



Image

Analisis Cluster dengan R

Author: Deden Istiawan

Contents

Chapter 1 Pengantar Analisis Cluster	2
1.1 Apa itu Analisis Cluster?	2
1.2 Apa itu Bahasa Pemrograman R?	2
1.3 Install R dan RStudio	2
1.4 Menginstal dan memuat R packages	3
1.5 Lorem Ipsum5	3
1.6 Lorem Ipsum6	4
Chapter 2 Algoritma K-Means	6
2.1 Pengantar Algoritma K-Means	6
2.2 Eksperimen Algoritma K-Means	6
Chapter 3 Algoritma Fuzzy C-Means	15
3.1 Pengantar Algoritma Fuzzy C-Means	15
3.2 Eksperimen Fuzzy C-Means	16
Chapter 4 Metode Cluster Hirarki	24
4.1 Data	24
4.2 Comparing dendrograms	27
Chapter 5 Evaluasi Analisis Cluster	35
5.1 Including Plots	35
Referensi	37

“Why is that when one man builds a wall, the next needs to know what’s on the other side?”

— Tyrion Lannister-Game of Thrones

Chapter 1 Pengantar Analisis Cluster

1.1 Apa itu Analisis Cluster?

Analisis *Cluster* mungkin salah satu yang paling banyak dipelajari dalam komunitas data mining dan *machine learning* (Bandyopadhyay, Saha, and Pedrycz 2011). Masalah ini telah dipelajari oleh para peneliti dari berbagai disiplin ilmu selama lima dekade. Penerapan analisis *cluster* mencakup berbagai domain masalah seperti teks, multimedia, jejaring sosial, dan data biologis. Clustering adalah topik yang agak beragam, dan algoritma yang mendasarinya sangat bergantung pada domain data dan skenario masalah (Aggarwal and Reddy 2014). *Clustering* adalah proses pengelompokan sekumpulan objek data menjadi beberapa kelompok atau *cluster* sehingga objek yang berada di dalam *cluster* memiliki kesamaan yang tinggi, tetapi sangat berbeda dengan objek di *cluster* lain (Han, Kamber, and Pei 2011). Pada buku ini akan dijelaskan konsep dasar dan metode analisis cluster, yang terdiri dari: metode pengelompokan partisi dan hirarki, Model Probabilistik dan evaluasi algoritma *clustering*.

Analisis *cluster* atau *clustering* adalah proses mengelompokkan sekumpulan objek data menjadi beberapa kelompok. Setiap subset adalah *cluster*, sehingga objek dalam *cluster* memiliki kemiripan yang sangat tinggi, namun berbeda dengan objek di *cluster* lain. Himpunan *cluster* yang dihasilkan dari analisis *cluster* dapat disebut sebagai *clustering*. Dalam konteks ini, algoritma *clustering* yang berbeda dapat menghasilkan pengelompokan yang berbeda pada dataset yang sama. Oleh karena itu, pengelompokan berguna karena dapat mengarah pada penemuan kelompok yang sebelumnya tidak diketahui dalam data.

1.2 Apa itu Bahasa Pemrograman R?

R adalah bahasa pemrograman dan software bebas yang dikembangkan oleh Ross Ihaka dan Robert Gentleman pada tahun 1993. R memiliki katalog ekstensif metode statistik dan grafis. Ini meliputi algoritma machine learning, regresi linier, time series, inferensi statistik untuk beberapa hal. R tidak hanya diandalkan oleh para akademisi, tetapi juga banyak perusahaan besar menggunakan bahasa pemrograman R, antara lain Uber, Google, Airbnb, Facebook, dan masih banyak lagi.¹

1.3 Install R dan RStudio

Untuk menginstall R, kamu tidak memerlukan budget untuk membeli lisensi. R merupakan aplikasi open source sehingga dapat kamu unduh secara gratis di laman aslinya. Berikut langkah penginstallan mulai dari cara download hingga R siap digunakan. Pertama buka [Halaman R Project](#). Pilih installer sesuai sistem operasi. Misalnya pilih **Download R for Windows** dan pilih **install R for the first time**. Tunggu proses unduh instaler dan jika sudah 100% terunduh klik file tersebut untuk mulai install R. setelah R berhasil diinstall langkah selanjutnya adalah menginstall program **RStudio**. Pada tahap ini kita akan install RStudio yaitu merupakan IDE yang umumnya digunakan untuk pemrograman R RStudio akan sangat membantu untuk melakukan coding R dan lainnya. Kita bisa langsung download installer versi terbaru atau jika ingin menggunakan versi sebelumnya bisa pilih all installer.

¹<https://wartaekonomi.co.id/read375198/apa-itu-r-programming-language>

1.4 Menginstal dan memuat R packages

Setelah selesai install R dan RStudio, serta memahami bagaimana menggunakannya, proses selanjutnya adalah instal R package. Kita dapat langsung melakukan instalasi R package yang sudah ada di CRAN repository (mayoritas R package ada di sini) dengan menuliskan perintah `install.packages("nama package")` pada RStudio atau di Console. Kita harus terhubung dengan koneksi internet untuk dapat instal package. Misal, kita instal package `cluster`.

```
install.packages("cluster")
```

selain melalui CRAN repository, intall packages juga dapat melauai Github. Anda harus terlebih dahulu harus menginstal devtools jika Anda belum menginstalnya di komputer Anda. Misalnya, kode R berikut menginstal paket factoextra R versi terbaru yang dikembangkan oleh A. Kassambara (<https://github.com/kassambara/factoextra>) untuk analisis data multivariat dan visualisasi yang elegan.

```
install.packages("devtools")
devtools::install_github("kassambara/factoextra")
```

Note

Note that, GitHub contains the developmental version of R packages

Setelah instalasi, pertama-tama Anda harus memuat paket untuk menggunakan fungsi-fungsi di dalam paket. Fungsi `library()`.

```
library(cluster)
```

Lemma 1.1

For any two random variables X_1, X_2 , they both have the same probability distribution if and only if

$$\varphi_{X_1}(t) = \varphi_{X_2}(t)$$



Theorem 1.1

If X_1, \dots, X_n are independent random variables, and a_1, \dots, a_n are some constants, then the characteristic function of the linear combination $S_n = \sum_{i=1}^n a_i X_i$ is

$$\varphi_{S_n}(t) = \prod_{i=1}^n \varphi_{X_i}(a_i t) = \varphi_{X_1}(a_1 t) \cdots \varphi_{X_n}(a_n t)$$



Proposition 1.1

The distribution of the sum of independent Poisson random variables $X_i \sim \text{Pois}(\lambda_i)$, $i = 1, 2, \dots, n$ is $\text{Pois}(\sum_{i=1}^n \lambda_i)$.



1.5 Lorem Ipsum5

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. ².

²<https://github.com/ElegantLaTeX/ElegantBook/blob/6ab10beda81252f0b478e05fa926199301347e4a/elegantbook.cls#L884>

```
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    TeX: {
      Macros: {
        bm: ["{\boldsymbol #1}",1],
      }
    }
  });
</script>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
<script type="text/x-mathjax-config">
  MathJax.Hub.Config({
    TeX: {
      Macros: {
        bm: ["{\boldsymbol #1}",1],
      },
      extensions: ["cancel.js"]
    }
  });
</script>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
\usepackage[makeroom]{cancel}
```

1.6 Lorem Ipsum6

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. <https://bookdown.org/yihui/rmarkdown-cookbook/custom-blocks.html>.

TODO: there is still a lot to do

Warning

Warning

Tips

Tips

Note

Note

Info

Chapter 2 Algoritma K-Means

2.1 Pengantar Algoritma K-Means

K-means merupakan salah satu metode hard partition yang banyak digunakan untuk pengelompokan data. Algoritma K-means adalah dasar pengelompokan metode partisi yang dipublikasikan oleh Lloyd dari Bell Telephone Laboratories pada tahun 1957. Penelitian pada K-means dapat ditelusuri kembali ke pertengahan abad yang lalu, yang dilakukan oleh berbagai peneliti diseluruh disiplin ilmu yang berbeda terutama oleh Lloyd (1957), Forgey (1965), Friedman dan Robin (1967) dan MacQueen (1967). K-means dapat didefinisikan sebagai algoritma klustering yang mengelompokkan data ke dalam k klaster berdasarkan jarak terdekat data dengan pusat klaster. Algoritma K-means sangat efisien untuk mengelompokkan dataset yang besar, kemudahan dalam pengaplikasiannya dan metode yang efisien dalam hal komputasi, menjadi alasan utama popularitas K-means, meskipun telah diusulkan lebih dari 50 tahun yang lalu.

Algoritma K-means mengelompokkan objek ke dalam kelompok sehingga objek dalam satu klaster memiliki kemiripan yang tinggi dibandingkan dengan objek di dalam klaster yang berbeda. K-means dimulai dengan menentukan jumlah klaster sebanyak k, kemudian membangkitkan k pusat klaster secara acak. Selanjutnya setiap objek akan dikelompokkan berdasarkan jarak terdekat dengan pusat klaster, pusat klaster diperbaharui berdasarkan titik data dalam setiap klaster. Proses ini diulangi sampai kriteria konvergen terpenuhi. Berikut ini adalah tahapan dari algoritma K-means:

1. Menentukan nilai k sebagai jumlah klaster yang dibentuk.
2. Memilih k pusat klaster secara acak untuk menjadi pusat klaster awal.
3. Alokasikan semua data ke pusat klaster terdekat dengan matrik jarak.
4. Hitung kembali pusat klaster baru berdasarkan data yang mengikuti klaster masing-masing.
5. Ulangi langkah 3 dan 4 hingga kondisi konvergen tercapai atau tidak ada data yang berpindah dari satu klaster ke klaster yang lainnya.

2.2 Eksperimen Algoritma K-Means

Pada eksperimen ini algoritma K-Means akan digunakan untuk mengelompokkan data kemiskinan di Jawa Tengah yang diambil dari website Tim Percepatan Penanggulangan Kemiskinan ([TNP2K](#))

2.2.1 Data

Package `readr` menyiapkan fungsi `read_csv()` untuk import data dari file CSV. Pada kasus ini digunakan data [Data 40% Kemiskinan di Jawa Tengah](#).

```
library (readr)
urlfile = "https://raw.githubusercontent.com/dedenistiawan/Dataset/main/Basis%20Data%20Terpadu%20Jateng.csv"

data<-read.csv(url(urlfile), row.names = "Kabupaten")

knitr::kable(
  head(data, 10), caption = 'Basis Data Terpadu Jawa Tengah',
  booktabs = TRUE)
```


Table 2.1: Basis Data Terpadu Jawa Tengah

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
CILACAP	5.19	5.67	5.08	5.44	5.22	6.05	11.47	9.78	5.55	5.12
BANYUMAS	5.71	4.47	5.18	5.51	5.02	6.21	7.39	6.96	5.98	8.22
PURBALINGGA	3.30	2.19	3.80	3.13	3.73	3.34	8.71	7.41	3.21	4.65
BANJARNEGARA	2.73	2.34	3.76	2.80	2.57	2.99	3.31	5.45	4.21	6.05
KEBUMEN	4.17	2.55	3.26	4.16	3.15	4.15	4.30	9.29	4.61	4.34
PURWOREJO	1.87	2.12	1.48	3.05	1.78	1.83	5.00	4.90	3.12	2.09
WONOSOBO	2.13	1.95	3.00	1.78	1.62	2.06	0.45	2.32	3.57	0.84
MAGELANG	3.95	3.01	4.22	4.15	3.01	3.64	1.44	3.35	5.69	3.67
BOYOLALI	2.19	3.07	1.61	2.74	2.11	1.82	1.71	2.34	3.41	1.55
KLATEN	3.84	5.15	1.93	4.64	4.04	3.78	8.71	4.45	3.99	3.09

2.2.2 Memeriksa Missing Value

```
colSums(is.na(data))
```

```
##  X1  X2  X3  X4  X5  X6  X7  X8  X9 X10
##   0   0   0   0   0   0   0   0   0   0
```

Hasil output di atas menunjukkan bahwa tidak *missing value* di semua variabel

2.2.3 Visualisasi Matriks jarak

```
#Plot Disatance
library(ggplot2)
library(factoextra)
distance <- get_dist(data)
fviz_dist(distance, gradient = list(low = "#00AFBB", mid = "white", high = "#FC4E07"))
```

Matriks jarak ini berfungsi untuk mengukur jarak antar variabel, semakin merah warnanya maka semakin jauh jarak antar variabel dan semakin biru semakin dekat jarak antar variabel.

