



Kuliah 8 – STA1381 Pengantar Sains Data

Nonlinear Regression (1)

Septian Rahardiantoro

Rencana Perkuliahan: Sesi UAS

Kuliah	Materi	Hari/ Tanggal	Keterangan
8	Nonlinear Regression I	Jumat/ 21 Okt '22	Offline
9	Nonlinear Regression II	Jumat/ 28 Okt '22	Online
10	Pengenalan Machine Learning	Jumat/ 4 Nov '22	Offline
11	Variables Selection	Jumat/ 11 Nov '22	Online
12	Pengenalan Unsupervised Learning	Jumat/ 18 Nov '22	Offline
13	Beberapa Aplikasi	Jumat/ 25 Nov '22	Online
14	Presentasi kelompok	Jumat/ 2 Des '22	Offline

Komponen Tugas

- **Tugas Kelompok untuk sesi UAS**
 - Membuat poster tentang suatu analisis data
 - Data: **akan diinfokan segera**
 - **ISI POSTER:** Latar Belakang, Metodologi (Data & Analisis Data), Hasil & Pembahasan, dan Kesimpulan
 - Deadline waktu pengumpulan: Rabu/ 30 Nov '22
 - Presentasi hasil poster pada pertemuan 14: Jumat/ 2 Des '22
- **Tugas Individu** → ada suatu kasus yang harus diselesaikan
 - Diberikan pada pertemuan 12: Jumat/ 18 Nov '22
 - Deadline waktu pengumpulan: Jumat/ 25 Nov '22 (sebelum waktu kuliah)

Outline

- Pengantar
- Aplikasi
- Beberapa metode dalam regresi nonlinier
 - Regresi polinomial
 - Regresi fungsi tangga
 - Basis function

Pengantar

- Regresi nonlinier adalah suatu bentuk analisis regresi di mana data cocok (pas) dengan sebuah model dan kemudian dinyatakan sebagai fungsi matematika.
- Ketika data menunjukkan hubungan melengkung yang bukan garis lurus, menerapkan model nonlinier memberikan output yang lebih akurat.
- Regresi linier sederhana menghubungkan dua variabel (X dan Y) dengan garis lurus ($y = mx + b$), sedangkan regresi nonlinier menghubungkan kedua variabel dalam hubungan nonlinier (melengkung).
- Tidak seperti regresi linier, tidak ada asumsi linieritas data dalam model nonlinier. Intinya, ketika kurva data tidak dapat dibentuk secara akurat menggunakan metode linier, opsi selanjutnya adalah metode nonlinier karena mengakomodasi beragam jenis kurva. Kesesuaian kurva menentukan kebenarannya, dan kurva yang diilustrasikan mencerminkan akuntabilitas kurva terhadap data.

- Tujuan dari model ini adalah untuk membuat jumlah kuadrat sekecil mungkin. Jumlah kuadrat adalah ukuran yang melacak seberapa jauh pengamatan Y bervariasi dari fungsi nonlinier (melengkung) yang digunakan untuk memprediksi Y .
- Konsepnya, dengan dihitung terlebih dahulu selisih (difference) antara fungsi nonlinier yang diduga dengan setiap titik Y . Kemudian, masing-masing selisih (difference) tersebut dikuadratkan. Terakhir, semua angka kuadrat dijumlahkan. Semakin kecil jumlah angka kuadrat ini, semakin baik fungsinya sesuai dengan titik datanya.
- Regresi nonlinier menggunakan fungsi logaritma, fungsi trigonometri, fungsi eksponensial, fungsi pangkat, kurva Lorenz, fungsi Gaussian, dan metode fitting lainnya.

- Salah satu rumus yang digunakan untuk mewakili model nonlinier tercantum di bawah ini.
$$Y = f(X, \beta) + \varepsilon$$
- Dimana f adalah fungsi regresi dan ε adalah galat sedangkan β adalah vektor parameter.

KEY TAKEAWAYS

- Both linear and nonlinear regression predict Y responses from an X variable (or variables).
- Nonlinear regression is a curved function of an X variable (or variables) that is used to predict a Y variable
- Nonlinear regression can show a prediction of population growth over time.

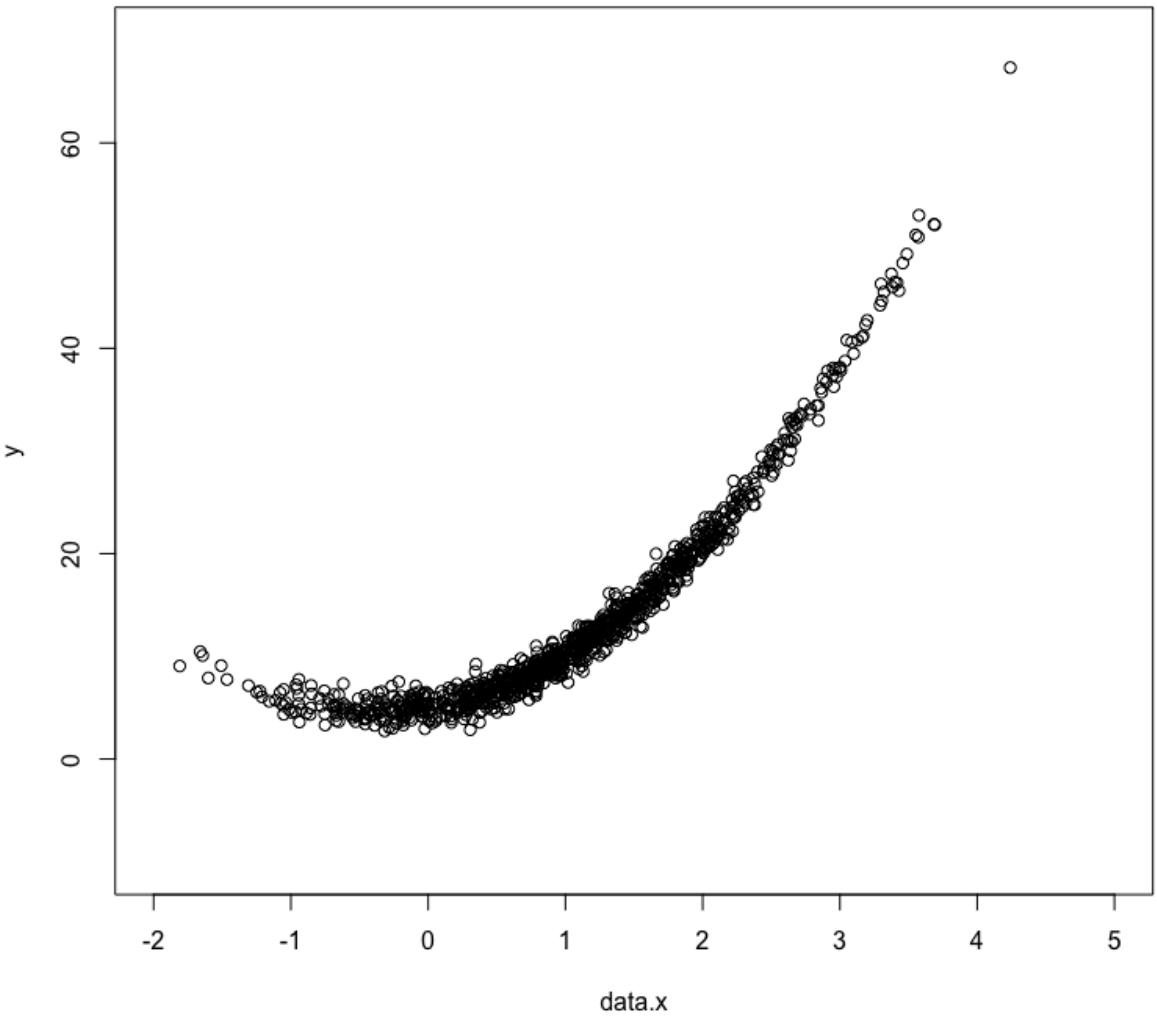
<https://www.investopedia.com/terms/n/nonlinear-regression.asp>

Ilustrasi dengan simulasi di R

```
set.seed(123)
data.x <- rnorm(1000,1,1)
err <- rnorm(1000)

y <- 5+2*data.x+3*data.x^2+err

plot(data.x,y,xlim=c(-2,5),ylim=c(-10,70))
```



Dengan menggunakan regresi linier

```
lin.mod <- lm(y~data.x)
lines(data.x,lin.mod$fitted.values,col="red")
summary(lin.mod)

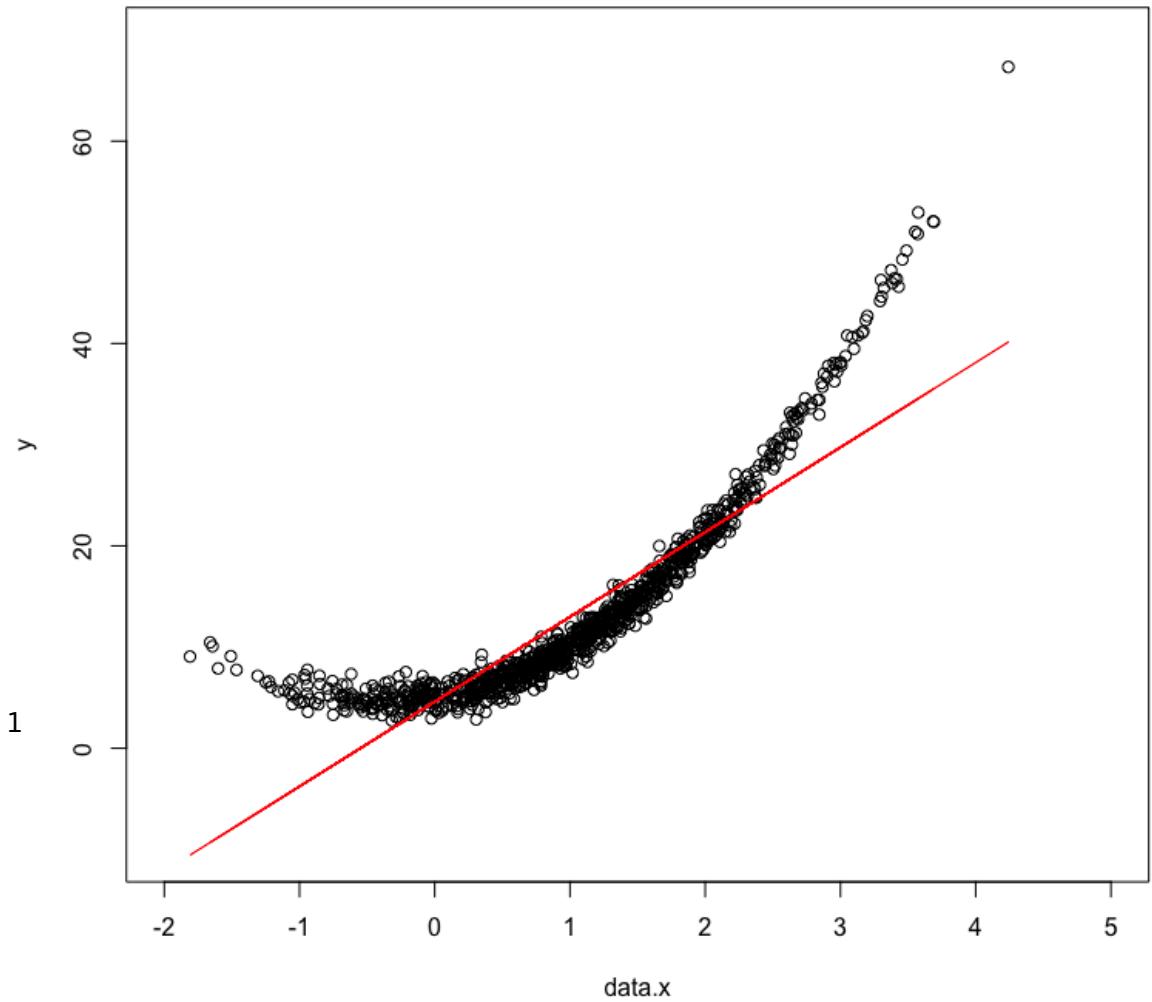
> summary(lin.mod)

Call:
lm(formula = y ~ data.x)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.686 -2.574 -1.428  1.195 27.185 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.6056    0.1902   24.22   <2e-16 ***
data.x       8.3790    0.1340   62.54   <2e-16 ***  
---
Signif. codes:  0 '****' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

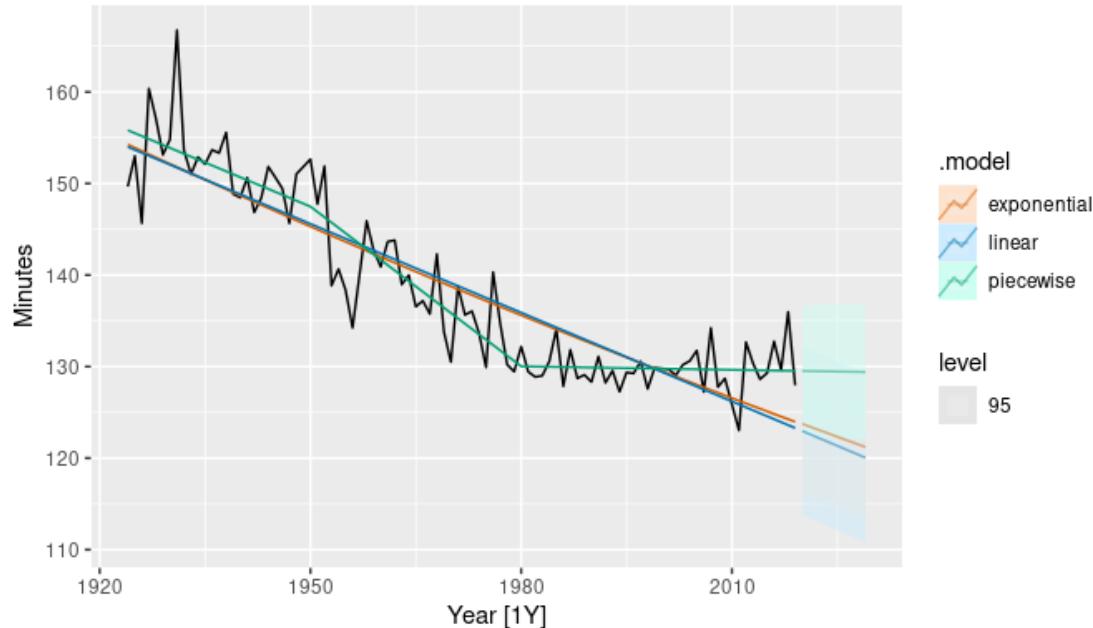
Residual standard error: 4.2 on 998 degrees of freedom
Multiple R-squared:  0.7967, Adjusted R-squared:  0.7965 
F-statistic: 3911 on 1 and 998 DF,  p-value: < 2.2e-16
```



