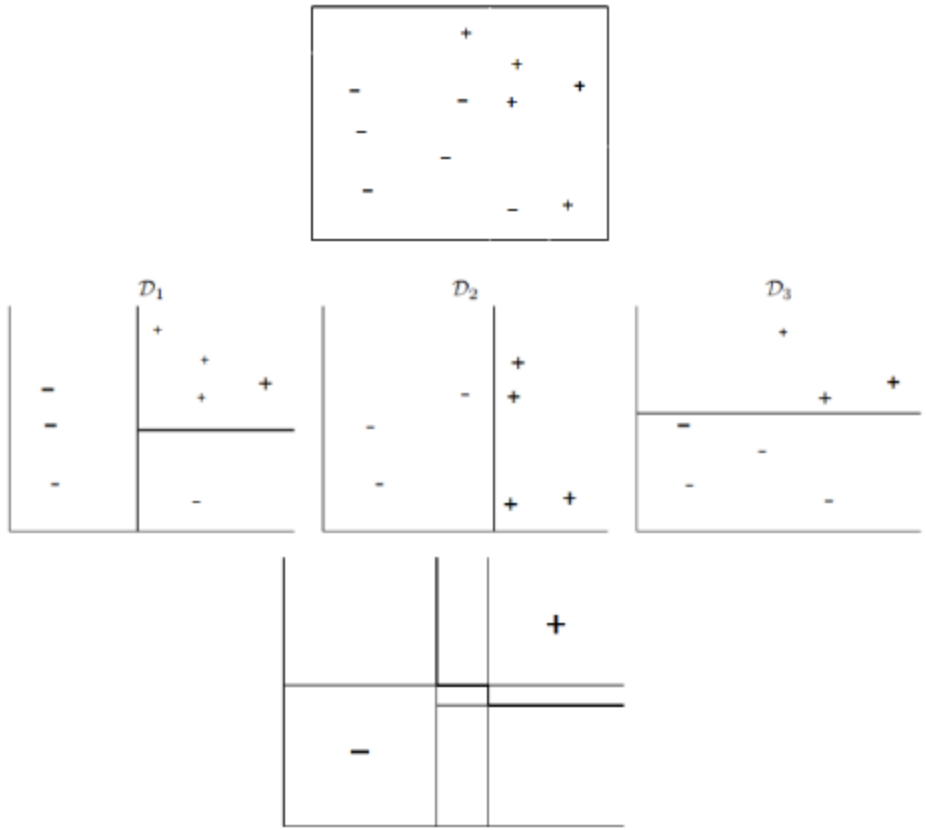
https://www.v7labs.com/blog/ensemble-learning