2.2.4 Estimasi Jumlah Cluster Optimal

Dalam metode k-means banyaknya kluster ditentukan sendiri oleh pengguna. Maka dari itu perlu dicari jumlah kluster yang optimum yang dapat mengelompokkan objek dengan baik (Perlu diketahui bahwa metode ini relatif subjektif). Salah satu metode yang digunakan adalah Elbow Plot. Elbow Plot merupakan plot antara banyak kluster dengan total within-cluster variation (total dari simpangan per kluster). Banyak kluster yang dipilih adalah bagian “siku” atau titik dimana terdapat penurunan yang tajam sebelum titik tersebut dan disusul penurunan yang tidak tajam setelah titik tersebut. Hal ini karena penambahan jumlah kluster tidak membawa pengaruh banyak atas variasi yang ada di dalam kluster tersebut.

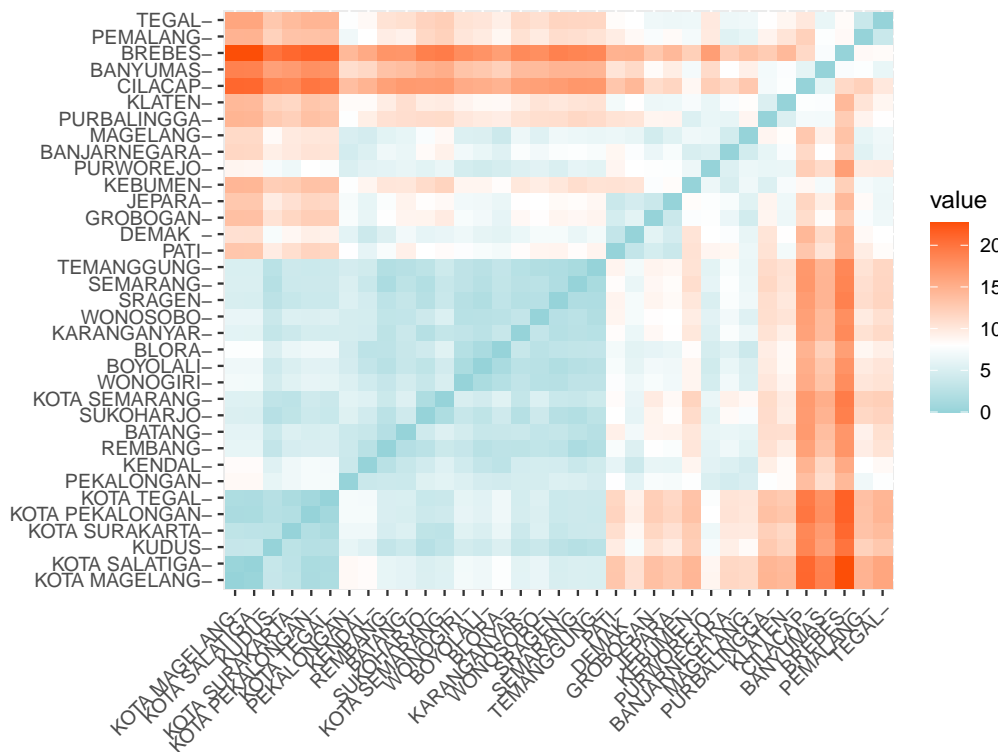


Figure 2.1: Matrik Jarak

2.2.5 Membuat Plot Cluster

Jumlah klaster yang dibentuk mulai dari 2 sampai 5, untuk melihat sebaran data pada masing-masing *cluster*

```
#use several different values of k
```

```
k2 <- kmeans(data, centers = 2, nstart = 25)
k3 <- kmeans(data, centers = 3, nstart = 25)
k4 <- kmeans(data, centers = 4, nstart = 25)
k5 <- kmeans(data, centers = 5, nstart = 25)
```

```
# plots to compare
```

```
p1 <- fviz_cluster(k2, geom = "point", data = data) + ggtitle("k = 2")
p2 <- fviz_cluster(k3, geom = "point", data = data) + ggtitle("k = 3")
p3 <- fviz_cluster(k4, geom = "point", data = data) + ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point", data = data) + ggtitle("k = 5")
```

```
library(gridExtra)
```

```
grid.arrange(p1, p2, p3, p4, nrow = 2)
```

2.2.6 Metode Elbow

Metode Elbow merupakan suatu metode yang digunakan untuk menghasilkan informasi dalam menentukan jumlah cluster terbaik dengan cara melihat persentase hasil perbandingan antara jumlah cluster yang akan membentuk siku pada suatu titik. Metode ini memberikan ide/gagasan dengan cara memilih nilai cluster dan kemudian menambah nilai cluster tersebut untuk dijadikan model data dalam penentuan cluster terbaik. Dan

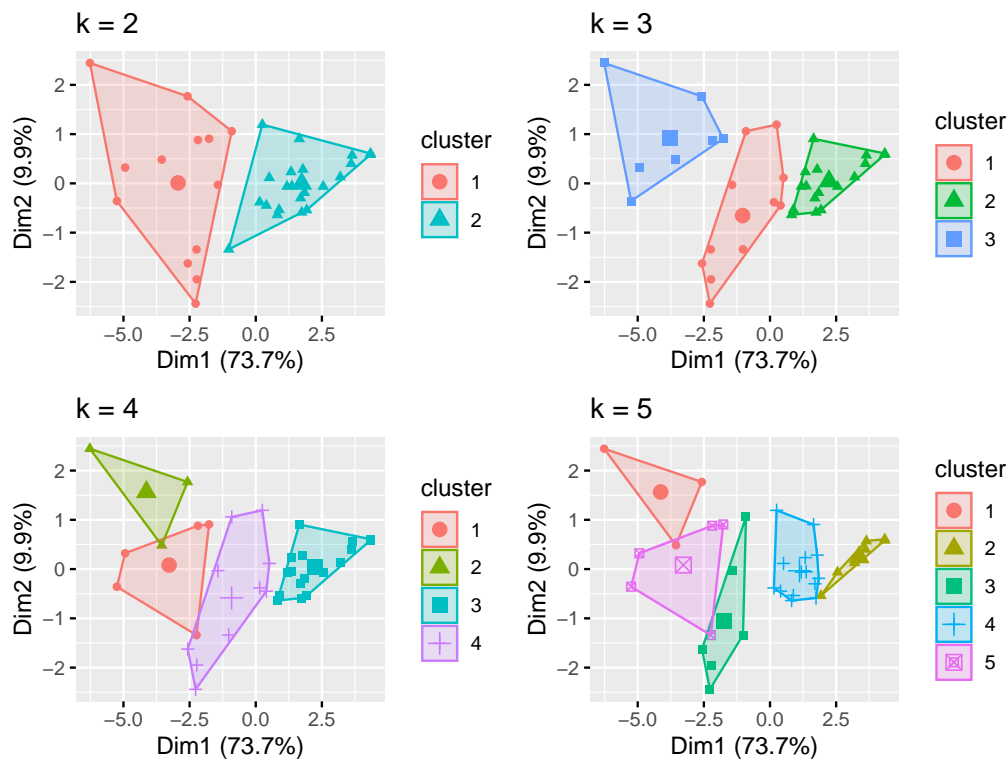


Figure 2.2: Plot Jumlah Cluster

selain itu persentase perhitungan yang dihasilkan menjadi pembandingan antara jumlah cluster yang ditambah. Hasil persentase yang berbeda dari setiap nilai cluster dapat ditunjukkan dengan menggunakan grafik sebagai sumber informasinya. Jika nilai cluster pertama dengan nilai cluster kedua memberikan sudut dalam grafik atau nilainya mengalami penurunan paling besar maka nilai cluster tersebut yang terbaik.

```
#Determining number Optimal Clusters
##Elbow Method
library(ggplot2)
library(factoextra)
fviz_nbclust(data, kmeans, method = "wss") +
  geom_vline(xintercept = 2, linetype = 2)
```

Metode elbow menggunakan nilai total wss (within sum square) sebagai penentu optimalnya. Dari gambar di atas terlihat garis mengalami patahan yang membentuk elbow atau siku pada saat $k = 2$. Maka dengan menggunakan metode ini diperoleh optimal pada saat berada di $k = 2$.

2.2.7 Metode Silhouette

Silhouette Coefficient digunakan untuk melihat kualitas dan kekuatan cluster, seberapa baik suatu objek ditempatkan dalam suatu cluster. Metode ini merupakan gabungan dari metode cohesion dan separation.

```
##Average Silhouette Method
fviz_nbclust(data, kmeans, method = "silhouette")
```

Pendekatan rata-rata nilai metode silhouette untuk menduga kualitas dari klaster yang terbentuk. Semakin tinggi nilai rata-ratanya maka akan semakin baik. Berdasarkan grafik pada gambar di atas banyak klaster optimal yang terbentuk pada $k = 2$.