Aplikasi

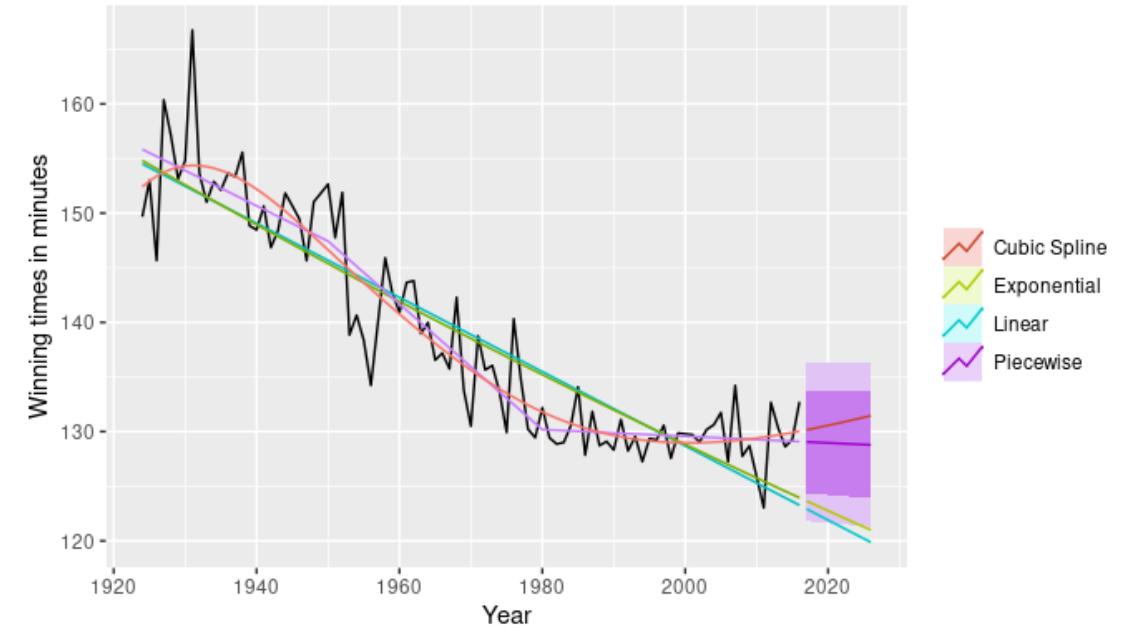
- Model regresi nonlinier sebagian besar digunakan untuk prediksi, pemodelan keuangan, dan tujuan peramalan.
- Model nonlinier digunakan di banyak bidang dan sektor seperti asuransi, pertanian, keuangan, investasi, AI pembelajaran mesin, dan memahami pasar yang lebih luas. Mari kita lihat beberapa aplikasi penting:
 - Karena sebagian besar proses biologis bersifat nonlinier, kita dapat menemukan aplikasi model nonlinier dalam penelitian kehutanan. Contohnya, fungsi sederhana untuk menghubungkan volume atau berat pohon dalam kaitannya dengan diameter atau tingginya.
 - Penggunaan model nonlinier dalam mengembangkan gas tak berwarna dengan jangkauan luas, formulasi HCFC-22 (contoh dari bidang Kimia)
 - Dalam penelitian dan pengembangan, digunakan dalam proses perumusan masalah dan mendapatkan solusi statistik untuk masalah kalibrasi.
 - Ini digunakan dalam domain asuransi. Sebagai contoh, penggunaannya dapat dilihat pada perhitungan cadangan IBNR (Incured But Not Reported Reserves (cadangan yang terjadi tetapi tidak dilaporkan)) → cadangan untuk klaim yang menjadi jatuhan tempo dengan terjadinya peristiwa yang tercakup dalam polis asuransi, tetapi belum dilaporkan.
 - Ini sangat penting dalam penelitian pertanian. Karena banyak proses tanaman dan tanah lebih baik ditangkap oleh model nonlinier daripada model linier.

Boston marathon winning times



https://www.google.com/url?sa=i&url=https%3A%2F%2Fotexts.com%2Ffpp3%2Fnonlinear-regression.html&psig=AOvVaw09lt6MsE7eUhC46xdg1GTF&ust=1666235547165000&source=images&cd=vfe&ved=0CA0QjRxqFwoTCPDEx6up6_oCFQAAAAAdAAAAABAR

Boston Marathon



https://www.google.com/url?sa=i&url=https%3A%2F%2Fotexts.com%2Ffpp2%2Fnonlinear-regression.html&psig=AOvVaw09lt6MsE7eUhC46xdg1GTF&ust=1666235547165000&source=images&cd=vfe&ved=0CA0QjRxqFwoTCPDEx6up6_oCFQAAAAAdAAAABAN

Beberapa metode dalam regresi nonlinier

1. Regresi Polinomial

$$y_i = \beta_0 + \beta_1 x + \varepsilon_i$$

Model linier



$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_d x_i^d + \varepsilon_i$$

Fungsi Polinomial

Regresi Polinomial

- Regresi polinomial dapat diduga dengan menggunakan MKT (OLS), karena setara dengan linier model dengan prediktornya adalah $x_i, x_i^2, x_i^3, \dots, x_i^d$
- Pada umumnya, penggunaan d lebih besar 3 atau 4 sangat tidak dianjurkan, karena kurva polinomial dapat menjadi terlalu fleksibel dan dapat menyerupai beberapa bentuk yang sangat aneh.

ilustrasi

Degree-4 Polynomial

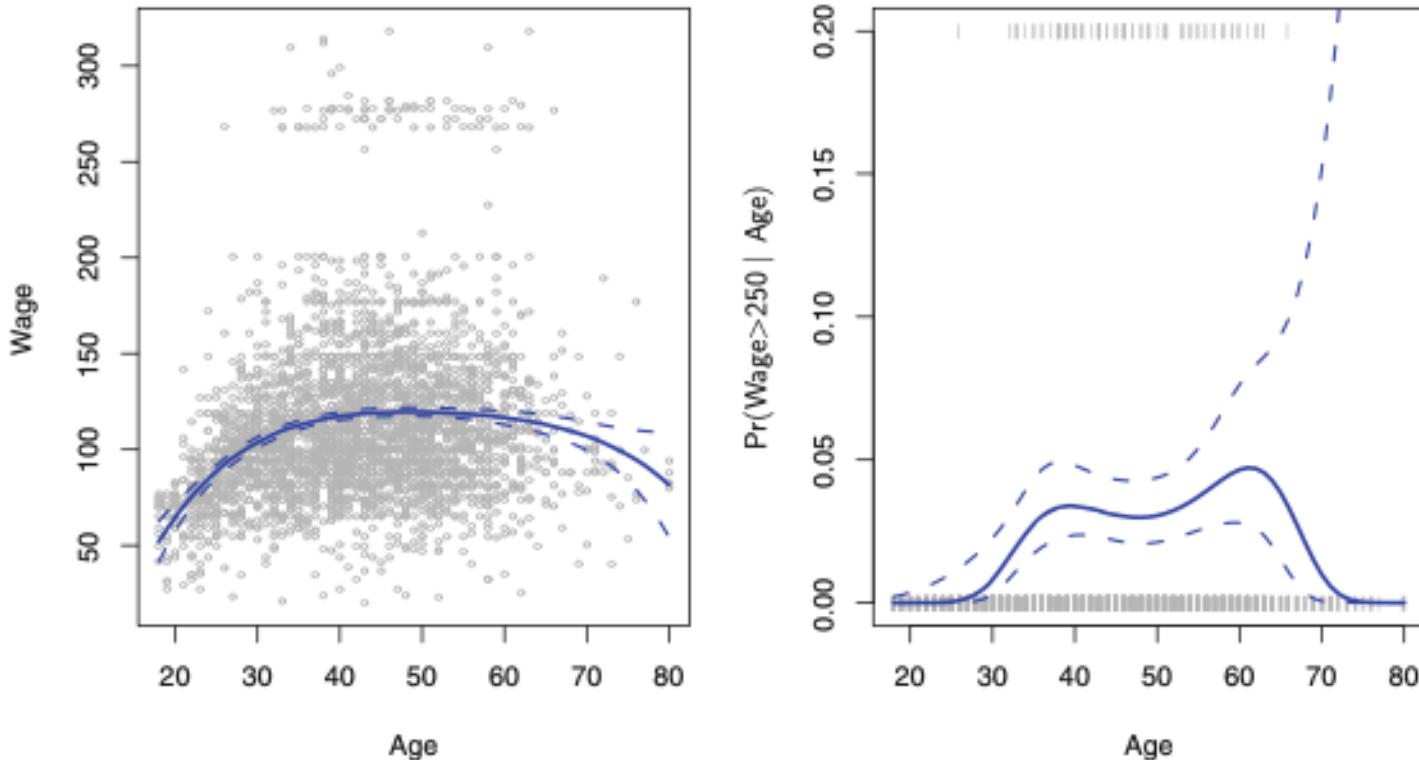


FIGURE 7.1. The `Wage` data. Left: The solid blue curve is a degree-4 polynomial of `wage` (in thousands of dollars) as a function of `age`, fit by least squares. The dotted curves indicate an estimated 95 % confidence interval. Right: We model the binary event `wage>250` using logistic regression, again with a degree-4 polynomial. The fitted posterior probability of `wage` exceeding \$250,000 is shown in blue, along with an estimated 95 % confidence interval.