Bagging: Illustration

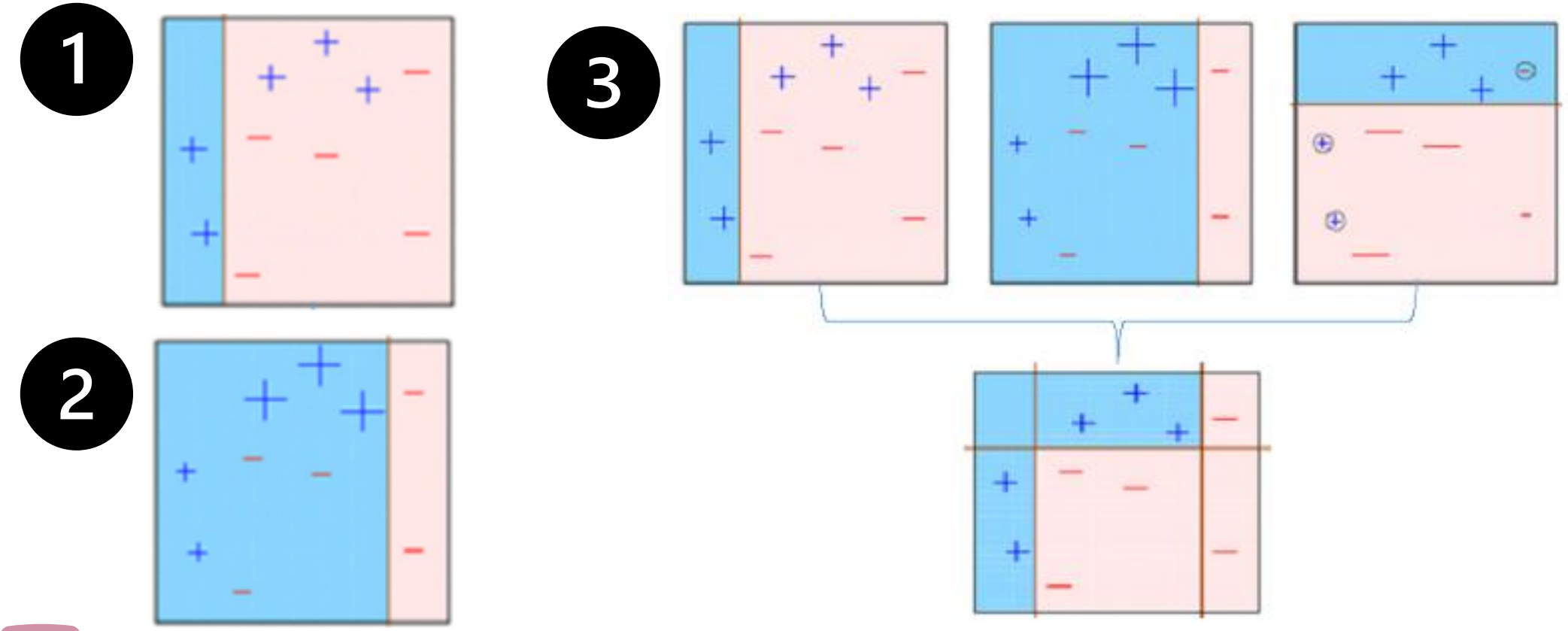
Original data

3 models learned from 3 datasets chosen using bootstrap

Averaged model



Boosting Illustration



https://medium.com/analytics-vidhya/intuition-behind-adaboost-5cda1e2e07ae

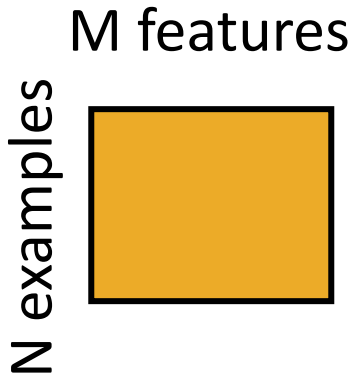