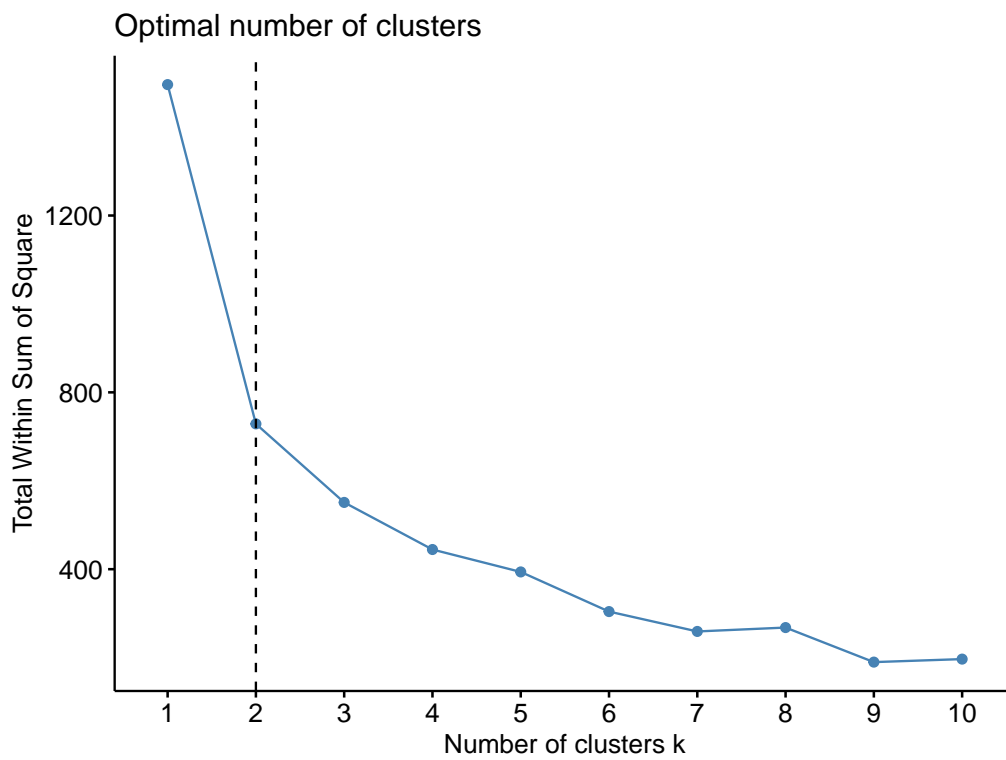


Figure 2.3: Plot Jumlah Cluster Metode Elbow

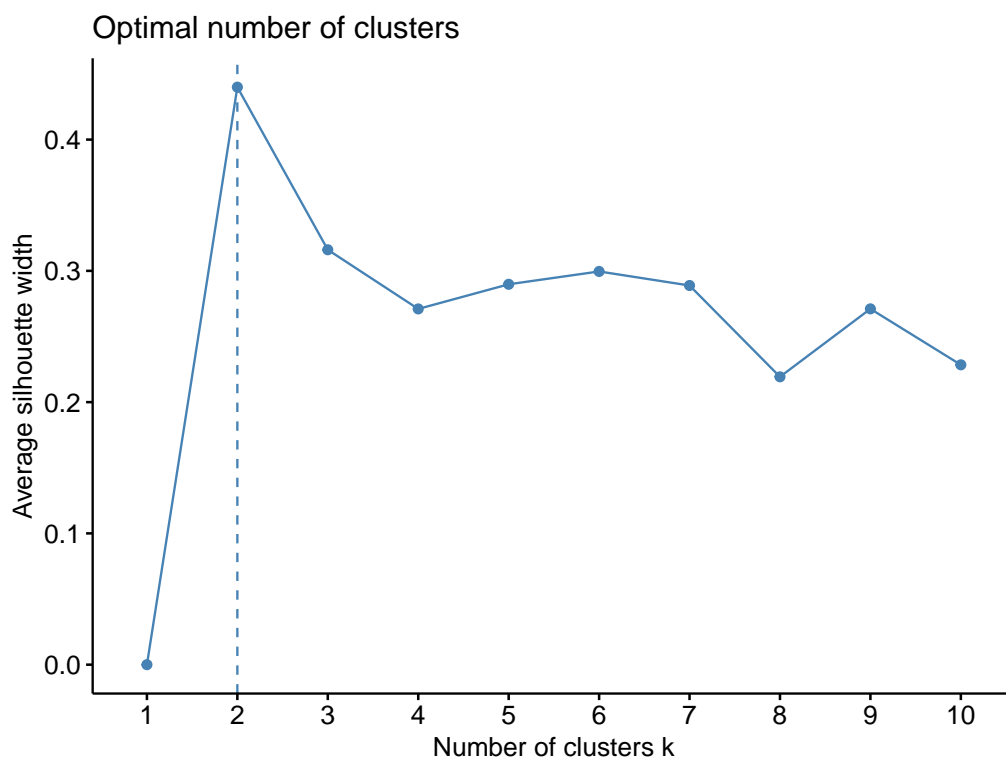


Figure 2.4: Plot Jumlah Cluster Metode silhouette

2.2.8 Eksperimen K-Means Clustering

Dari pendekatan metode elbow dan metode Silhouette di dapatkan jumlah *cluster* optimal adalah K=2. setelah ini dilakukan eksperimen jumlah K=2

```
#Computing k-means clustering
#Compute k-means with k = 2
set.seed(123)
km.res <- kmeans(data, 2, nstart = 25)
# Print the results
print(km.res)
```

```
## K-means clustering with 2 clusters of sizes 22, 13
##
## Cluster means:
##      X1      X2      X3      X4      X5      X6      X7
## 1 1.918182 2.017273 1.675000 1.949091 1.890455 1.728182 1.557273
## 2 4.446923 4.276923 4.854615 4.394615 4.494615 4.769231 5.056154
##      X8      X9      X10
## 1 1.286818 1.952727 1.455455
## 2 5.515385 4.389231 5.230000
##
## Clustering vector:
##      CILACAP      BANYUMAS      PURBALINGGA      BANJARNEGARA
##      2          2          2          2
##      KEBUMEN      PURWOREJO      WONOSOBO      MAGELANG
##      2          1          1          2
##      BOYOLALI      KLATEN      SUKOHARJO      WONOGIRI
##      1          2          1          1
##      KARANGANYAR      SRAGEN      GROBOGAN      BLORA
##      1          1          2          1
##      REMBANG      PATI      KUDUS      JEPARA
##      1          2          1          2
##      DEMAK      SEMARANG      TEMANGGUNG      KENDAL
##      1          1          1          1
##      BATANG      PEKALONGAN      PEMALANG      TEGAL
##      1          1          2          2
##      BREBES      KOTA MAGELANG      KOTA SURAKARTA      KOTA SALATIGA
##      2          1          1          1
##      KOTA SEMARANG      KOTA PEKALONGAN      KOTA TEGAL
##      1          1          1
##
## Within cluster sum of squares by cluster:
## [1] 262.6536 466.0163
## (between_SS / total_SS =  51.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"      "withinss"
```

```
## [5] "tot.withinss" "betweenss" "size" "iter"
## [9] "ifault"
```

Melihat hasil *cluster* akhir pada setiap kabupaten

```
# Cluster number for each of the observations
km.res$cluster
```

```
##          CILACAP          BANYUMAS      PURBALINGGA      BANJARNEGARA
##              2              2              2              2
##          KEBUMEN          PURWOREJO          WONOSOBO          MAGELANG
##              2              1              1              2
##          BOYOLALI          KLATEN          SUKOHARJO          WONOGIRI
##              1              2              1              1
##          KARANGANYAR          SRAGEN          GROBOGAN          BLORA
##              1              1              2              1
##          REMBANG          PATI              KUDUS          JEPARA
##              1              2              1              2
##          DEMAK          SEMARANG          TEMANGGUNG          KENDAL
##              1              1              1              1
##          BATANG          PEKALONGAN          PEMALANG          TEGAL
##              1              1              2              2
##          BREBES      KOTA MAGELANG      KOTA SURAKARTA      KOTA SALATIGA
##              2              1              1              1
##          KOTA SEMARANG      KOTA PEKALONGAN          KOTA TEGAL
##              1              1              1
```

Melihat jumlah anggota *cluster*

```
# Cluster size
km.res$size
```

```
## [1] 22 13
```

2.2.9 Visualisasi Hasil *clustering*

```
# Cluster means
km.res$centers
```

```
##          X1          X2          X3          X4          X5          X6          X7
## 1 1.918182 2.017273 1.675000 1.949091 1.890455 1.728182 1.557273
## 2 4.446923 4.276923 4.854615 4.394615 4.494615 4.769231 5.056154
##          X8          X9          X10
## 1 1.286818 1.952727 1.455455
## 2 5.515385 4.389231 5.230000
```

```
fviz_cluster(km.res, data = data)
```

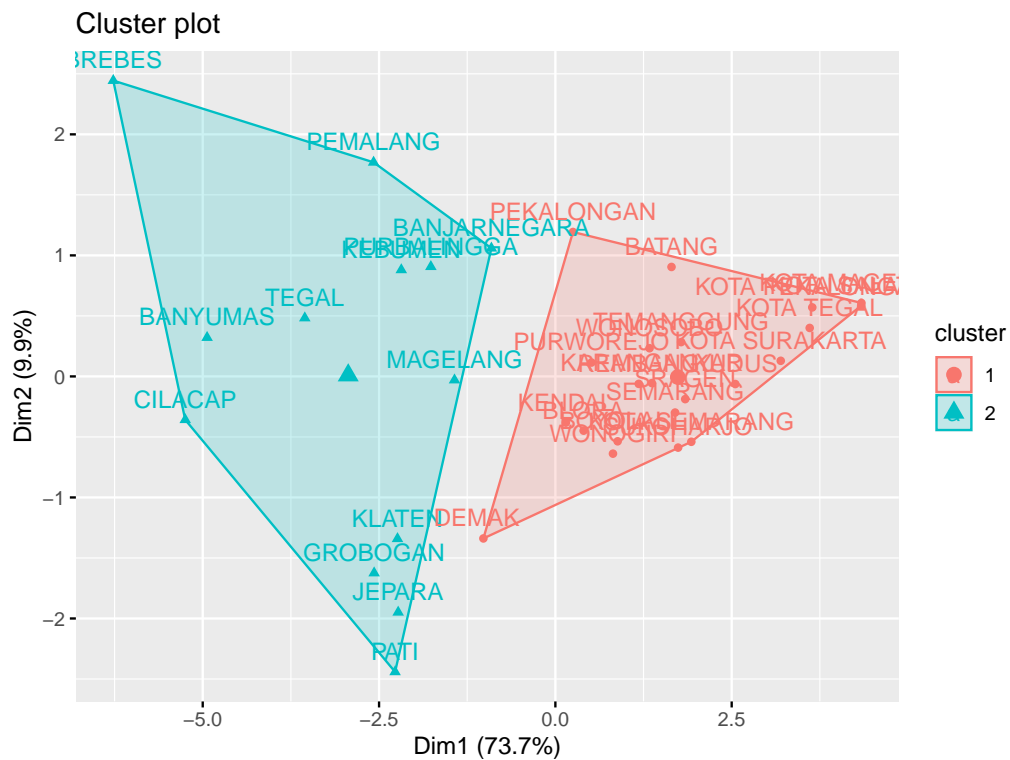


Figure 2.5: Plot Hasil Cluster

```
fviz_cluster(km.res, data = data,
  palette = c("#FC4E07", "#00AFBB"),
  ellipse.type = "euclid", # Concentration ellipse
  star.plot = TRUE, # Add segments from centroids to items
  repel = TRUE, # Avoid label overplotting (slow)
  ggtheme = theme_minimal())
```

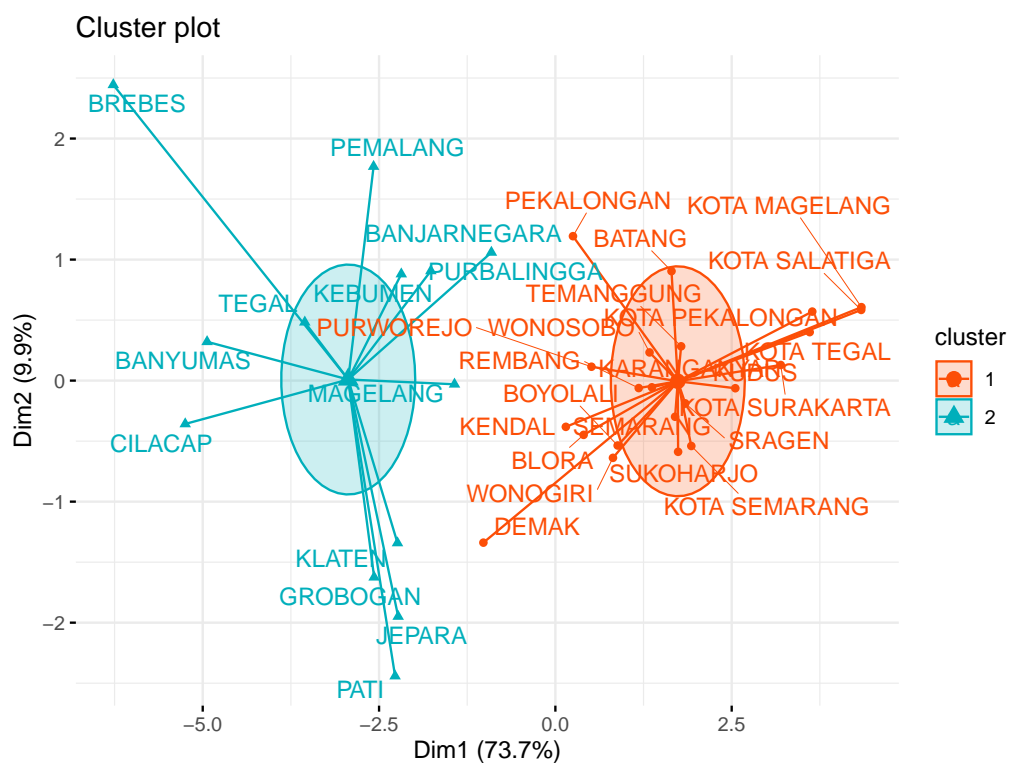


Figure 2.6: Plot Hasil Cluster

Chapter 3 Algoritma Fuzzy C-Means

3.1 Pengantar Algoritma Fuzzy C-Means

Fuzzy c-means merupakan metode yang dikenal baik dalam mendeteksi kluster (Pimentel and Souza 2016). Metode ini menggunakan model pengelompokan fuzzy sehingga data dapat menjadi anggota dari semua kelas atau kluster terbentuk dengan derajat atau tingkat keanggotaan yang berbeda antara 0 hingga 1. Tingkat keberadaan data dalam suatu kelas atau kluster ditentukan oleh derajat keanggotaannya. Kelebihan dari metode ini adalah penempatan pusat kluster yang lebih tepat dibandingkan dengan metode lain. Caranya adalah dengan memperbaiki pusat kluster secara berulang, maka akan dapat dilihat bahwa pusat kluster akan bergerak menuju lokasi yang tepat (Wijaya, 2014). Namun, pada algoritma Fuzzy c-means dibutuhkan waktu komputasi yang lama (Stetco, Zeng, and Keane 2015).