Ilustrasi di R

```
#regresi polinomial
pol.mod <- lm(y~data.x+I(data.x^2))
ix <- sort(data.x, index.return=T)$ix
lines(data.x[ix],pol.mod$fitted.values[ix],col="blue",cex=2)

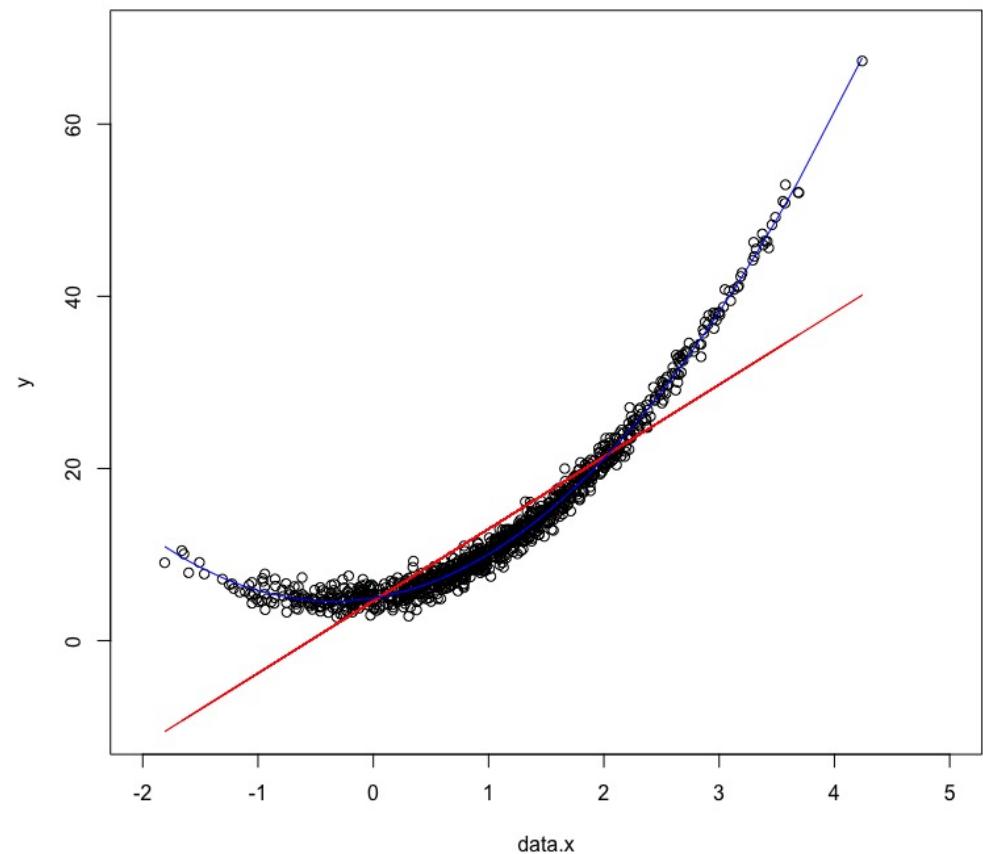
summary(pol.mod)
> summary(pol.mod)

Call:
lm(formula = y ~ data.x + I(data.x^2))

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0319 -0.6942  0.0049  0.7116  3.2855 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 4.95193   0.04568 108.41   <2e-16 ***
data.x       2.10732   0.05861  35.95   <2e-16 ***
I(data.x^2)  2.99081   0.02338 127.93   <2e-16 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.007 on 997 degrees of freedom
Multiple R-squared:  0.9883, Adjusted R-squared:  0.9883 
F-statistic: 4.221e+04 on 2 and 997 DF,  p-value: < 2.2e-16
```



2. Regresi Fungsi Tangga (Step Functions)

- Menggunakan fungsi polinomial sebelumnya, memaksakan struktur atau bentuk nonlinier X secara global
- Sebagai gantinya, kita dapat menggunakan fungsi tangga (step function) untuk menghindari penerapan struktur global seperti itu.
- Di sini kita memecah rentang X ke dalam bin, dan memasukkan konstanta yang berbeda di setiap bin. Ini sama dengan mengubah variabel kontinu menjadi variabel kategoris terurut.
- Detailnya, kita buat titik potong (cutpoints) c_1, c_2, \dots, c_K di dalam rentang X , lalu membentuk $K + 1$ peubah baru

$$\begin{aligned} C_0(X) &= I(X < c_1), \\ C_1(X) &= I(c_1 \leq X < c_2), \\ C_2(X) &= I(c_2 \leq X < c_3), \\ &\vdots \\ C_{K-1}(X) &= I(c_{K-1} \leq X < c_K), \\ C_K(X) &= I(c_K \leq X), \end{aligned}$$

Dummy variables $C_0(X) + C_1(X) + \dots + C_K(X) = 1$

- Pendugaan parameternya dapat diselesaikan dengan MKT (OLS) -> setara dengan model linier

Modelnya: $y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \dots + \beta_K C_K(x_i) + \epsilon_i$.

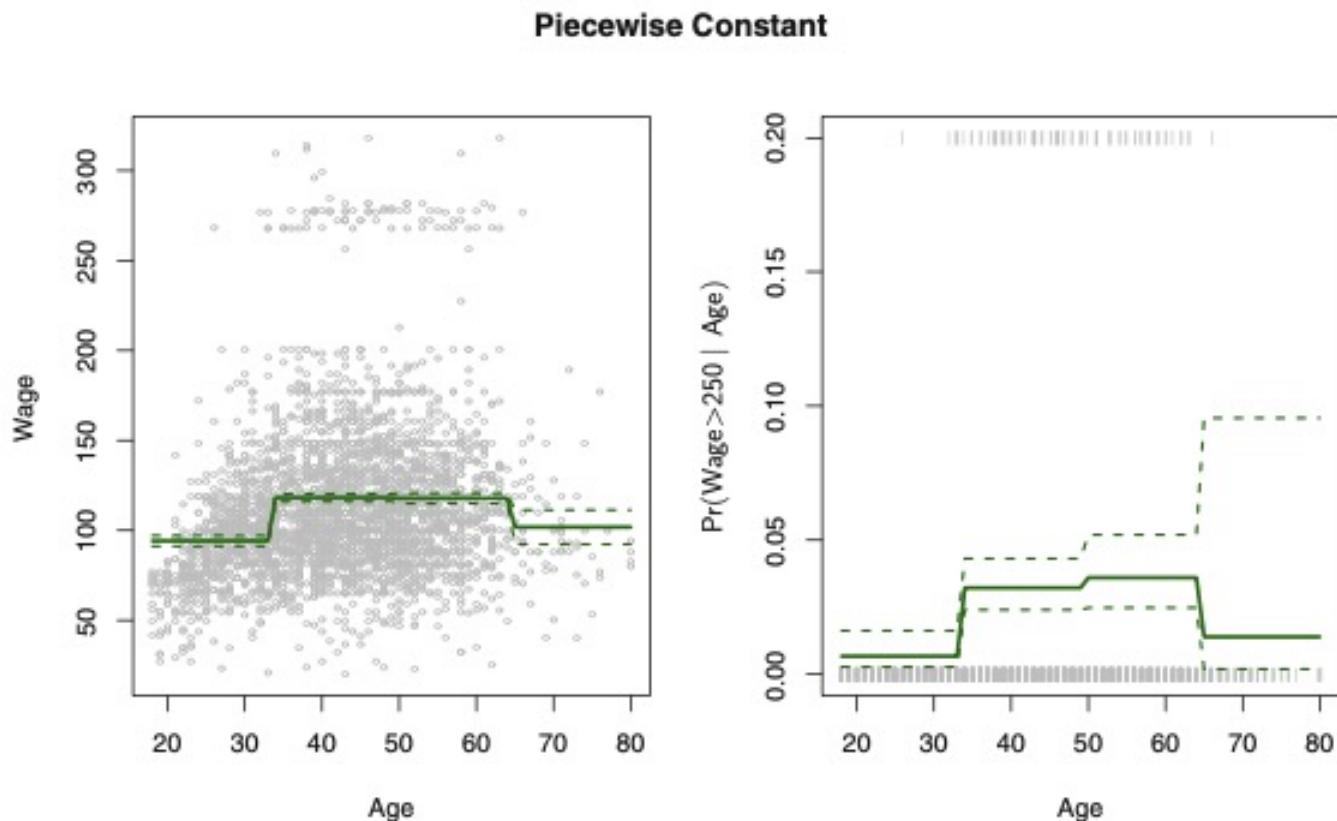


FIGURE 7.2. The `Wage` data. Left: The solid curve displays the fitted value from a least squares regression of `wage` (in thousands of dollars) using step functions of `age`. The dotted curves indicate an estimated 95 % confidence interval. Right: We model the binary event `wage>250` using logistic regression, again using step functions of `age`. The fitted posterior probability of `wage` exceeding \$250,000 is shown, along with an estimated 95 % confidence interval.

Ilustrasi di R

```
#regresi fungsi tangga
range(data.x)
c1 <- as.factor(ifelse(data.x<=0,1,0))
c2 <- as.factor(ifelse(data.x<=2 & data.x>0,1,0))
c3 <- as.factor(ifelse(data.x>2,1,0))
step.mod <- lm(y~c1+c2+c3)
lines(data.x[ix],step.mod$fitted.values[ix],col="green")
```

```
summary(step.mod)
```

```
> summary(step.mod)
```

Call:

```
lm(formula = y ~ c1 + c2 + c3)
```

Residuals:

Min	1Q	Median	3Q	Max
-10.395	-3.534	-0.530	2.527	36.876

Coefficients: (1 not defined because of singularities)

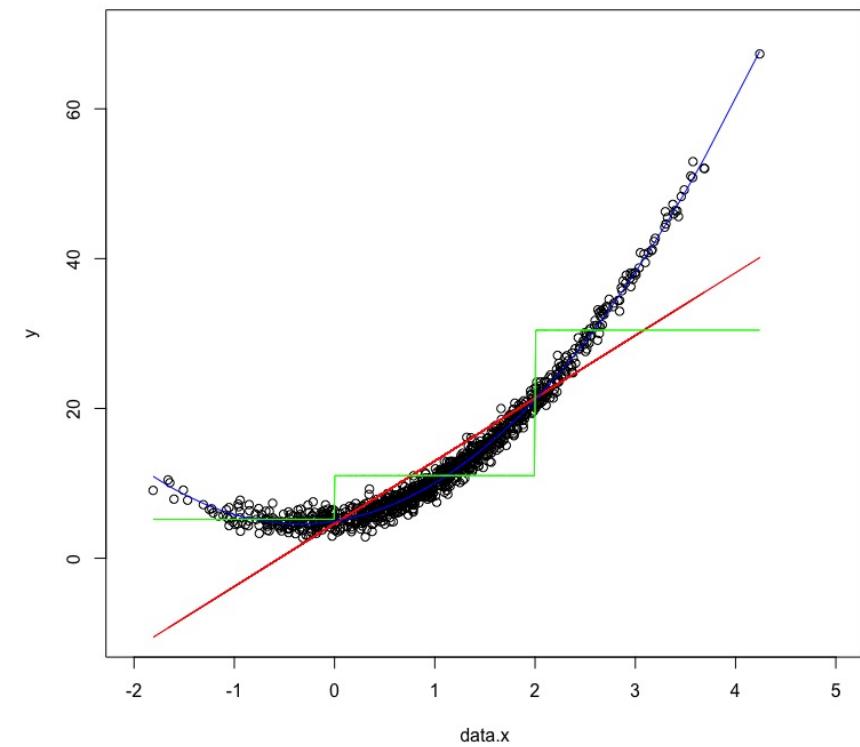
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	30.4499	0.4087	74.50	<2e-16 ***
c11	-25.2395	0.5710	-44.21	<2e-16 ***
c21	-19.4184	0.4536	-42.81	<2e-16 ***
c31	NA	NA	NA	NA

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.121 on 997 degrees of freedom

Multiple R-squared: 0.698, Adjusted R-squared: 0.6974

F-statistic: 1152 on 2 and 997 DF, p-value: < 2.2e-16



3. Basis Function

- Model regresi polinomial dan regresi fungsi tangga sebenarnya adalah kasus khusus dari pendekatan *basis function regression*.
- Idenya adalah untuk menetapkan suatu bentuk transformasi yang dapat diterapkan pada variabel X , yaitu: $b_1(X), b_2(X), \dots, b_K(X)$, sehingga modelnya menjadi:

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \dots + \beta_K b_K(x_i) + \epsilon_i.$$

dengan $b_1(\cdot), b_2(\cdot), \dots, b_K(\cdot)$ disebut *basis functions* yang bersifat *fixed and known*

- Akibatnya, berlaku:
 - Polinomial: $b_j(x_i) = x_i^j$
 - Fungsi tangga: $b_j(x_i) = I(c_j \leq x_i \leq c_{j+1})$
- Lebih lanjut, model ini setara dengan model linier dengan prediktor $b_1(x_i), b_2(x_i), \dots, b_K(x_i)$, sehingga kita dapat menggunakan MKT (OLS) dalam pendugaan parameternya
- Tentunya ada *basis function* yang lainnya: *regression splines* (**akan dibahas pada pertemuan selanjutnya**)