Random Forest

Random Forest

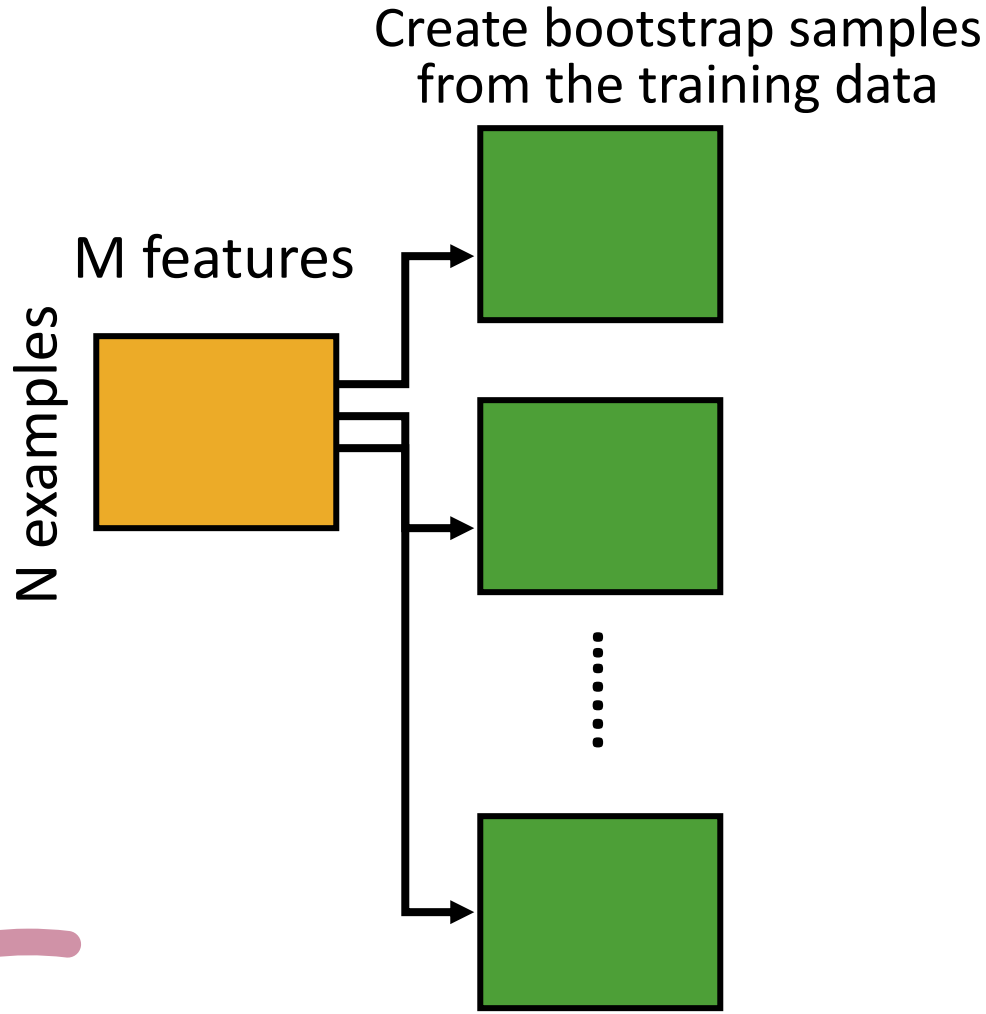
- Focus on ensembles of decision trees
- It was championed by Leo Breiman and Adele Cutler
- It combines the principle of bagging with random feature selection
- The model uses a vote to combine the trees' predictions