Klustering dengan algoritma Fuzzy C-Means didasarkan pada teori logika fuzzy yang diperkenalkan oleh Lotfi Zadeh pada tahun 1965 dengan nama himpunan fuzzy (fuzzy set). Fuzzy C-Means Clustering pertama kali diperkenalkan oleh Dun pada (1973) dan diperbaiki oleh Bezdek (Bezdek, Ehrlich, and Full 1984). Dalam teori fuzzy, keanggotaan sebuah data diberikan dengan suatu nilai derajat keanggotaan yang jangkauan nilainya 0 sampai 1. Semakin tinggi nilai derajat keanggotaannya maka semakin tinggi nilai keanggotaan sebuah data dalam suatu kelompok dan semakin kecil nilai derajat keanggotaannya maka semakin rendah nilai keanggotaan sebuah data dalam suatu kelompok.

Asumsikan terdapat sejumlah data dalam dataset X yang berisi n data yang dinotasikan $X = x_1, x_2, \dots, x_n$, dimana setiap data mempunyai fitur r dimensi: $x_{i1}, x_{i2}, \dots, x_{ir}$, dinotasikan $x_i = x_{i1}, x_{i2}, \dots, x_{ir}$. Ada sejumlah kluster C dengan centroid: C_1, C_2, \dots, C_k , dimana k adalah jumlah kluster. Setiap data mempunyai derajat keanggotaan pada setiap kluster, dinyatakan dengan u_{ij} , dengan nilai diantara 0 dan 1, i menyatakan data x_i dan j menyatakan kluster c_j . Jumlah nilai derajat keanggotaan setiap data x_i selalu sama dengan 1, yang diformulasikan pada persamaan berikut:

Theorem 3.1

$$\sum_{j=1}^k u_{ij} = 1$$



Fuzzy c-means clustering merupakan suatu metode clustering yang hampir mirip seperti k-means clustering. Karena metode clustering ini mirip dengan k-means clustering, ada yang menyebut metode ini fuzzy k-means clustering. Fuzzy c-means merupakan salah satu jenis soft clustering dimana dalam mengelompokkan suatu data, setiap data bisa dimiliki lebih dari satu cluster.

Cara kerja dari fuzzy c-means clustering dalam mengelompokkan datanya adalah sebagai berikut :

1. Menentukan banyak cluster (k) yang akan dibuat.
2. Menentukan nilai proporsi untuk setiap data poin secara random untuk masuk dalam suatu cluster. Menghitung nilai centroid.
3. Dalam menghitung nilai centroid, kita menggunakan formula berikut:

Lemma 3.1

$$C_j = \frac{\sum \mu_{ij}^m x}{\sum \mu_{ij}^m}$$



4. Menghitung kembali nilai proporsi untuk setiap data poin untuk masuk pada setiap cluster. formula yang digunakan yaitu sebagai berikut:

Lemma 3.2

$$\mu_{ij}^m = \frac{1}{\sum \left(\frac{|x_i - c_j|}{|x_i - c_k|} \right)^{\frac{2}{m-1}}}$$



3.2 Eksperimen Fuzzy C-Means

3.2.1 Install dan Load Packages

```
library(ppclust)
library(factoextra)
library(fclust)
library(cluster)
```

3.2.2 Data

```
library(readr)
urlfile = "https://raw.githubusercontent.com/dedenistiawan/Dataset/main/Basis%20Data%20Terpadu%20Jateng.cs"
data<-read.csv(url(urlfile), row.names = "Kabupaten")
```

3.2.3 Hasil Clustering

```
library(ppclust)
res.fcm <- fcm(data, centers=3)
as.data.frame(res.fcm$u)
```

```
##           Cluster 1 Cluster 2 Cluster 3
## CILACAP      0.70608174 0.09432155 0.19959671
## BANYUMAS     0.87685700 0.03242738 0.09071562
## PURBALINGGA  0.58855259 0.12503329 0.28641412
## BANJARNEGARA 0.27431153 0.15770912 0.56797934
## KEBUMEN      0.55326811 0.12183017 0.32490172
## PURWOREJO    0.15508384 0.37665821 0.46825795
## WONOSOBO     0.04794255 0.73997169 0.21208577
## MAGELANG     0.11864932 0.12565978 0.75569090
## BOYOLALI     0.04746322 0.65917928 0.29335751
## KLATEN       0.40625017 0.15366683 0.44008300
## SUKOHARJO    0.01812578 0.90659203 0.07528219
## WONOGIRI     0.05984340 0.62649398 0.31366262
## KARANGANYAR  0.03151333 0.83280963 0.13567704
## SRAGEN       0.01369224 0.92987167 0.05643609
## GROBOGAN     0.20509357 0.10906165 0.68584478
```

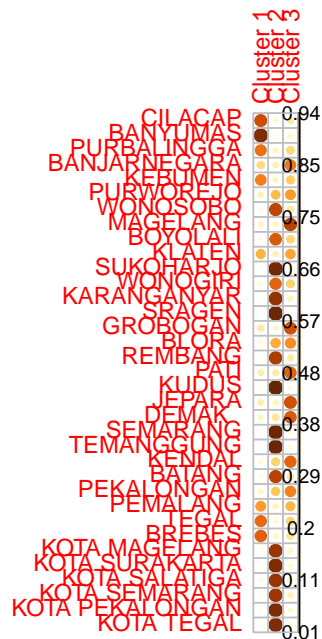
```
## BLORA 0.06584083 0.42364979 0.51050938
## REMBANG 0.03515460 0.77496097 0.18988443
## PATI 0.18018760 0.18226626 0.63754615
## KUDUS 0.01358093 0.93999864 0.04642043
## JEPARA 0.17434136 0.13357477 0.69208387
## DEMAK 0.12077163 0.22405973 0.65516864
## SEMARANG 0.01375558 0.92443899 0.06180543
## TEMANGGUNG 0.01515432 0.92181846 0.06302723
## KENDAL 0.07103512 0.30783534 0.62112953
## BATANG 0.05146780 0.74568211 0.20285008
## PEKALONGAN 0.11753007 0.33879435 0.54367557
## PEMALANG 0.47108349 0.13508254 0.39383397
## TEGAL 0.61445746 0.08391671 0.30162583
## BREBES 0.64100931 0.11958050 0.23941019
## KOTA MAGELANG 0.05127744 0.82227544 0.12644712
## KOTA SURAKARTA 0.03051218 0.88284897 0.08663885
## KOTA SALATIGA 0.05093432 0.82356236 0.12550332
## KOTA SEMARANG 0.04152177 0.81177689 0.14670134
## KOTA PEKALONGAN 0.03226874 0.87988142 0.08784985
## KOTA TEGAL 0.03188749 0.88100104 0.08711147
```

```
# Visualize using corrplot
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(res.fcm$u, is.corr = FALSE)
```



```
res.fcm$u0
```

```
## X1 X2 X3 X4 X5 X6 X7 X8 X9 X10
```

```
## Cluster 1 4.78 3.91 6.70 4.84 6.35 5.82 3.00 5.99 2.98 5.98
## Cluster 2 2.19 3.07 1.61 2.74 2.11 1.82 1.71 2.34 3.41 1.55
## Cluster 3 3.84 5.15 1.93 4.64 4.04 3.78 8.71 4.45 3.99 3.09
```

```
res.fcm$v
```

```
##           X1          X2          X3          X4          X5          X6
## Cluster 1 5.001721 4.041093 5.531054 4.669414 4.638562 5.500335
## Cluster 2 1.724771 1.743303 1.432408 1.706002 1.626688 1.504833
## Cluster 3 3.418238 4.025821 3.497306 3.657048 3.843888 3.477061
##           X7          X8          X9          X10
## Cluster 1 6.458007 7.009924 4.458629 6.474407
## Cluster 2 1.335312 1.021470 1.697120 1.056498
## Cluster 3 3.074419 3.057133 3.802309 3.529256
```

3.2.4 Hasil Clustering FCM

```
summary(res.fcm)
```

```
## Summary for 'res.fcm'
##
## Number of data objects: 35
##
## Number of clusters: 3
##
## Crisp clustering vector:
## [1] 1 1 1 3 1 3 2 3 2 3 2 2 2 2 3 3 2 3 2 3 3 2 2 3 2 3 1 1 1 2 2 2
## [33] 2 2 2
##
## Initial cluster prototypes:
##           X1  X2  X3  X4  X5  X6  X7  X8  X9  X10
## Cluster 1 4.78 3.91 6.70 4.84 6.35 5.82 3.00 5.99 2.98 5.98
## Cluster 2 2.19 3.07 1.61 2.74 2.11 1.82 1.71 2.34 3.41 1.55
## Cluster 3 3.84 5.15 1.93 4.64 4.04 3.78 8.71 4.45 3.99 3.09
##
## Final cluster prototypes:
##           X1          X2          X3          X4          X5          X6
## Cluster 1 5.001721 4.041093 5.531054 4.669414 4.638562 5.500335
## Cluster 2 1.724771 1.743303 1.432408 1.706002 1.626688 1.504833
## Cluster 3 3.418238 4.025821 3.497306 3.657048 3.843888 3.477061
##           X7          X8          X9          X10
## Cluster 1 6.458007 7.009924 4.458629 6.474407
## Cluster 2 1.335312 1.021470 1.697120 1.056498
## Cluster 3 3.074419 3.057133 3.802309 3.529256
##
## Distance between the final cluster prototypes
##           Cluster 1 Cluster 2
```

```

## Cluster 2 165.71763
## Cluster 3 48.57171 42.66853
##
## Difference between the initial and final cluster prototypes
##           X1           X2           X3           X4           X5
## Cluster 1 0.2217214 0.1310934 -1.1689460 -0.1705855 -1.7114383
## Cluster 2 -0.4652294 -1.3266973 -0.1775916 -1.0339977 -0.4833119
## Cluster 3 -0.4217618 -1.1241790 1.5673058 -0.9829516 -0.1961115
##           X6           X7           X8           X9           X10
## Cluster 1 -0.3196647 3.458007 1.019924 1.4786289 0.4944068
## Cluster 2 -0.3151670 -0.374688 -1.318530 -1.7128795 -0.4935021
## Cluster 3 -0.3029386 -5.635581 -1.392867 -0.1876907 0.4392557
##
## Root Mean Squared Deviations (RMSD): 4.736774
## Mean Absolute Deviation (MAD): 100.4222
##
## Membership degrees matrix (top and bottom 5 rows):
##           Cluster 1 Cluster 2 Cluster 3
## CILACAP      0.7060817 0.09432155 0.19959671
## BANYUMAS      0.8768570 0.03242738 0.09071562
## PURBALINGGA   0.5885526 0.12503329 0.28641412
## BANJARNEGARA 0.2743115 0.15770912 0.56797934
## KEBUMEN       0.5532681 0.12183017 0.32490172
## ...
##           Cluster 1 Cluster 2 Cluster 3
## KOTA SURAKARTA 0.03051218 0.8828490 0.08663885
## KOTA SALATIGA  0.05093432 0.8235624 0.12550332
## KOTA SEMARANG  0.04152177 0.8117769 0.14670134
## KOTA PEKALONGAN 0.03226873 0.8798814 0.08784985
## KOTA TEGAL     0.03188749 0.8810010 0.08711147
##
## Descriptive statistics for the membership degrees by clusters
##           Size      Min      Q1      Mean      Median      Q3
## Cluster 1      7 0.4710835 0.5709104 0.6359014 0.6144575 0.6735455
## Cluster 2     17 0.6264940 0.7749610 0.8295979 0.8328096 0.9065920
## Cluster 3     11 0.4400830 0.5270925 0.5979972 0.6211295 0.6705067
##           Max
## Cluster 1 0.8768570
## Cluster 2 0.9399986
## Cluster 3 0.7556909
##
## Dunn's Fuzziness Coefficients:
## dunn_coeff normalized
## 0.5999684 0.3999525
##
## Within cluster sum of squares by cluster:
##           1           2           3
## 220.3251 130.5953 200.0818