Terima kasih 😊



Kuliah 9 – STA1381 Pengantar Sains Data

Nonlinear Regression (2)

Septian Rahardiantoro

Outline

- Review
- Regression splines
- Smoothing splines

Review

- Regresi nonlinier:

- Regresi polinomial

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \dots + \beta_d x_i^d + \varepsilon_i$$

- Regresi fungsi tangga

$$y_i = \beta_0 + \beta_1 C_1(x_i) + \beta_2 C_2(x_i) + \dots + \beta_K C_K(x_i) + \epsilon_i.$$

- Basis function

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \beta_3 b_3(x_i) + \dots + \beta_K b_K(x_i) + \epsilon_i.$$

Regression Splines

- Regression splines merupakan salah satu bentuk basis function yang sering digunakan dalam nonlinear regression
- Idenya dengan menggunakan fungsi polinomial derajat- d pada setiap bin (selang), di bawah batasan bahwa fungsi polinomial tersebut (dan mungkin turunan pertamanya) bersifat kontinu.
- Regression splines menggunakan gabungan dari fungsi polinomial dengan fungsi tangga, yakni menerapkan fungsi polinomial pada setiap selang yang ditentukan
- Akibatnya terdapat dua hal yang penting untuk diperhatikan:
 - Penggunaan fungsi polinomial untuk setiap bin
 - Pemilihan batas bin yang sesuai (lebih lanjut disebut sebagai *knots*)

- Penggunaan fungsi polinomial untuk setiap bin
 - Daripada menggunakan fungsi polinomial tingkat tinggi di seluruh rentang X , regresi polinomial sepotong-sepotong (*piecewise polynomial regression*) untuk setiap bin menggunakan fungsi polinomial tingkat rendah yang terpisah di berbagai wilayah X .
 - Misalnya, *piecewise cubic polynomial* bekerja dengan memasang model regresi kubik dari bentuk

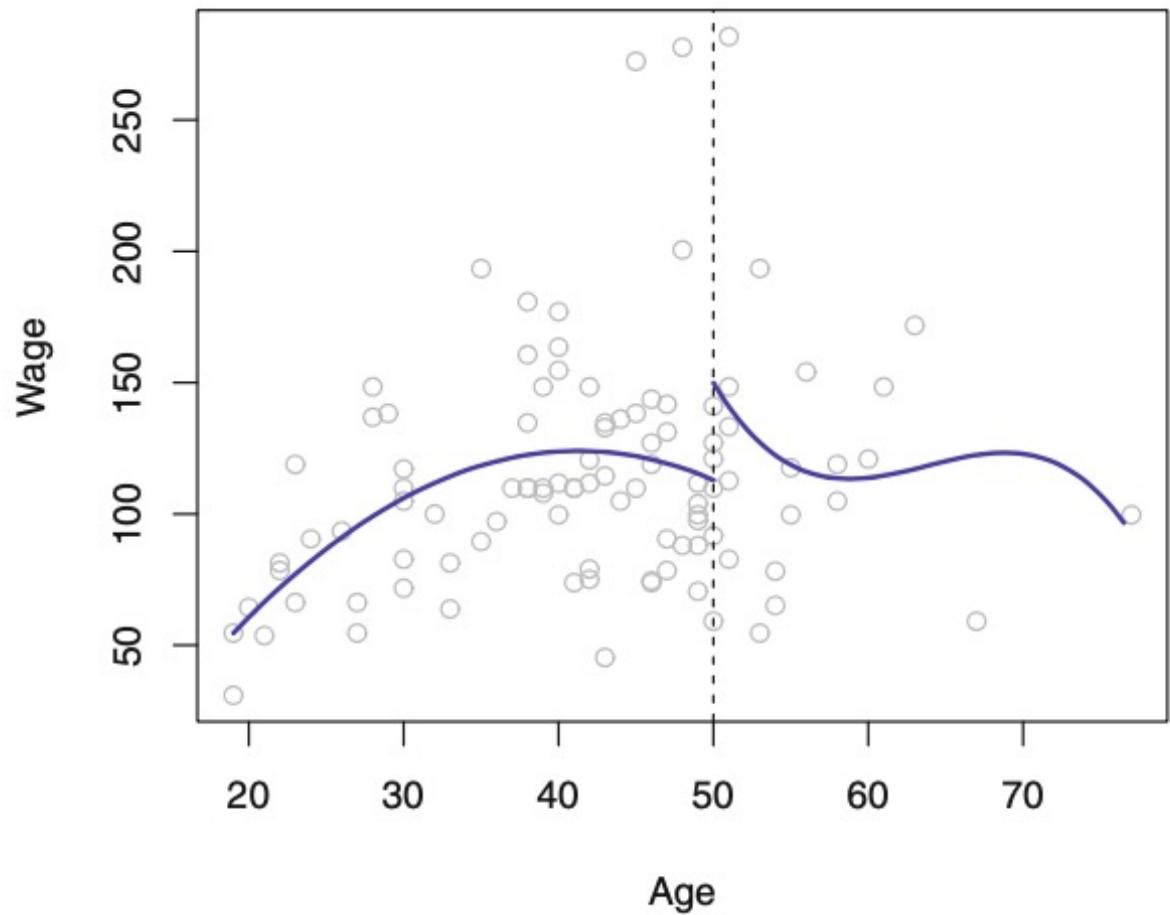
$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \varepsilon_i$$
 dengan koefisien $\beta_0, \beta_1, \beta_2, \beta_3$ berbeda-beda di bagian yang berbeda dari rentang X .
 - Titik-titik di mana koefisien berubah disebut *knots*.

Ilustrasi penggunaan *piecewise cubic polynomial* dengan sebuah knot (titik c)

$$y_i = \begin{cases} \beta_{01} + \beta_{11}x_i + \beta_{21}x_i^2 + \beta_{31}x_i^3 + \varepsilon_i & ; x_i < c \\ \beta_{02} + \beta_{12}x_i + \beta_{22}x_i^2 + \beta_{32}x_i^3 + \varepsilon_i & ; x_i \geq c \end{cases}$$

- Dengan kata lain, kita memasukkan dua fungsi polinomial yang berbeda ke data, satu pada subset pengamatan dengan $x_i < c$, dan satu lagi pada subset pengamatan dengan $x_i \geq c$.
- Fungsi polinomial pertama memiliki koefisien $\beta_{01}, \beta_{11}, \beta_{21}, \beta_{31}$, dan yang kedua memiliki koefisien $\beta_{02}, \beta_{12}, \beta_{22}, \beta_{32}$.
- Masing-masing fungsi polinomial ini dapat diduga dengan menggunakan MKT yang diterapkan pada fungsi dari prediktor asalnya.

Piecewise Cubic



Ilustrasi di R

```
set.seed(123)
data.x <- rnorm(1000,1,1)
err <- rnorm(1000,0,5)

y <- 5+2*data.x+3*data.x^2+err

plot(data.x,y,xlim=c(-2,5),ylim=c(-10,70),pch=16,col="orange")
abline(v=1,col="red",lty=2)

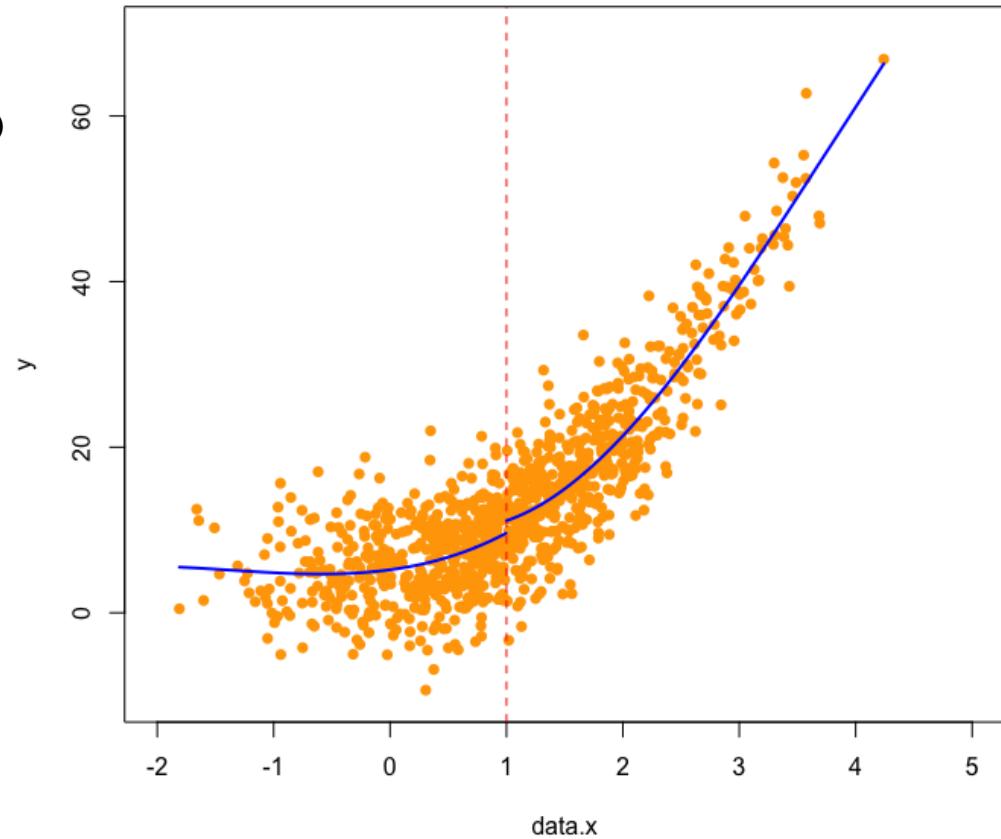
#piecewise cubic polynomial
dt.all <- cbind(y,data.x)

##knots = 1
dt1 <- dt.all[data.x <=1,]
dim(dt1)

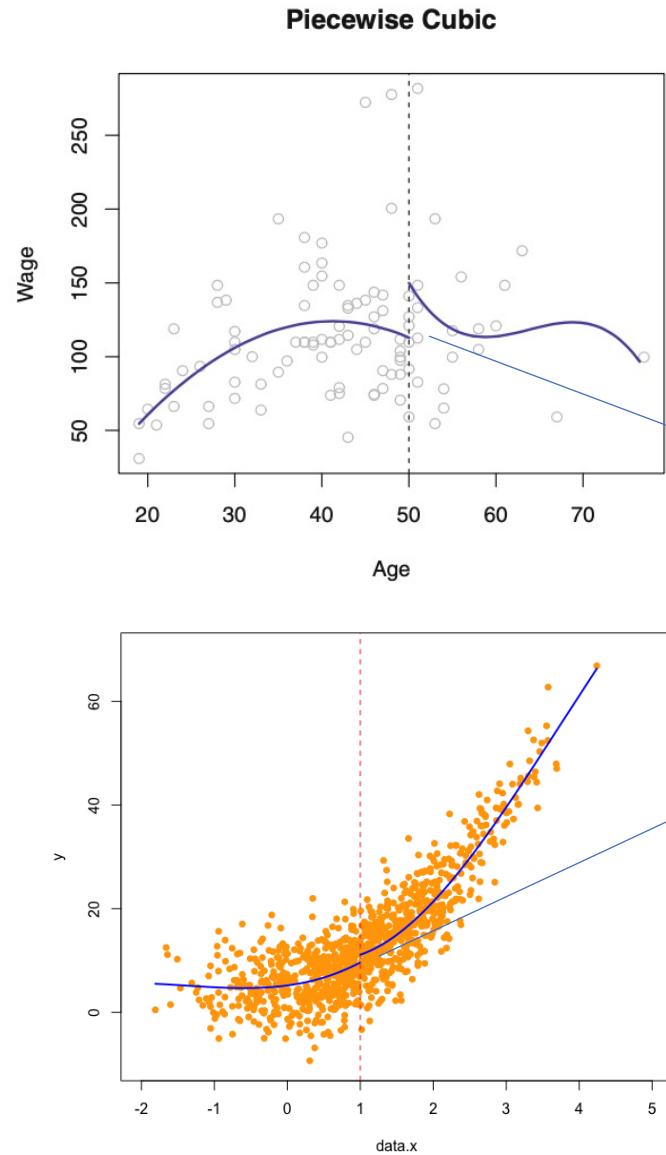
dt2 <- dt.all[data.x >1,]
dim(dt2)

cub.mod1 <- lm(dt1[,1]~dt1[,2]+I(dt1[,2]^2)+I(dt1[,2]^3))
ix <- sort(dt1[,2],index.return=T)$ix
lines(dt1[ix,2],cub.mod1$fitted.values[ix],col="blue",lwd=2)

cub.mod2 <- lm(dt2[,1]~dt2[,2]+I(dt2[,2]^2)+I(dt2[,2]^3))
ix <- sort(dt2[,2],index.return=T)$ix
lines(dt2[ix,2],cub.mod2$fitted.values[ix],col="blue",lwd=2)
```



Perhatikan!