Random Forest Classifier

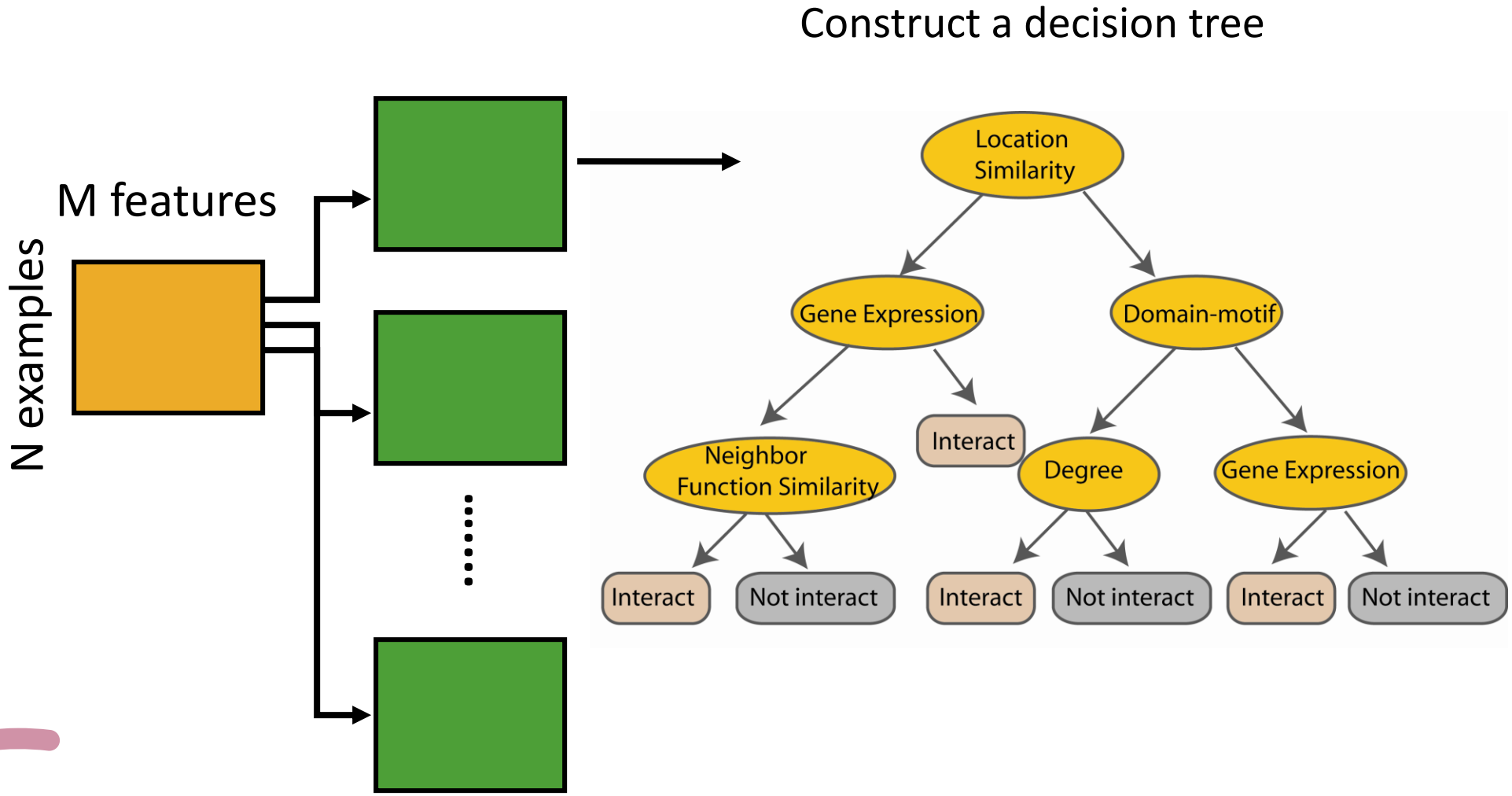
Training Data



Random Forest Classifier

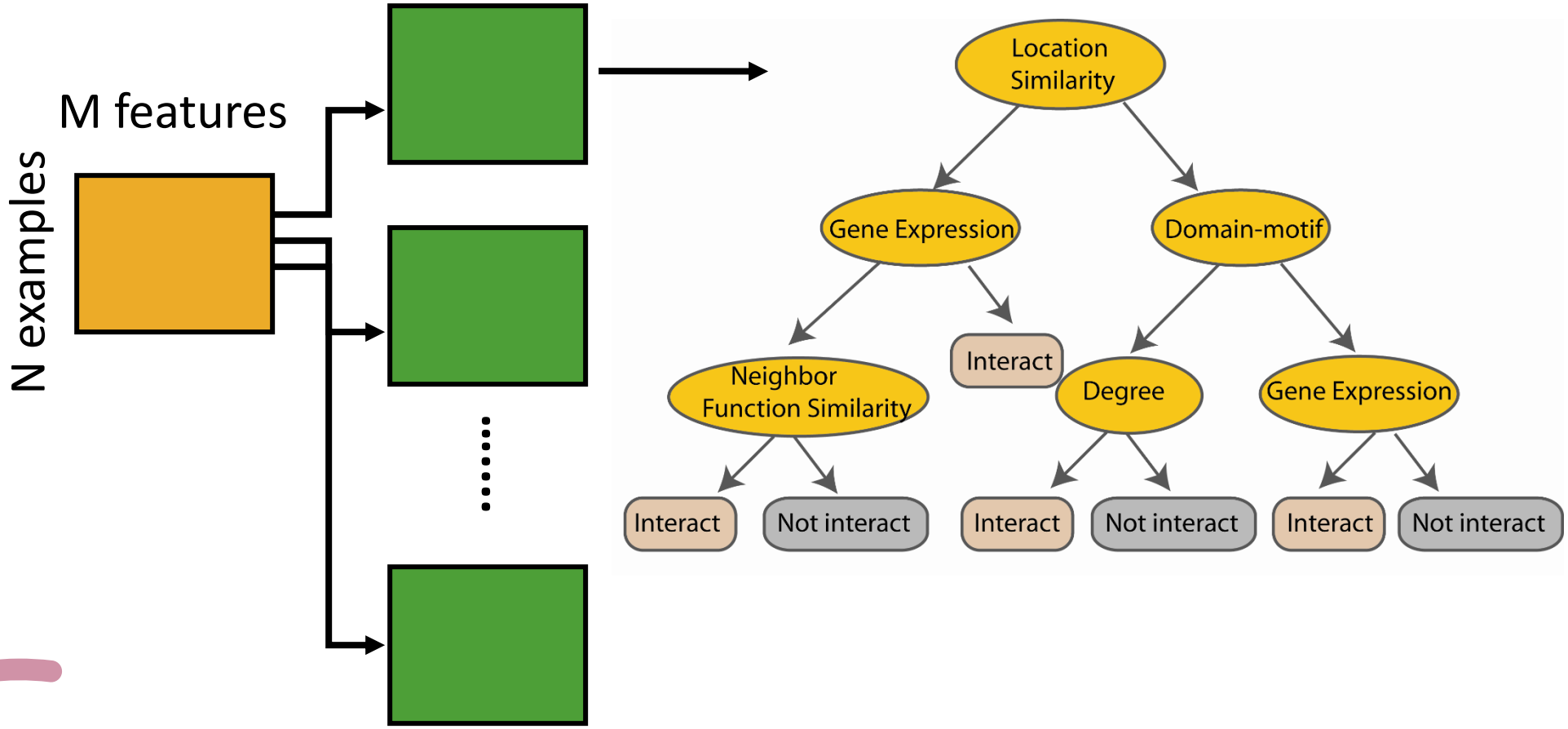


Random Forest Classifier