```

```
## (between_SS / total_SS = 61.98%)
##
## Available components:
## [1] "u"          "v"          "v0"         "d"          "x"
## [6] "cluster"    "csize"      "sumsqrs"    "k"          "m"
## [11] "iter"       "best.start" "func.val"    "comp.time"  "inpargs"
## [16] "algorithm"  "call"
```

3.2.5 Run FCM with Multiple Starts

```
res.fcm <- fcm(data, centers=3, nstart=5)
```

```
res.fcm <- fcm(data, centers=3, nstart=5, fixmemb=TRUE)
```

3.2.6 Display the best solution

```
res.fcm$func.val
```

```
## [1] 360.931 360.931 360.931 360.931 360.931
```

```
res.fcm$iter
```

```
## [1] 82 90 68 77 69
```

```
res.fcm$best.start
```

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

3.2.7 Display the summary of clustering results

```
summary(res.fcm)
```

```
## Summary for 'res.fcm'
##
## Number of data objects: 35
##
## Number of clusters: 3
##
## Crisp clustering vector:
## [1] 2 2 2 1 2 1 3 1 3 1 3 3 3 1 1 3 1 3 1 1 3 3 1 3 1 2 2 2 3 3 3
## [33] 3 3 3
##
## Initial cluster prototypes:
##          X1  X2  X3  X4  X5  X6  X7  X8  X9  X10
```

```

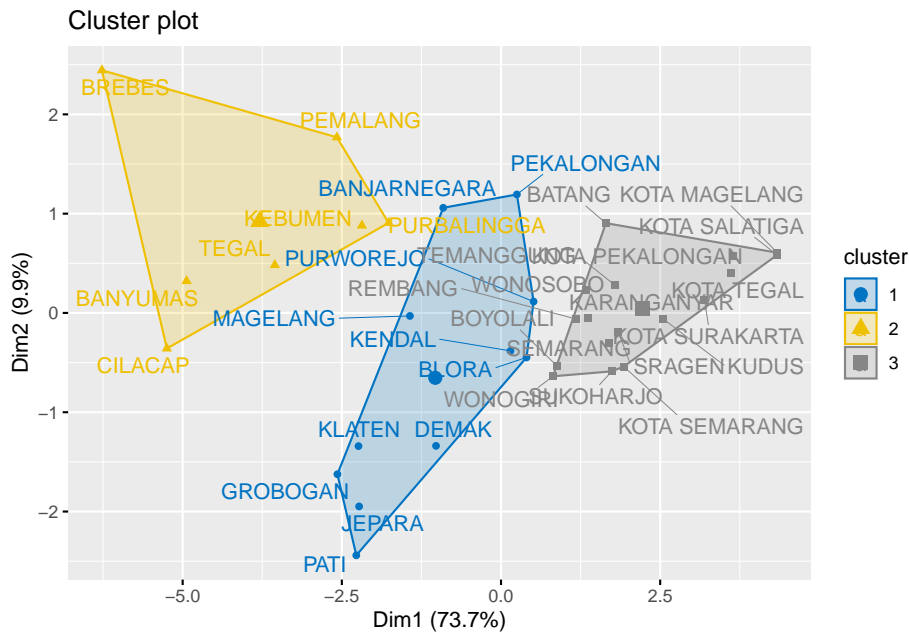
## Cluster 1 3.84 5.15 1.93 4.64 4.04 3.78 8.71 4.45 3.99 3.09
## Cluster 2 1.87 2.12 1.48 3.05 1.78 1.83 5.00 4.90 3.12 2.09
## Cluster 3 5.71 4.47 5.18 5.51 5.02 6.21 7.39 6.96 5.98 8.22
##
## Final cluster prototypes:
##           X1      X2      X3      X4      X5      X6
## Cluster 1 3.418238 4.025821 3.497306 3.657048 3.843888 3.477061
## Cluster 2 5.001721 4.041093 5.531054 4.669414 4.638562 5.500335
## Cluster 3 1.724771 1.743303 1.432408 1.706002 1.626688 1.504833
##           X7      X8      X9      X10
## Cluster 1 3.074419 3.057133 3.802309 3.529256
## Cluster 2 6.458007 7.009924 4.458629 6.474407
## Cluster 3 1.335312 1.021470 1.697120 1.056498
##
## Distance between the final cluster prototypes
##           Cluster 1 Cluster 2
## Cluster 2  48.57171
## Cluster 3  42.66853 165.71763
##
## Difference between the initial and final cluster prototypes
##           X1      X2      X3      X4      X5
## Cluster 1 -0.4217618 -1.124179  1.567306 -0.9829516 -0.1961115
## Cluster 2  3.1317214  1.921093  4.051054  1.6194145  2.8585617
## Cluster 3 -3.9852294 -2.726697 -3.747592 -3.8039977 -3.3933119
##           X6      X7      X8      X9      X10
## Cluster 1 -0.3029386 -5.635581 -1.392867 -0.1876907  0.4392557
## Cluster 2  3.6703353  1.458007  2.109924  1.3386289  4.3844068
## Cluster 3 -4.7051670 -6.054688 -5.938530 -4.2828795 -7.1635021
##
## Root Mean Squared Deviations (RMSD): 10.7666
## Mean Absolute Deviation (MAD): 281.9846
##
## Membership degrees matrix (top and bottom 5 rows):
##           Cluster 1 Cluster 2 Cluster 3
## CILACAP      0.19959671 0.7060817 0.09432155
## BANYUMAS      0.09071562 0.8768570 0.03242738
## PURBALINGGA   0.28641412 0.5885526 0.12503329
## BANJARNEGARA  0.56797934 0.2743115 0.15770912
## KEBUMEN       0.32490172 0.5532681 0.12183017
## ...
##           Cluster 1 Cluster 2 Cluster 3
## KOTA SURAKARTA 0.08663885 0.03051218 0.8828490
## KOTA SALATIGA  0.12550332 0.05093432 0.8235624
## KOTA SEMARANG  0.14670134 0.04152177 0.8117769
## KOTA PEKALONGAN 0.08784985 0.03226873 0.8798814
## KOTA TEGAL     0.08711147 0.03188749 0.8810010
##
## Descriptive statistics for the membership degrees by clusters

```

```
##           Size      Min      Q1      Mean      Median      Q3
## Cluster 1    11 0.4400830 0.5270925 0.5979972 0.6211295 0.6705067
## Cluster 2     7 0.4710835 0.5709104 0.6359014 0.6144575 0.6735455
## Cluster 3    17 0.6264940 0.7749610 0.8295979 0.8328096 0.9065920
##
##           Max
## Cluster 1 0.7556909
## Cluster 2 0.8768570
## Cluster 3 0.9399986
##
## Dunn's Fuzziness Coefficients:
## dunn_coeff normalized
## 0.5999684 0.3999525
##
## Within cluster sum of squares by cluster:
##      1      2      3
## 200.0818 220.3251 130.5953
## (between_SS / total_SS = 61.98%)
##
## Available components:
## [1] "u"      "v"      "v0"     "d"      "x"
## [6] "cluster" "csize"  "sumsqrs" "k"      "m"
## [11] "iter"    "best.start" "func.val" "comp.time" "inpargs"
## [16] "algorithm" "call"
```

3.2.8 Cluster Plot with fviz_cluster

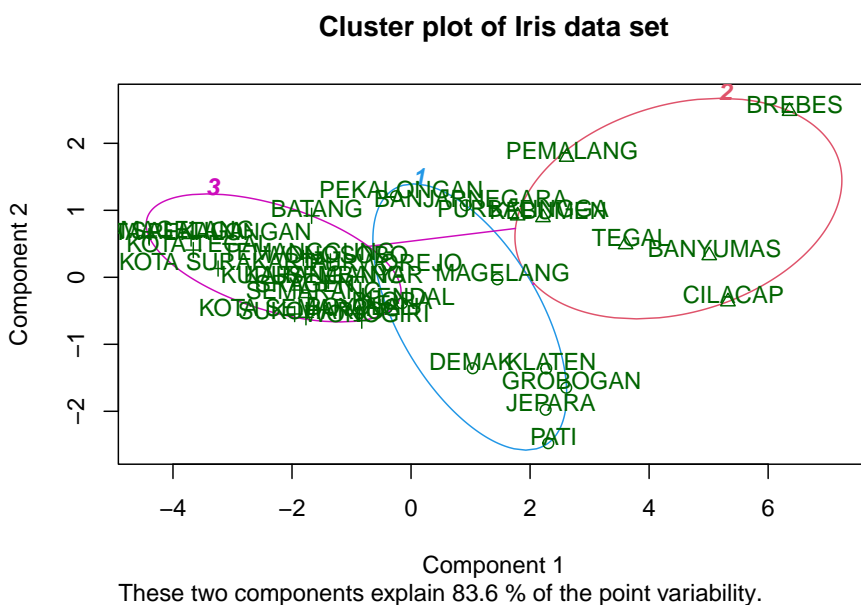
```
res.fcm2 <- ppclust2(res.fcm, "kmeans")
factoextra::fviz_cluster(res.fcm2, data = data,
  ellipse.type = "convex",
  palette = "jco",
  repel = TRUE)
```

3.2.9 Cluster Plot with clusplot

```
res.fcm3 <- ppclust2(res.fcm, "fanny")

cluster::clusplot(scale(data), res.fcm3$cluster,
  main = "Cluster plot of Iris data set",
  color=TRUE, labels = 2, lines = 2, cex=1)
```



Chapter 4 Metode Cluster Hirarki

4.1 Data

```
library (readr)
urlfile = "https://raw.githubusercontent.com/dedenistiawan/Dataset/main/Basis%20Data%20Terpadu%20Jateng.cs

data<-read.csv(url(urlfile), row.names = "Kabupaten")

knitr::kable(
  head(data, 10), caption = 'Basis Data Terpadu Jawa Tengah',
  booktabs = TRUE)
```