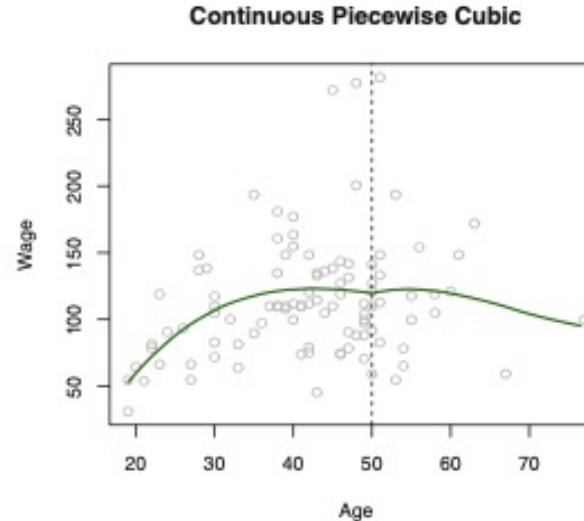
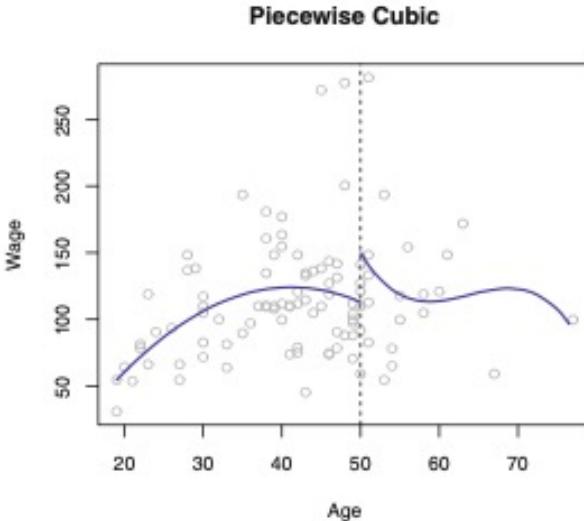
Fungsinya
terputus
pada knot



Diberikan kendala
(constraint) supaya menjadi
kontinu (tersambung)
*The cubic polynomials are
constrained to be continuous*

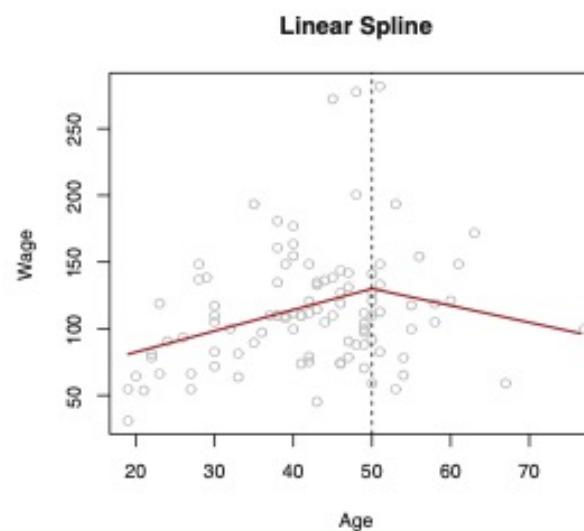
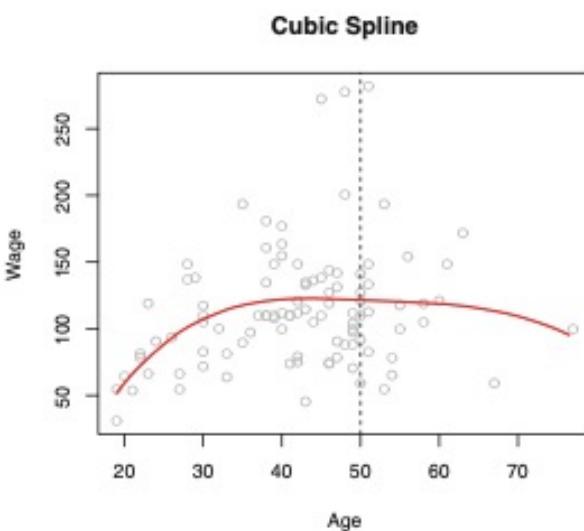
Ilustrasi

The cubic polynomials are unconstrained.



The cubic polynomials are constrained to be continuous at age=50.

The cubic polynomials are constrained to be continuous, and to have continuous first and second derivatives.



A linear spline is constrained to be continuous.

FIGURE 7.3. Various piecewise polynomials are fit to a subset of the `Wage` data, with a knot at `age=50`. Top Left: The cubic polynomials are unconstrained. Top Right: The cubic polynomials are constrained to be continuous at `age=50`. Bottom Left: The cubic polynomials are constrained to be continuous, and to have continuous first and second derivatives. Bottom Right: A linear spline is shown, which is constrained to be continuous.

Lalu bagaimana menerapkan fungsi polinomial mengikuti suatu kendala agar garisnya kontinu (tidak terputus) dengan halus?

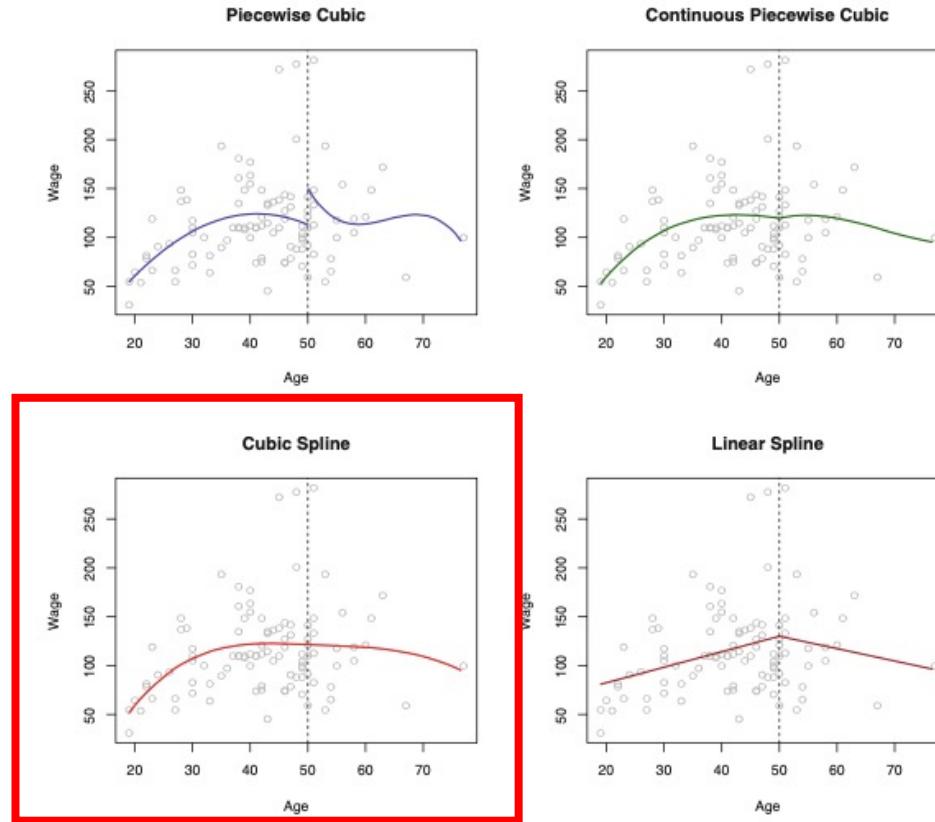
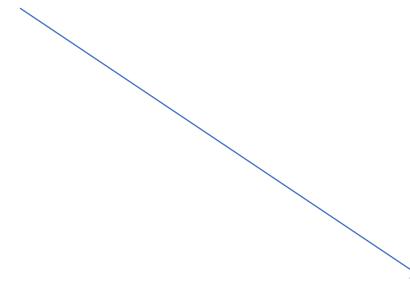


FIGURE 7.3. Various piecewise polynomials are fit to a subset of the `Wage` data, with a knot at `age=50`. Top Left: The cubic polynomials are unconstrained. Top Right: The cubic polynomials are constrained to be continuous at `age=50`. Bottom Left: The cubic polynomials are constrained to be continuous, and to have continuous first and second derivatives. Bottom Right: A linear spline is shown, which is constrained to be continuous.

- Pada umumnya, cara langsung untuk menampilkan *cubic spline* adalah dengan memulai dengan basis dari ***cubic polynomial*** (x, x^2, x^3) dan menambahkan fungsi ***truncated power basis*** untuk setiap knot, yaitu

$$h(x, \xi) = (x - \xi)_+^3 = \begin{cases} (x - \xi)^3 & ; x > \xi \\ 0 & ; x \text{ lainnya} \end{cases} \quad \xi \text{ adalah knot}$$

- Akibatnya, untuk *piecewise cubic polynomial* dengan 1 knot, modelnya menjadi

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 h(x, \xi_1) + \varepsilon_i$$

- Lebih luas, untuk menggunakan *cubic spline* dengan K knot, modelnya menjadi

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 h(x, \xi_1) + \cdots + \beta_{K+3} h(x, \xi_K) + \varepsilon_i$$

atau memodelkan regresi dengan sebuah intersep dan $K + 3$ prediktor, yaitu

$X, X^2, X^3, h(X, \xi_1), \dots, h(X, \xi_K)$, dengan $\xi_1, \xi_2, \dots, \xi_K$ adalah knot.

Sehingga terdapat $K + 4$ predictor yang diduga (derajat bebas = $K + 4$)

- Bentuk umumnya:
- Banyaknya fungsi basis pada regresi spline dengan ordo M dan K buah knot adalah $K + M - 1$ buah, yaitu:

$$b_j(x_i) = x_i^j \quad ; j = 1, 2, \dots, M - 1$$

$$b_{M-1+k} = (x_i - \xi_k)_+^{M-1} \quad ; k = 1, 2, \dots, K.$$

- Fungsi ***truncated power basis*** untuk setiap knot ke- k , yaitu:

$$h(x, \xi_k) = (x - \xi_k)_+^{M-1} = \begin{cases} (x - \xi_k)^{M-1} & ; x > \xi_k \\ 0 & ; x \text{ lainnya} \end{cases}$$

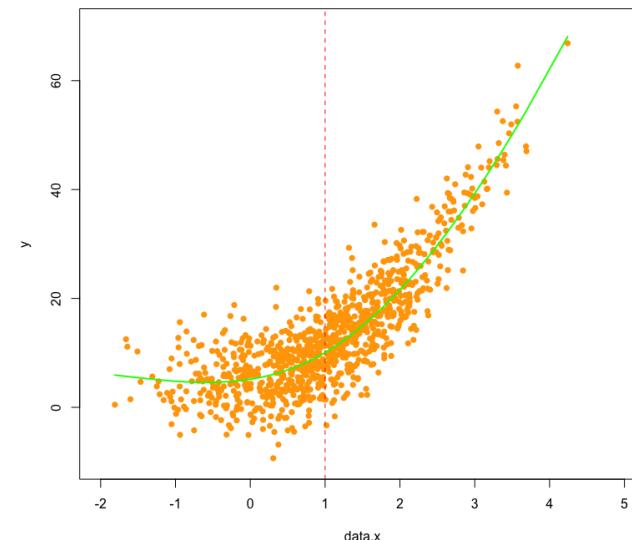
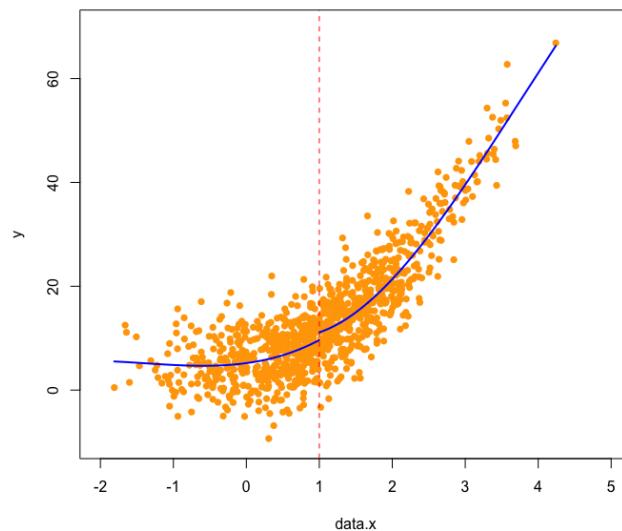
- Namun, spline dapat memiliki keragaman yang sangat tinggi pada nilai X yang sangat rendah atau sangat tinggi. Solusinya adalah dengan menerapkan ***natural spline***, yaitu regresi spline dengan tambahan kendala ***boundary constraint***: fungsi yang diterapkan harus linier pada ujung selang X . (***tidak dibahas dalam kuliah ini***)