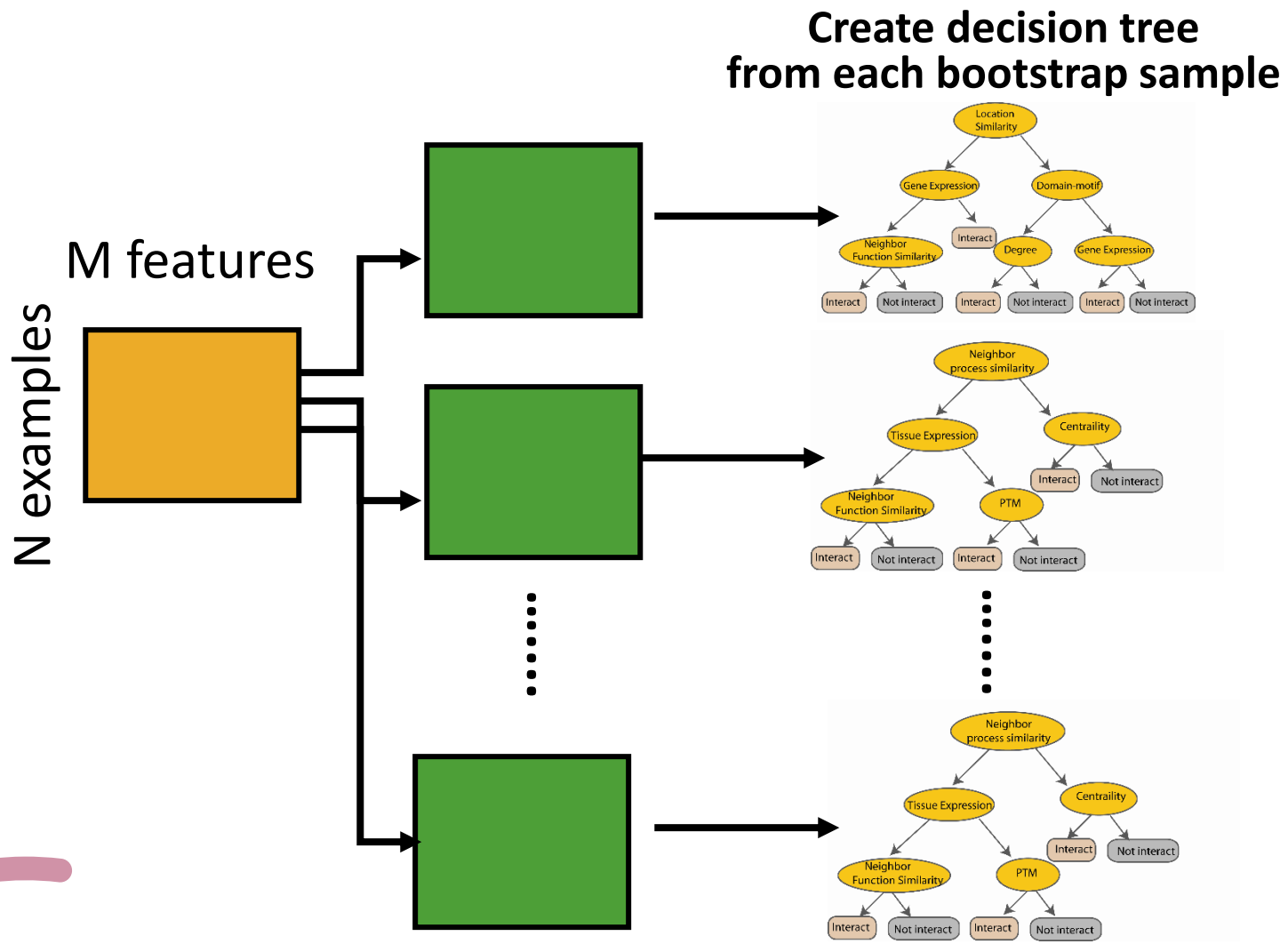


Random Forest Classifier

At each node in choosing the split feature choose only among $m<M$ features



Random Forest Classifier



Random Forest Classifier