Table 4.1: Basis Data Terpadu Jawa Tengah

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
CILACAP	5.19	5.67	5.08	5.44	5.22	6.05	11.47	9.78	5.55	5.12
BANYUMAS	5.71	4.47	5.18	5.51	5.02	6.21	7.39	6.96	5.98	8.22
PURBALINGGA	3.30	2.19	3.80	3.13	3.73	3.34	8.71	7.41	3.21	4.65
BANJARNEGARA	2.73	2.34	3.76	2.80	2.57	2.99	3.31	5.45	4.21	6.05
KEBUMEN	4.17	2.55	3.26	4.16	3.15	4.15	4.30	9.29	4.61	4.34
PURWOREJO	1.87	2.12	1.48	3.05	1.78	1.83	5.00	4.90	3.12	2.09
WONOSOBO	2.13	1.95	3.00	1.78	1.62	2.06	0.45	2.32	3.57	0.84
MAGELANG	3.95	3.01	4.22	4.15	3.01	3.64	1.44	3.35	5.69	3.67
BOYOLALI	2.19	3.07	1.61	2.74	2.11	1.82	1.71	2.34	3.41	1.55
KLATEN	3.84	5.15	1.93	4.64	4.04	3.78	8.71	4.45	3.99	3.09

```
# Standardize the data
df <- scale(data)

# Compute the dissimilarity matrix
# df = the standardized data
res.dist <- dist(df, method = "euclidean")

as.matrix(res.dist)[1:5, 1:5]
```

```
##           CILACAP BANYUMAS PURBALINGGA BANJARNEGARA  KEBUMEN
## CILACAP      0.000000  2.327193      3.828424      5.188508  3.891360
## BANYUMAS      2.327193  0.000000      3.809719      4.232529  3.310710
## PURBALINGGA   3.828424  3.809719      0.000000      2.418211  2.235801
## BANJARNEGARA  5.188508  4.232529      2.418211      0.000000  2.159694
## KEBUMEN       3.891360  3.310710      2.235801      2.159694  0.000000
```

```
res.hc <- hclust(d = res.dist, method = "ward.D2")
```

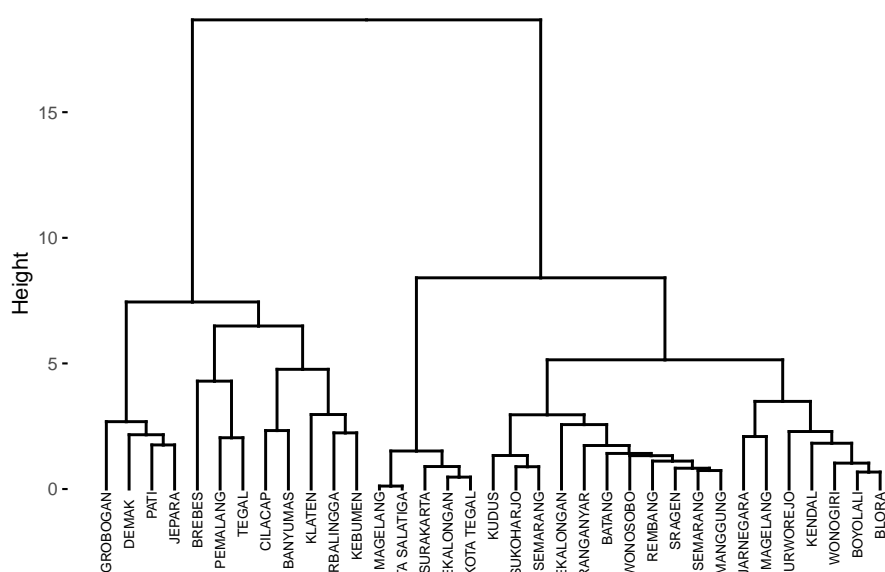
```
# cex: label size
library("factoextra")
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggplot2)
fviz_dend(res.hc, cex = 0.5)
```

Cluster Dendrogram



```
# Cut tree into 2 groups
grp <- cutree(res.hc, k = 2)
head(grp, n = 2)
```

```
## CILACAP BANYUMAS
##      1      1
```

```
# Number of members in each cluster
table(grp)
```

```
## grp
##  1  2
## 12 23
```

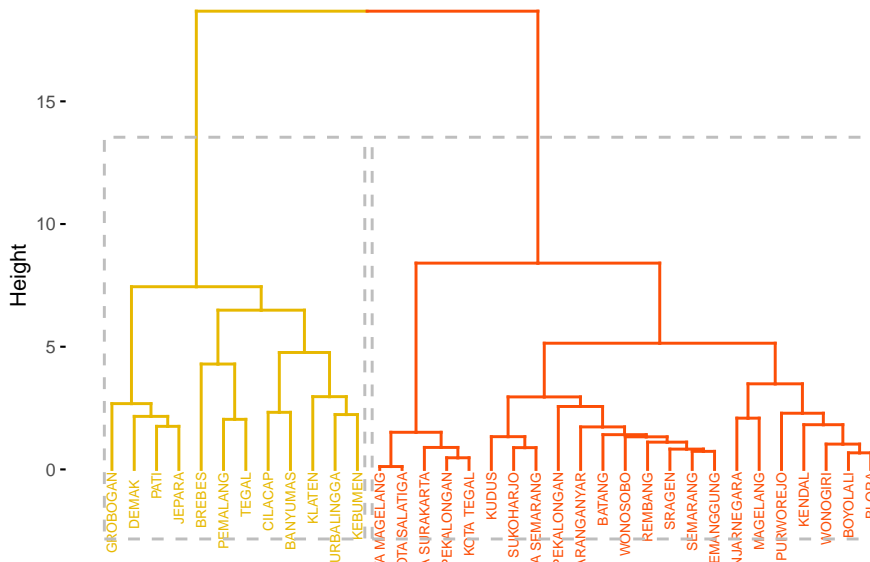
```
# Cut in 2 groups and color by groups
fviz_dend(res.hc, k = 2, # Cut in four groups
          cex = 0.5, # label size
```

```

k_colors = c("#E7B800", "#FC4E07"),
color_labels_by_k = TRUE, # color labels by groups
rect = TRUE # Add rectangle around groups
)

```

Cluster Dendrogram

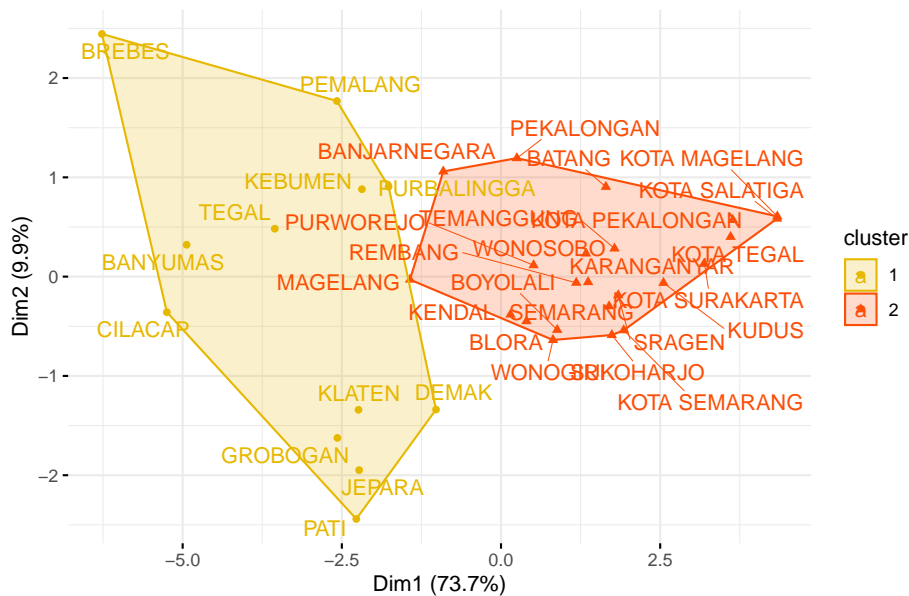


```

fviz_cluster(list(data = df, cluster = grp),
palette = c("#E7B800", "#FC4E07"),
ellipse.type = "convex", # Concentration ellipse
repel = TRUE, # Avoid label overplotting (slow)
show.clust.cent = FALSE, ggtheme = theme_minimal())

```

Cluster plot



4.2 Comparing dendrograms

```
library(dendextend)

##
## -----
## Welcome to dendextend version 1.17.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----

##
## Attaching package: 'dendextend'

## The following object is masked from 'package:stats':
##
##   cutree

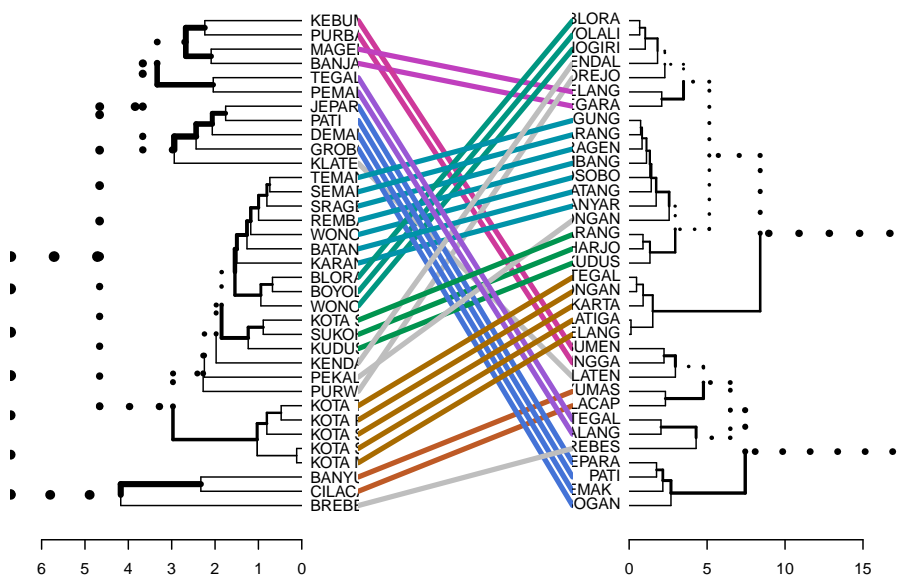
# Compute distance matrix
res.dist <- dist(df, method = "euclidean")

# Compute 2 hierarchical clusterings
hc1 <- hclust(res.dist, method = "average")
hc2 <- hclust(res.dist, method = "ward.D2")

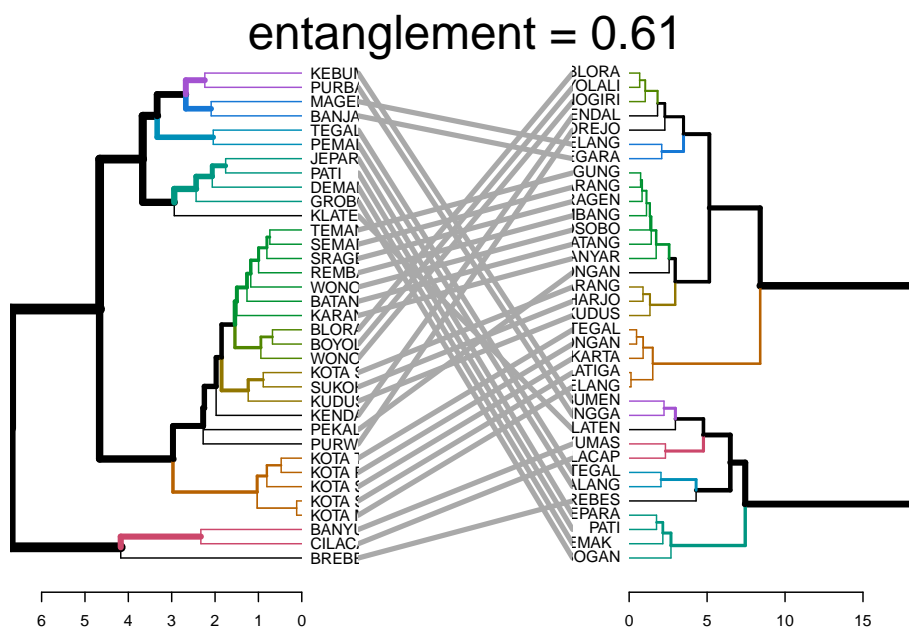
# Create two dendrograms
dend1 <- as.dendrogram (hc1)
dend2 <- as.dendrogram (hc2)

# Create a list to hold dendrograms
dend_list <- dendlist(dend1, dend2)

tanglegram(dend1, dend2)
```



```
tanglegram(dend1, dend2,
highlight_distinct_edges = FALSE, # Turn-off dashed lines
common_subtrees_color_lines = FALSE, # Turn-off line colors
common_subtrees_color_branches = TRUE, # Color common branches
main = paste("entanglement =", round(entanglement(dend_list), 2))
)
```



```
# Create multiple dendrograms by chaining
dend1 <- df %>% dist %>% hclust("complete") %>% as.dendrogram
dend2 <- df %>% dist %>% hclust("single") %>% as.dendrogram
dend3 <- df %>% dist %>% hclust("average") %>% as.dendrogram
dend4 <- df %>% dist %>% hclust("centroid") %>% as.dendrogram
# Compute correlation matrix
```