Ilustrasi di R

```
#dengan menggunakan truncated power basis
plot(data.x,y,xlim=c(-2,5),ylim=c(-10,70),pch=16,col="orange")
abline(v=1,col="red",lty=2)

hx <- ifelse(data.x>1,(data.x-1)^3,0)

cubspline.mod <- lm(y~data.x+I(data.x^2)+I(data.x^3)+hx)
ix <- sort(data.x,index.return=T)$ix
lines(data.x[ix],cubspline.mod$fitted.values[ix],col="green",lwd=2)
```



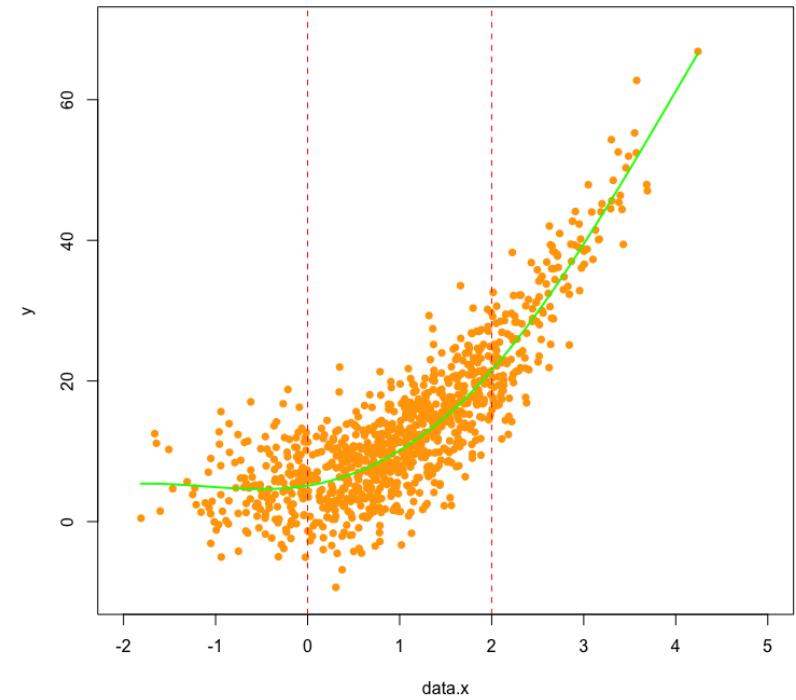
- Pemilihan banyak dan lokasi knot
 - Pada umumnya, banyaknya knot ditentukan berdasarkan besarnya derajat bebas yang ingin digunakan (banyaknya knot yang digunakan ekuivalen dengan besarnya derajat bebas dalam modelnya)
 - Biasanya, banyaknya knot (derajat bebas) dicobakan beberapa buah, kemudian dilihat banyaknya knot yang menampilkan kurva yang terlihat sesuai.
 - Atau, untuk hasil yang lebih objektif, dapat menggunakan pendekatan dengan *cross-validation*
 - Lebih lanjut, lokasi knot umumnya ditetapkan menyebar secara seragam (uniform)

Ilustrasi di R – perbandingan dengan k-fold CV

```
plot(data.x,y,xlim=c(-2,5),ylim=c(-10,70),pch=16,col="orange")
abline(v=0,col="red",lty=2)
abline(v=2,col="red",lty=2)

hx1 <- ifelse(data.x>0,(data.x-0)^3,0)
hx2 <- ifelse(data.x>2,(data.x-2)^3,0)

cubspline.mod2 <- lm(y~data.x+I(data.x^2)+I(data.x^3)+hx1+hx2)
ix <- sort(data.x,index.return=T)$ix
lines(data.x[ix],cubspline.mod2$fitted.values[ix],col="green",lwd=2)
```



```
#perbandingan 1 knot dan 2 knot dengan 5-fold CV
##1 knot
data.all <- cbind(y,data.x,hx)

set.seed(456)
ind <- sample(1:5,length(data.x),replace=T)
res <- c()
for(i in 1:5){
  dt.train <- data.all[ind!=i,]
  x.test <- data.all[ind==i,-1]
  y.test <- data.all[ind==i,1]
  mod1 <- lm(dt.train[,1]~dt.train[,2]+I(dt.train[,2]^2)+I(dt.train[,2]^3)+dt.train[,3])
  x.test.olah <- cbind(1,x.test[,1],x.test[,1]^2,x.test[,1]^3,x.test[,2])
  beta <- coefficients(mod1)
  prediksi <- x.test.olah%*%beta
  res <- c(res,mean((y.test-prediksi)^2))
}
res
mean(res)
```

```
> res
[1] 22.08767 21.39598 29.47677 29.68683 25.18070
> mean(res)
[1] 25.56559
```

```
##2 knot (knot = 0, 2)
data.all2 <- cbind(y,data.x,hx1,hx2)

set.seed(456)
ind2 <- sample(1:5,length(data.x),replace=T)
res2 <- c()
for(i in 1:5){
  dt.train2 <- data.all2[ind2!=i,]
  x.test2 <- data.all2[ind2==i,-1]
  y.test2 <- data.all2[ind2==i,1]
  mod2 <- lm(dt.train2[,1]~dt.train2[,2]+I(dt.train2[,2]^2)+I(dt.train2[,2]^3)+dt.train2[,3]+dt.train2[,4])
  x.test.olah2 <- cbind(1,x.test2[,1],x.test2[,1]^2,x.test2[,1]^3,x.test2[,2],x.test2[,3])
  beta2 <- coefficients(mod2)
  prediksi2 <- x.test.olah2%*%beta2
  res2 <- c(res2,mean((y.test2-prediksi2)^2))
}
res2
mean(res2)
```

```
> res2
[1] 22.32898 21.33868 29.40459 29.79913 25.22047
> mean(res2)
[1] 25.61837
```

Smoothing Spline

- Ide: mencari suatu fungsi $g(x)$ sedemikian sehingga fungsi $g(x)$ sangat pas (fit) dengan data yang ditunjukkan dengan jumlah kuadrat galat yang kecil, dan juga fungsi tersebut haruslah smooth
- Untuk menjamin fungsi $g(x)$ smooth, umumnya dipenuhi dengan mencari fungsi $g(x)$ yang meminimumkan

$$\sum_{i=1}^n (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

dengan $\lambda > 0$ adalah tuning parameter.

- Maka, fungsi $g(x)$ yang meminimumkan persamaan di atas disebut sebagai **smoothing spline**

$$\sum_{i=1}^n (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

Loss function
(untuk data fitting dengan fungsi g)

Turunan kedua thd fungsi g
(mendeteksi perubahan slope)

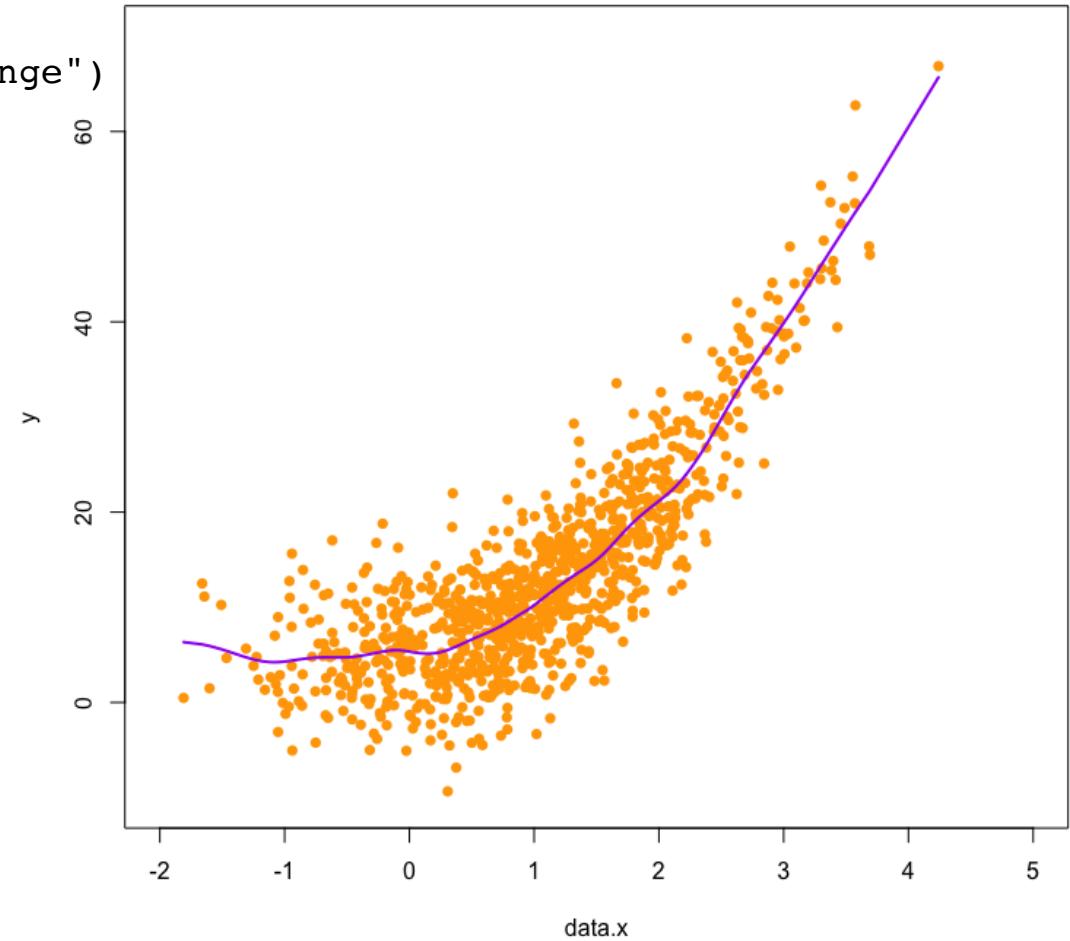
Penalty term
(mempenalti keragaman di g)

- Jika fungsi g sangat smooth, maka $\int g''(t)^2 dt$ memiliki nilai yang kecil
- Jika fungsi g sangat berfluktuatif, maka $\int g''(t)^2 dt$ memiliki nilai yang besar
- Akibatnya $\lambda \int g''(t)^2 dt$ memaksa fungsi g untuk selalu smooth
- Semakin besar λ , maka semakin smooth fungsi g

- Untuk menerapkan smoothing splines, kita tidak perlu menetapkan banyak atau lokasi dari knot, karena knot untuk metode ini terletak pada setiap observasi nilai x (x_1, x_2, \dots, x_n)
- Namun, kita perlu memilih nilai λ yang paling sesuai. Caranya dengan menggunakan cross-validation untuk meminimumkan jumlah kuadrat galat, atau dapat juga menggunakan LOOCV (leave-one-out cross validation).