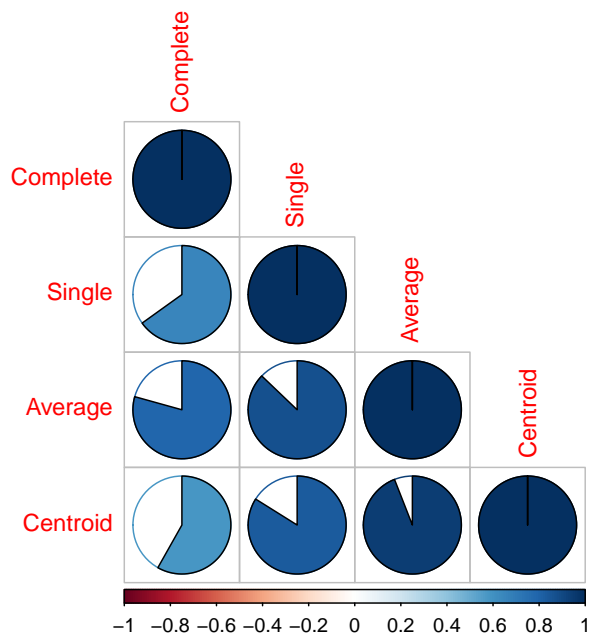
```
dend_list <- dendlist("Complete" = dend1, "Single" = dend2,
  "Average" = dend3, "Centroid" = dend4)
cors <- cor.dendlist(dend_list)
# Print correlation matrix
round(cors, 2)
```

```
##           Complete Single Average Centroid
## Complete    1.00    0.65    0.79    0.58
## Single      0.65    1.00    0.87    0.84
## Average     0.79    0.87    1.00    0.94
## Centroid    0.58    0.84    0.94    1.00
```

```
# Visualize the correlation matrix using corrplot package
library(corrplot)
```

```
## corrplot 0.92 loaded
```

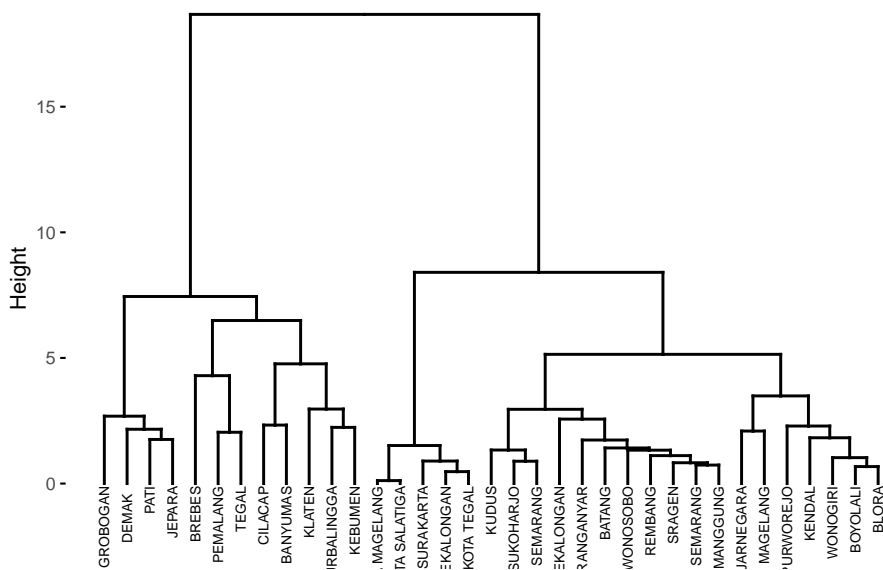
```
corrplot(cors, "pie", "lower")
```



```
# Compute distances and hierarchical clustering
dd <- dist(scale(data), method = "euclidean")
hc <- hclust(dd, method = "ward.D2")
```

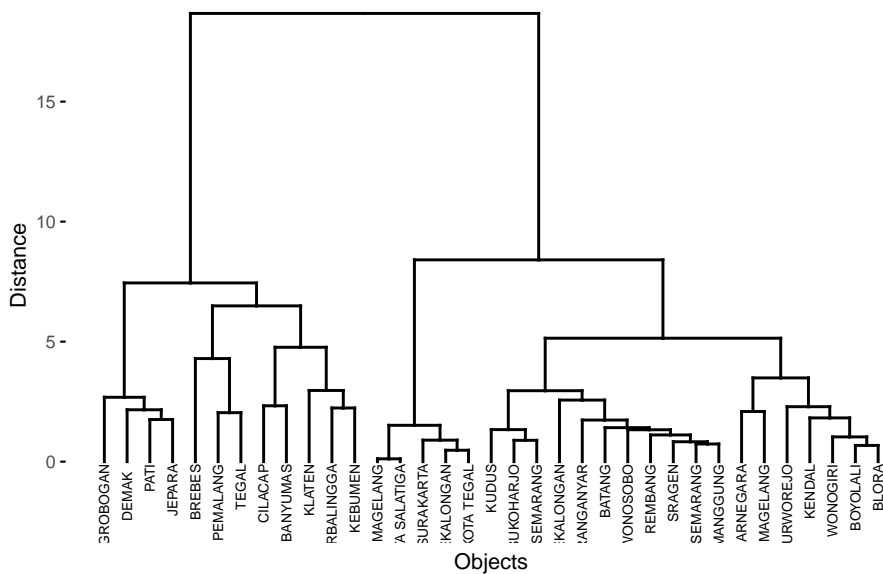
```
library(factoextra)
fviz_dend(hc, cex = 0.5)
```

Cluster Dendrogram



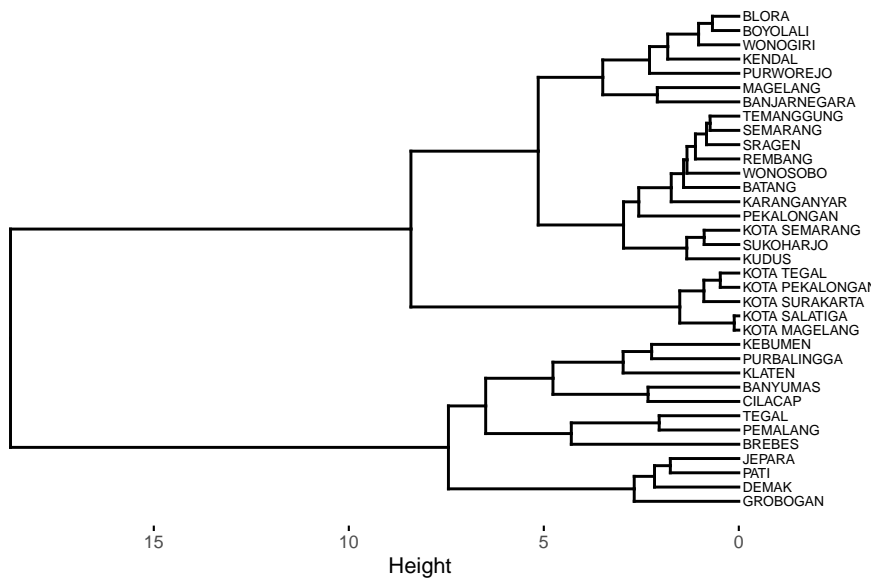
```
fviz_dend(hc, cex = 0.5,
main = "Dendrogram - ward.D2",
xlab = "Objects", ylab = "Distance", sub = "")
```

Dendrogram – ward.D2



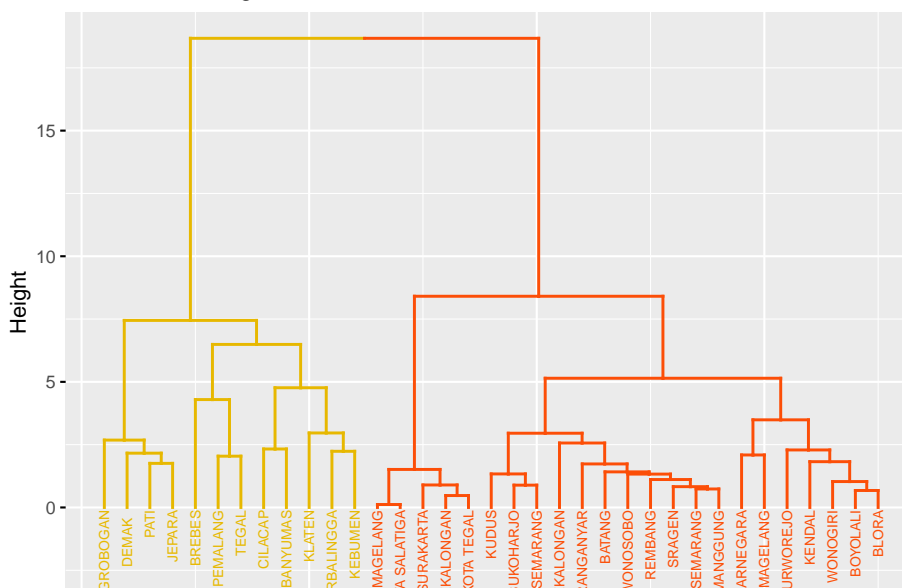
```
fviz_dend(hc, cex = 0.5, horiz = TRUE)
```


Cluster Dendrogram



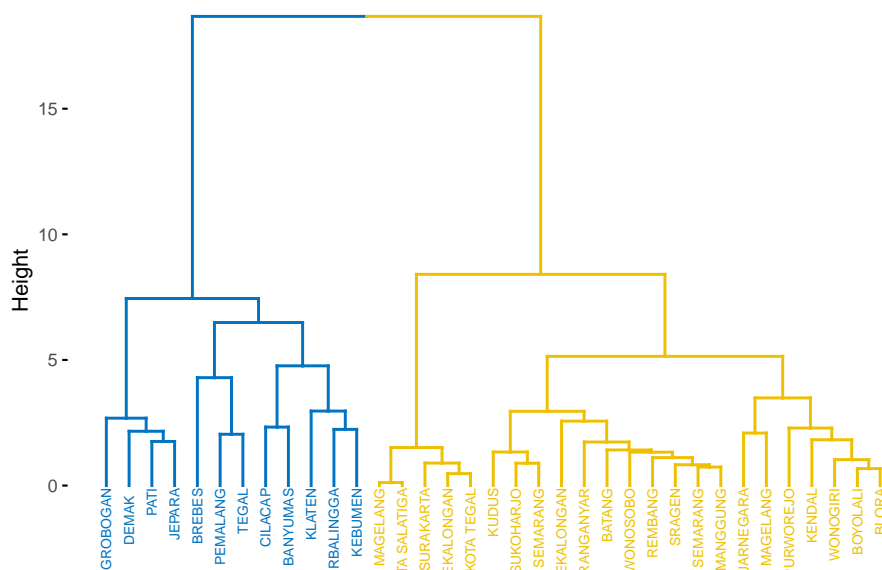
```
fviz_dend(hc, k = 2,
# Cut in four groups
cex = 0.5,
# label size
k_colors = c("#E7B800", "#FC4E07"),
color_labels_by_k = TRUE, # color labels by groups
ggtheme = theme_gray()
# Change theme
)
```

Cluster Dendrogram



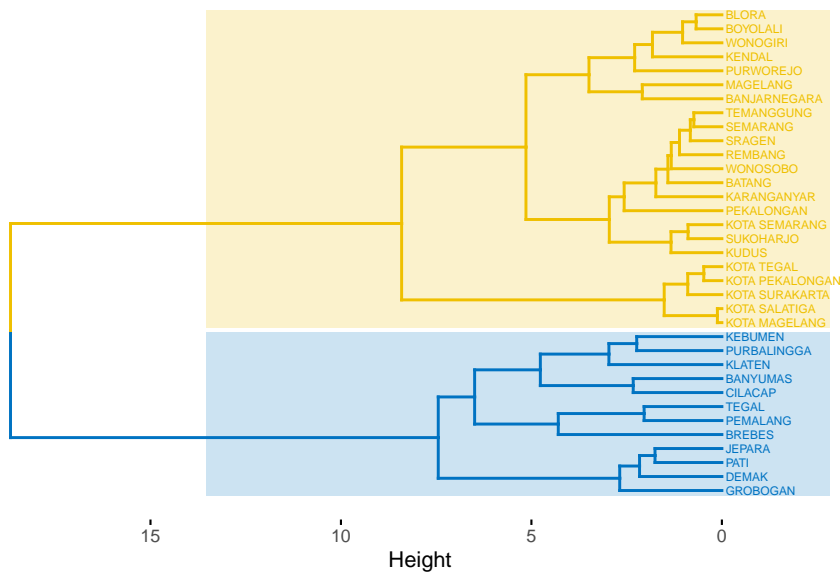
```
fviz_dend(hc, cex = 0.5, k = 2, # Cut in four groups
k_colors = "jco")
```

Cluster Dendrogram

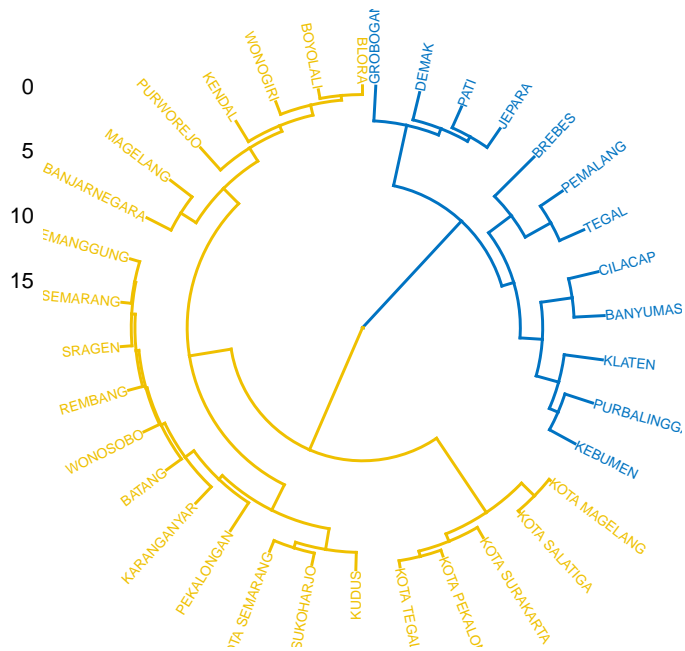


```
fviz_dend(hc, k = 2, cex = 0.4, horiz = TRUE, k_colors = "jco",
rect = TRUE, rect_border = "jco", rect_fill = TRUE)
```

Cluster Dendrogram



```
fviz_dend(hc, cex = 0.5, k = 2,
k_colors = "jco", type = "circular")
```



```
require("igraph")
```

```
## Loading required package: igraph
```

```
##
```

```
## Attaching package: 'igraph'
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

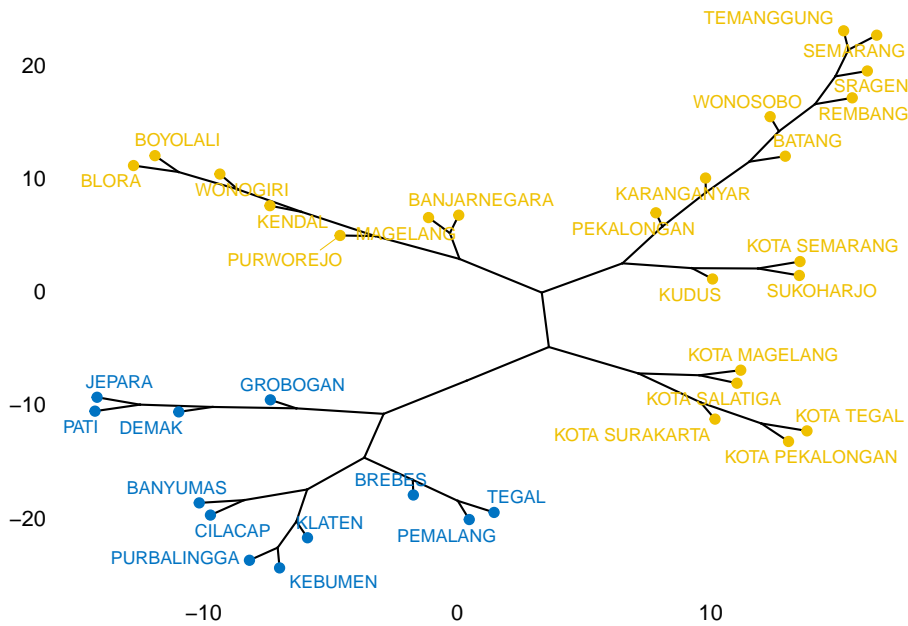
```
## decompose, spectrum
```

```
## The following object is masked from 'package:base':
```

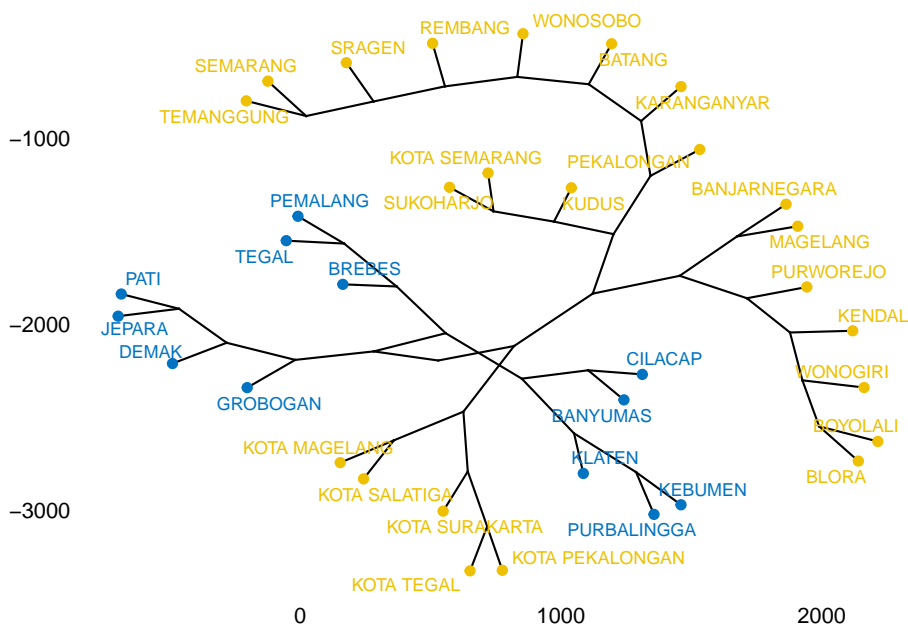
```
##
```

```
## union
```

```
fviz_dend(hc, k = 2, k_colors = "jco",
type = "phylogenetic", repel = TRUE)
```



```
require("igraph")
fviz_dend(hc, k = 2, # Cut in four groups
k_colors = "jco",
type = "phylogenetic", repel = TRUE,
phylo_layout = "layout.gem")
```



Chapter 5 Evaluasi Analisis Cluster

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0    Min.   : 2.00
## 1st Qu.:12.0    1st Qu.: 26.00
##  Median :15.0    Median : 36.00
##   Mean  :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
##   Max.  :25.0    Max.   :120.00
```

5.1 Including Plots

You can also embed plots, for example:

```
par(mar = c(4, 4, .1, .1))
plot(pressure)
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

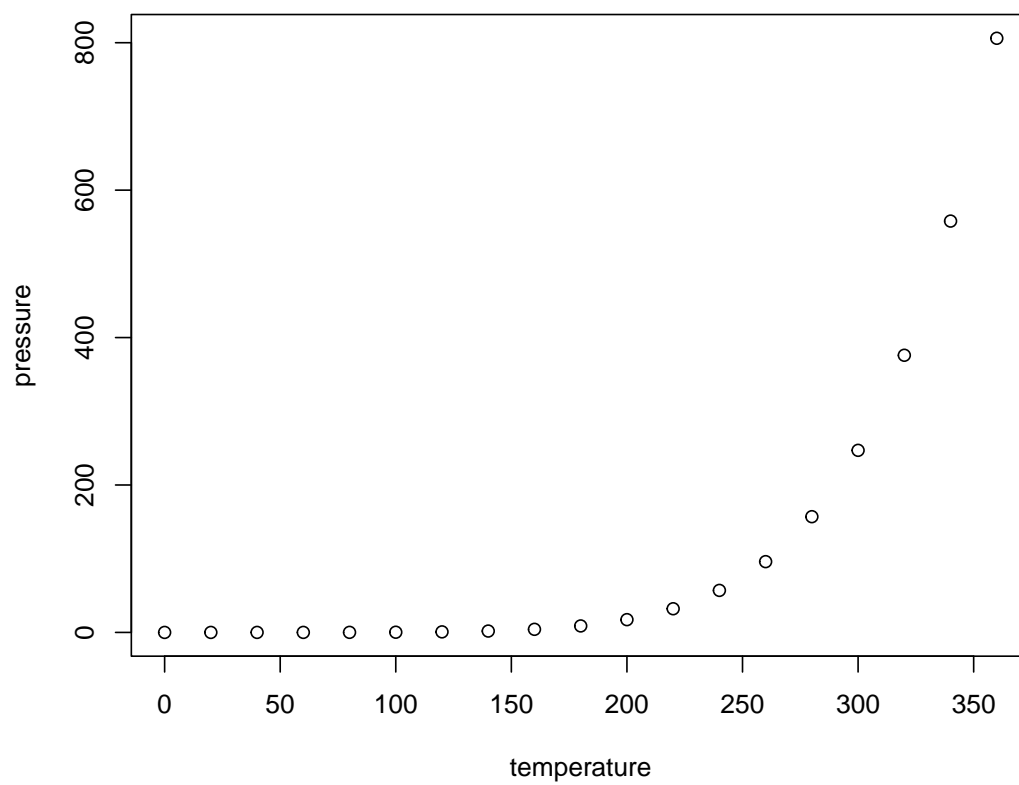


Figure 5.1: Here is another nice figure!

Referensi

- Aggarwal, Charu C., and Chandan K. Reddy, eds. 2014. *Data Clustering: Algorithms and Applications*. CRC Press. <http://www.charuaggarwal.net/clusterbook.pdf>.
- Bandyopadhyay, Sanghamitra, Sriparna Saha, and Witold Pedrycz. 2011. "Use of a Fuzzy Granulation-degranulation Criterion for Assessing Cluster Validity." *Fuzzy Sets and Systems* 170 (1): 22–42. <https://doi.org/10.1016/j.fss.2010.11.015>.
- Bezdek, James C., Robert Ehrlich, and William Full. 1984. "FCM: The Fuzzy c-Means Clustering Algorithm." *Computers & Geosciences* 10 (2-3): 191–203. [https://doi.org/10.1016/0098-3004\(84\)90020-7](https://doi.org/10.1016/0098-3004(84)90020-7).
- Han, Jiawei, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques*. 3rd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers.
- Pimentel, Bruno Almeida, and Renata M. C. R. de Souza. 2016. "Multivariate Fuzzy C-Means Algorithms with Weighting." *Neurocomputing* 174 (January): 946–65. <https://doi.org/10.1016/j.neucom.2015.10.011>.
- Stetco, Adrian, Xiao-Jun Zeng, and John Keane. 2015. "Fuzzy C-Means++: Fuzzy C-Means with Effective Seeding Initialization." *Expert Systems with Applications* 42 (21): 7541–48. <https://doi.org/10.1016/j.eswa.2015.05.014>.