Ilustrasi di R – smoothing splines

```
#SMOOTHING SPLINES  
ss1 <- smooth.spline(data.x,y,all.knots = T)  
plot(data.x,y,xlim=c(-2,5),ylim=c(-10,70),pch=16,col="orange")  
lines(ss1,col="purple",lwd=2)
```



Terima kasih 😊

Data untuk tugas kelompok

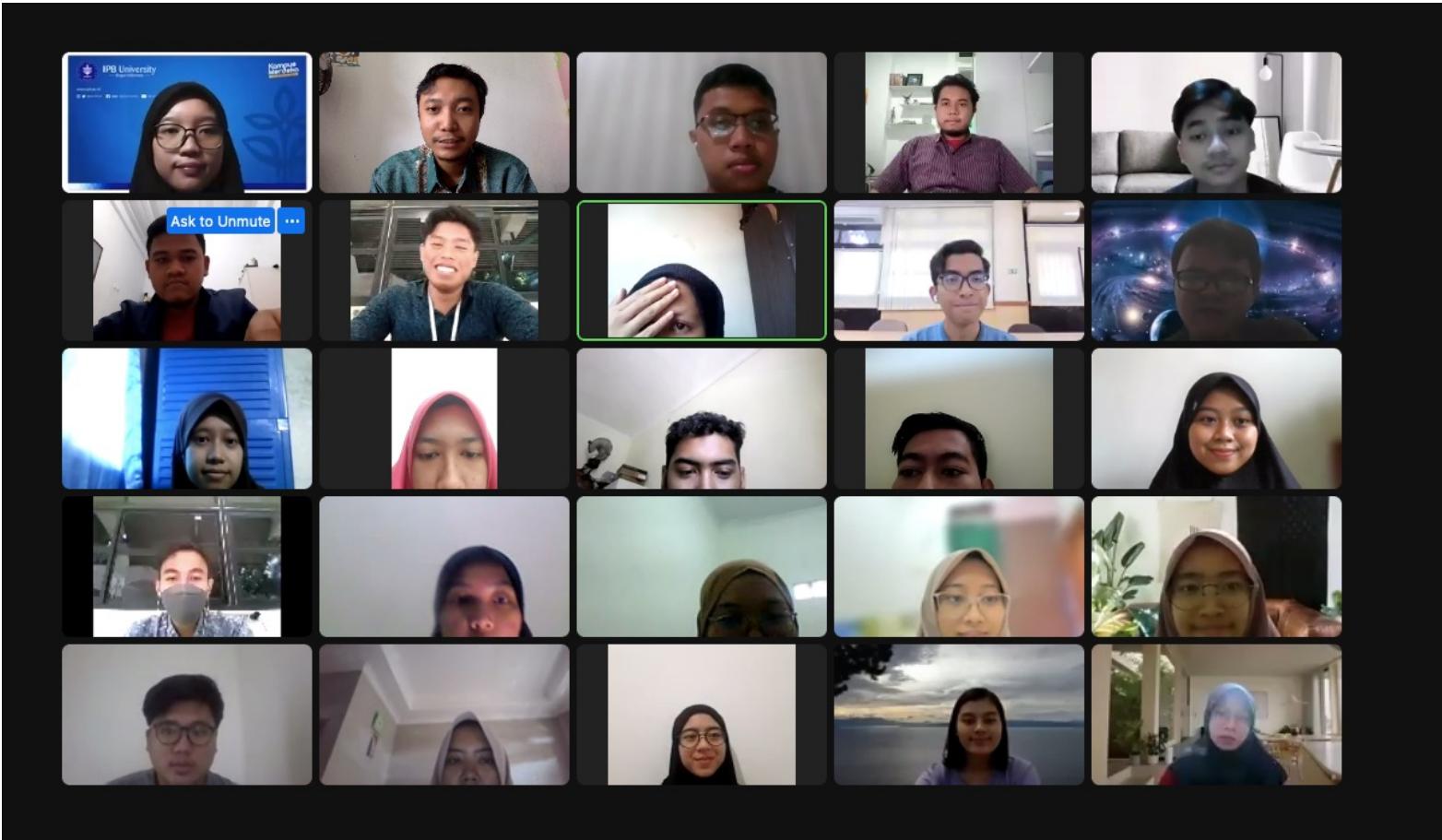
- Gunakan link
<https://www.bps.go.id/site/pilihdata>
- Pilihlah:
 - Minimal 4 peubah yang akan dianalisis (boleh terdapat 1 peubah respon dan minimal 3 predictor)
 - Gunakan satu titik waktu yakni pada suatu tahun tertentu (optional)
 - Gunakan semua peubah dengan level pengamatan dalam kabupaten
- Analisis (pilih):
 - Gunakan pemodelan regresi (dan atau pengembangannya) terhadap peubah-peubah yang dipilih untuk tujuan mencari peubah-peubah prediktor yang signifikan atau untuk tujuan prediksi
 - Gunakan metode unsupervised learning

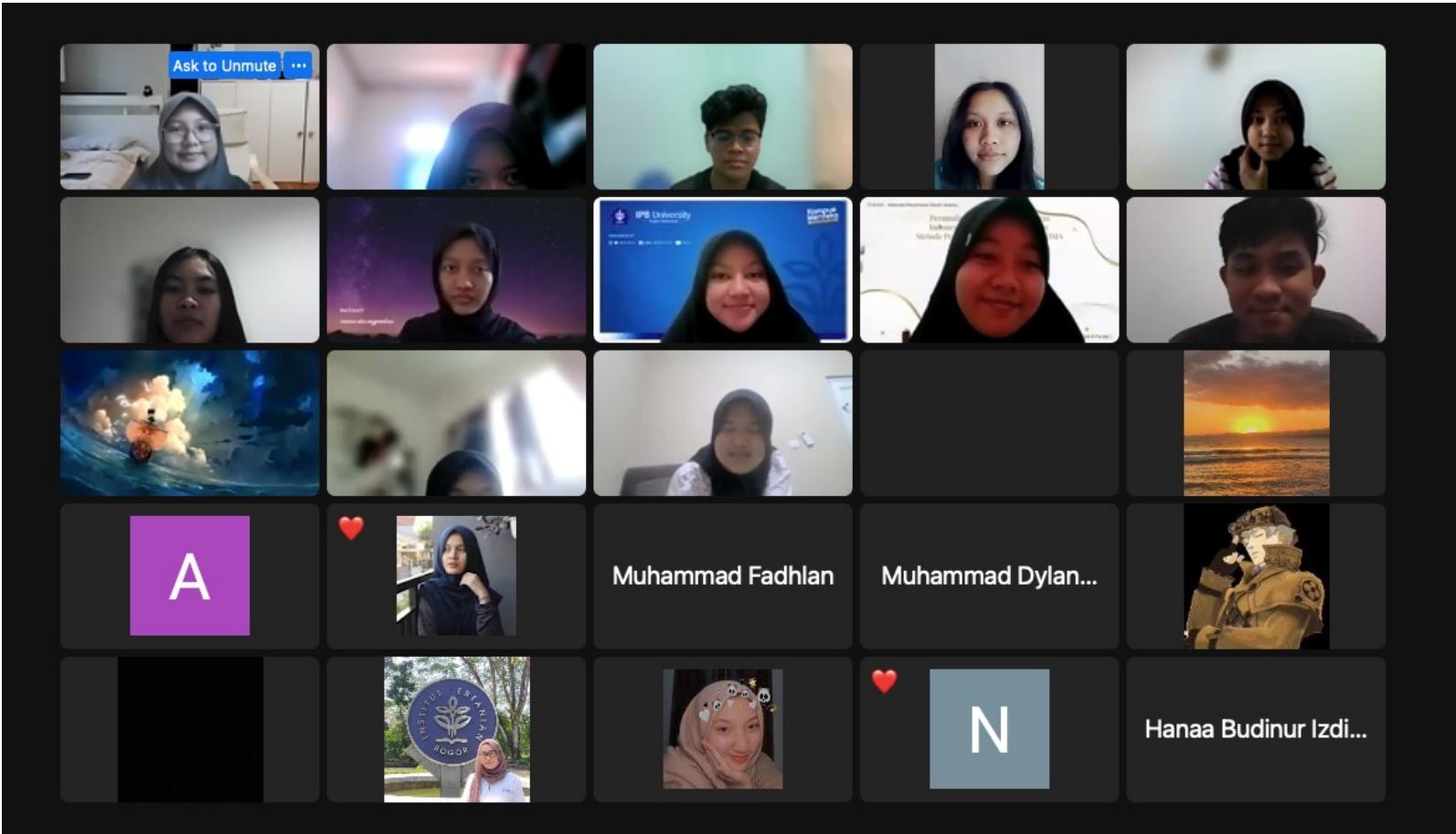
ISI POSTER: Latar Belakang, Metodologi (Data & Analisis Data), Hasil & Pembahasan, dan Kesimpulan

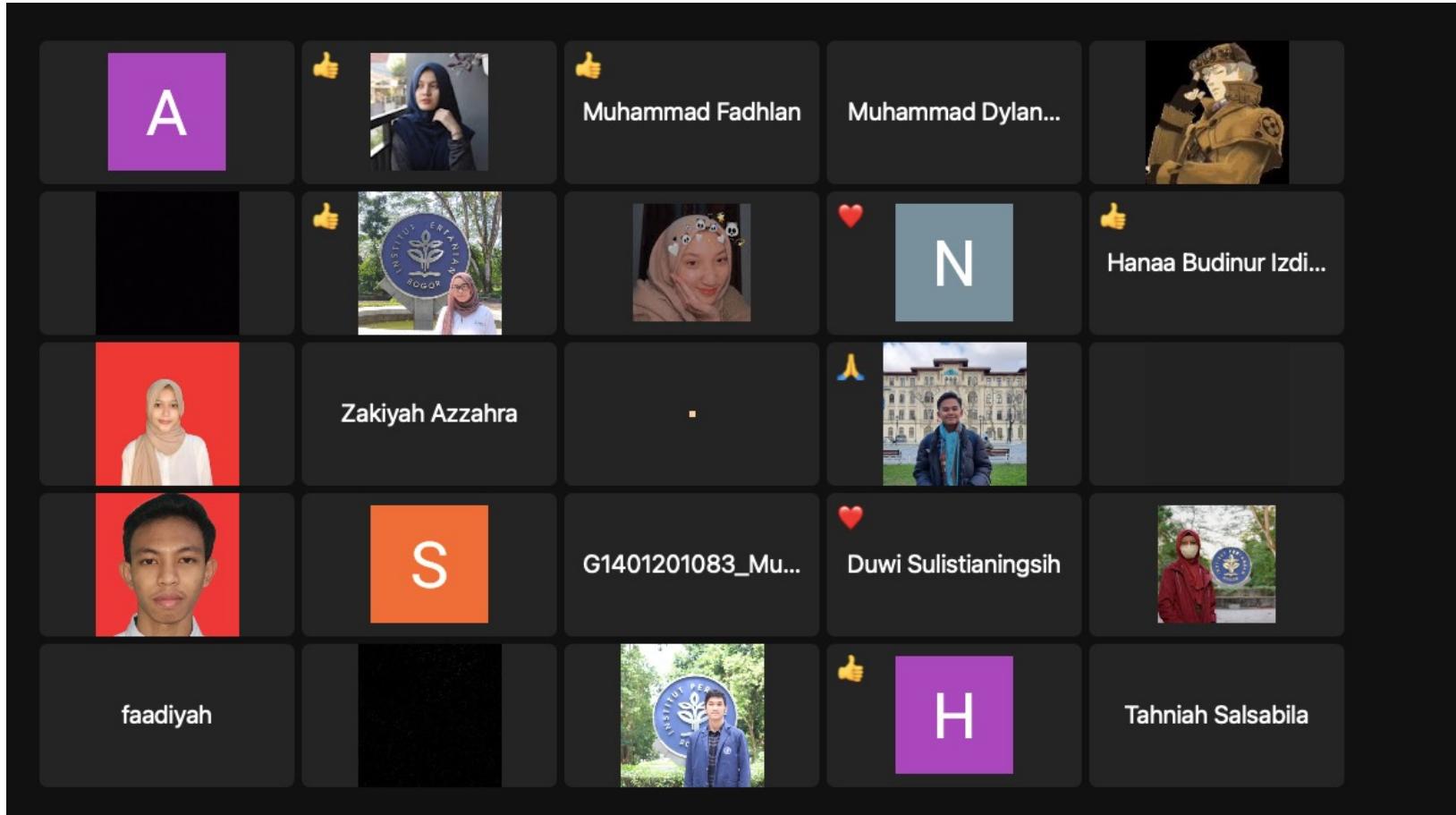
Deadline waktu pengumpulan: Rabu/ 30 Nov '22

Presentasi hasil poster pada pertemuan 14: Jumat/ 2 Des '22

Terima kasih 😊









Introduction to Machine Learning

Kuliah 9 – STA1381 Pengantar Sains
Data

Septian Rahardiantoro



Outline: Introduction to Machine Learning

- Pengantar
- Statistical Learning
 - Supervised vs unsupervised
- Beberapa metode supervised
 - Ridge Regression
 - Lasso Regression
 - Model Averaging

Pengantar

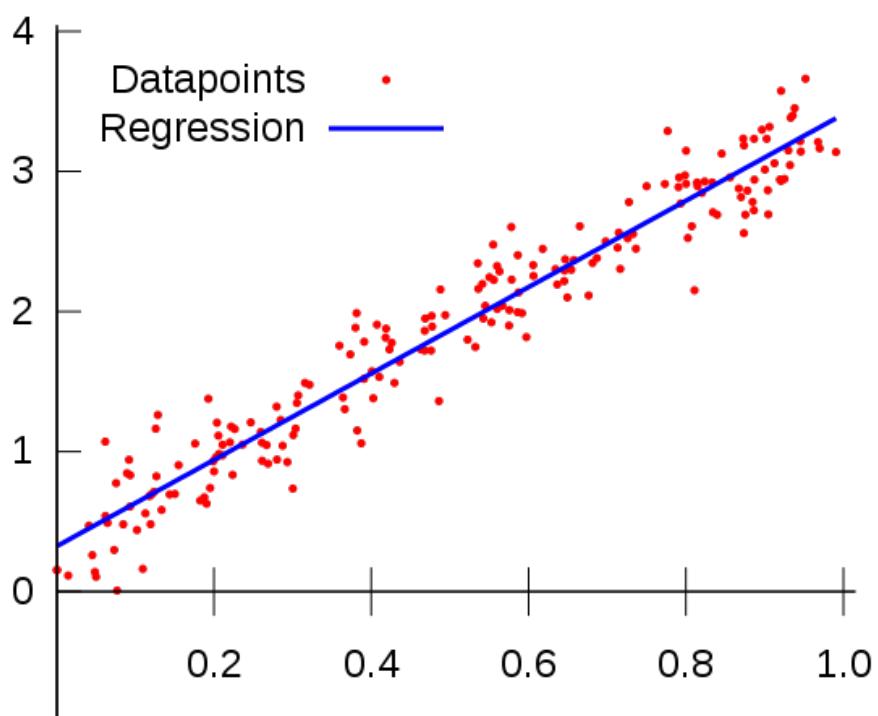
- Machine learning adalah subbidang kecerdasan buatan (AI: Artificial Intelligence).
- Tujuan machine learning secara umum adalah untuk memahami struktur data dan memasukkan data tersebut ke dalam model yang dapat dipahami dan dimanfaatkan oleh orang-orang.
- Meskipun machine learning adalah bidang dalam ilmu komputer, teknisnya seringkali menggunakan analisis statistik untuk menghasilkan nilai sesuai tujuan analisisnya.
- Karena itu, machine learning memfasilitasi komputer dalam membangun model dari data sampel untuk mengotomatisasi proses pengambilan keputusan berdasarkan input data.

Lalu apa perbedaan antara machine learning dengan statistika (statistics)?

		Statistics	Machine Learning
Subfield of...	Mathematics	Computer Science (AI)	
Focus on...	Building models with explicitly programmed instructions	Creating systems that learn from data	
Purpose	Inferences; Relationships between variables	Optimization; Prediction accuracy	
Prior assumptions about data	Some knowledge about population usually required	None	
Dimensionality of data	Usually applied to low-dimensional data	Usually applied to high dimensional data; ML learns from data	
Knowledge overlap	No ML knowledge required	Some stats knowledge usually needed; stats is basis for algorithms	

Ilustrasi 1

Statistical Models vs Machine learning – Linear Regression Example



Machine Learning:

- Tujuan machine learning, dalam hal ini, adalah untuk mendapatkan performa terbaik pada set pengujian.
- Sehingga prakteknya dibentuk **data training** untuk **pembentukan model**, dan **data testing** untuk **evaluasi model** tersebut

Statistical Models:

- Menemukan garis linear yang meminimalkan JKG, dengan asumsi Gaussian, dan **tidak ada data training dan testing yang diperlukan**.
- Tujuannya lebih untuk identifikasi peubah prediktor yang mempengaruhi peubah respon, meskipun juga dapat digunakan untuk prediksi

Ilustrasi 2

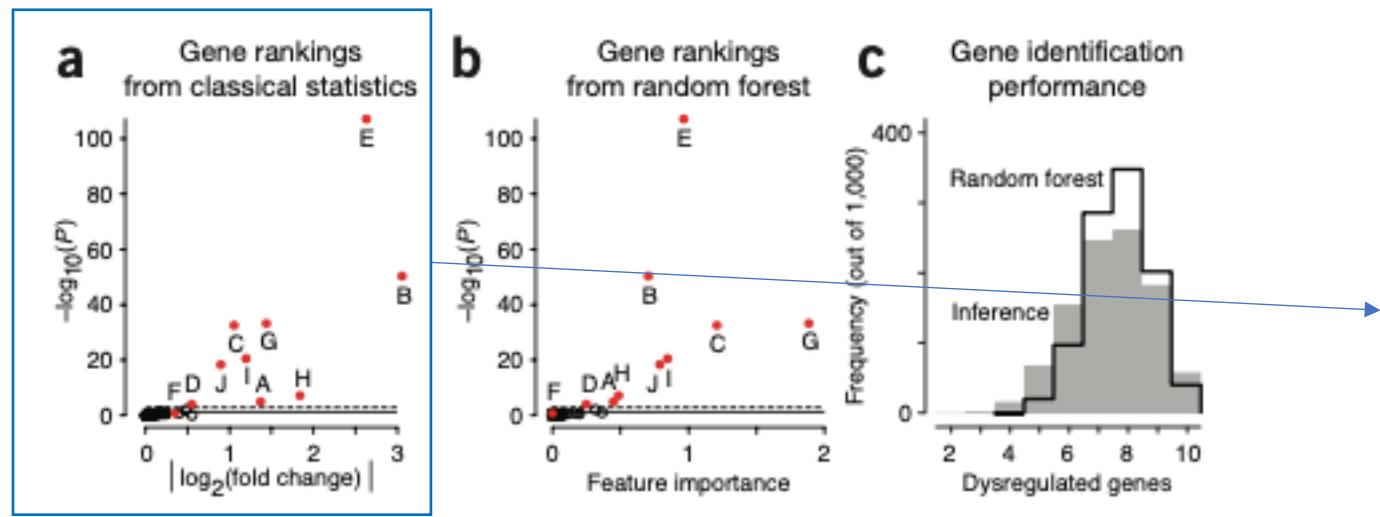


Figure 2 | Analysis of gene ranking by classical inference and ML.

- (a) Unadjusted log-scaled P values from statistical differential expression analysis as a function of effect size, measured by fold change in expression.
- (b) Log-scaled P values from a as a function of gene importance from random forest classification. In a and b, red circles identify the ten differentially expressed genes from **Figure 1**; the remaining genes are indicated by open circles. (c) Distribution of the number of dysregulated genes correctly identified in 1,000 simulations by inference (gray fill) and random forest (black line).

menunjukkan p-value dari uji antar fenotipe sebagai fungsi dari perubahan log fold dalam ekspresi gen. Sepuluh gen disregulasi ditampilkan dengan titik berwarna merah; Hasilnya: diperoleh sembilan dari sepuluh (kecuali F, dengan perubahan log fold terkecil) sebagai gen yang signifikan dengan p-value $<0,05$.

Ilustrasi 2

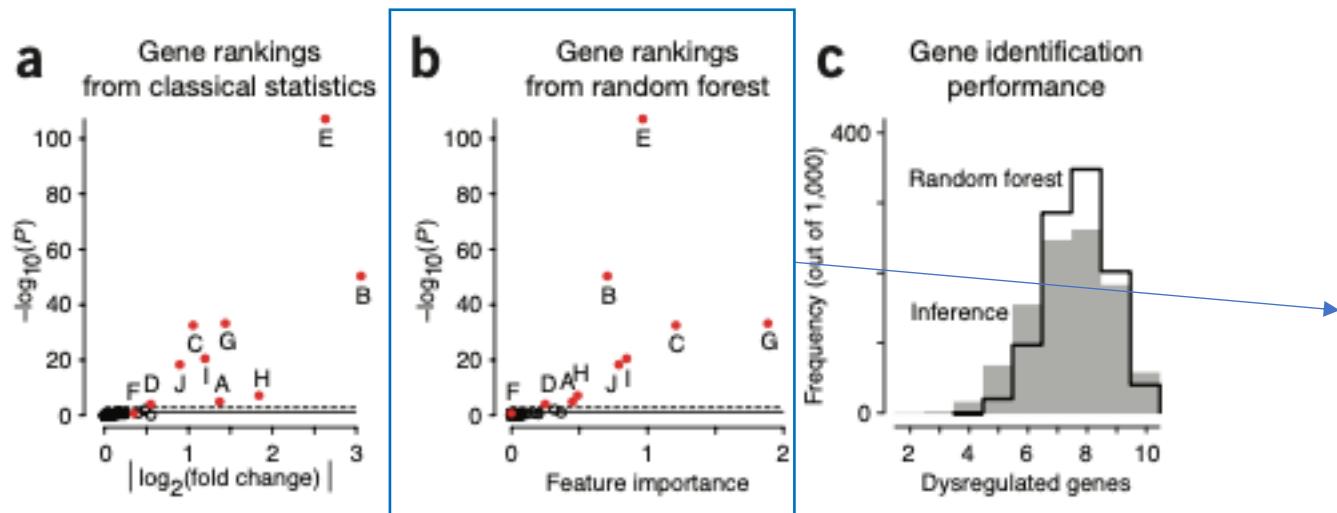


Figure 2 | Analysis of gene ranking by classical inference and ML.

- (a) Unadjusted log-scaled P values from statistical differential expression analysis as a function of effect size, measured by fold change in expression.
- (b) Log-scaled P values from a as a function of gene importance from random forest classification. In a and b, red circles identify the ten differentially expressed genes from **Figure 1**; the remaining genes are indicated by open circles. (c) Distribution of the number of dysregulated genes correctly identified in 1,000 simulations by inference (gray fill) and random forest (black line).

hasil klasifikasi RF dengan 100 pohon, dimana p-value dari inferensi klasik (hasil a) diplot sebagai fungsi feature importance (gen). Skor ini mengkuantifikasi kontribusi gen tertentu terhadap peningkatan klasifikasi rata-rata dalam sebuah partisi ketika pohon dipisah memilih gen itu.

Ilustrasi 2

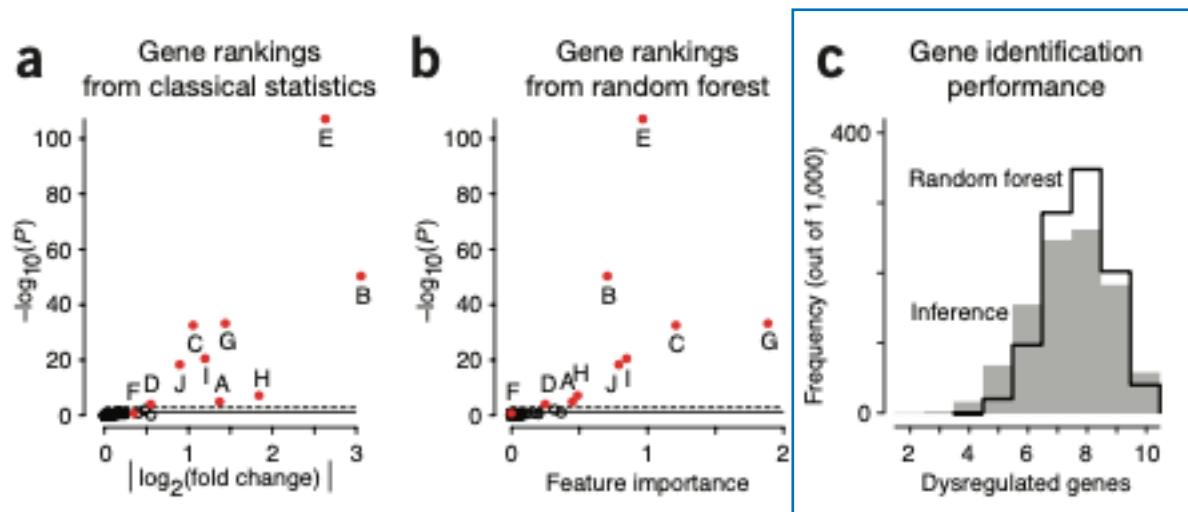


Figure 2 | Analysis of gene ranking by classical inference and ML.

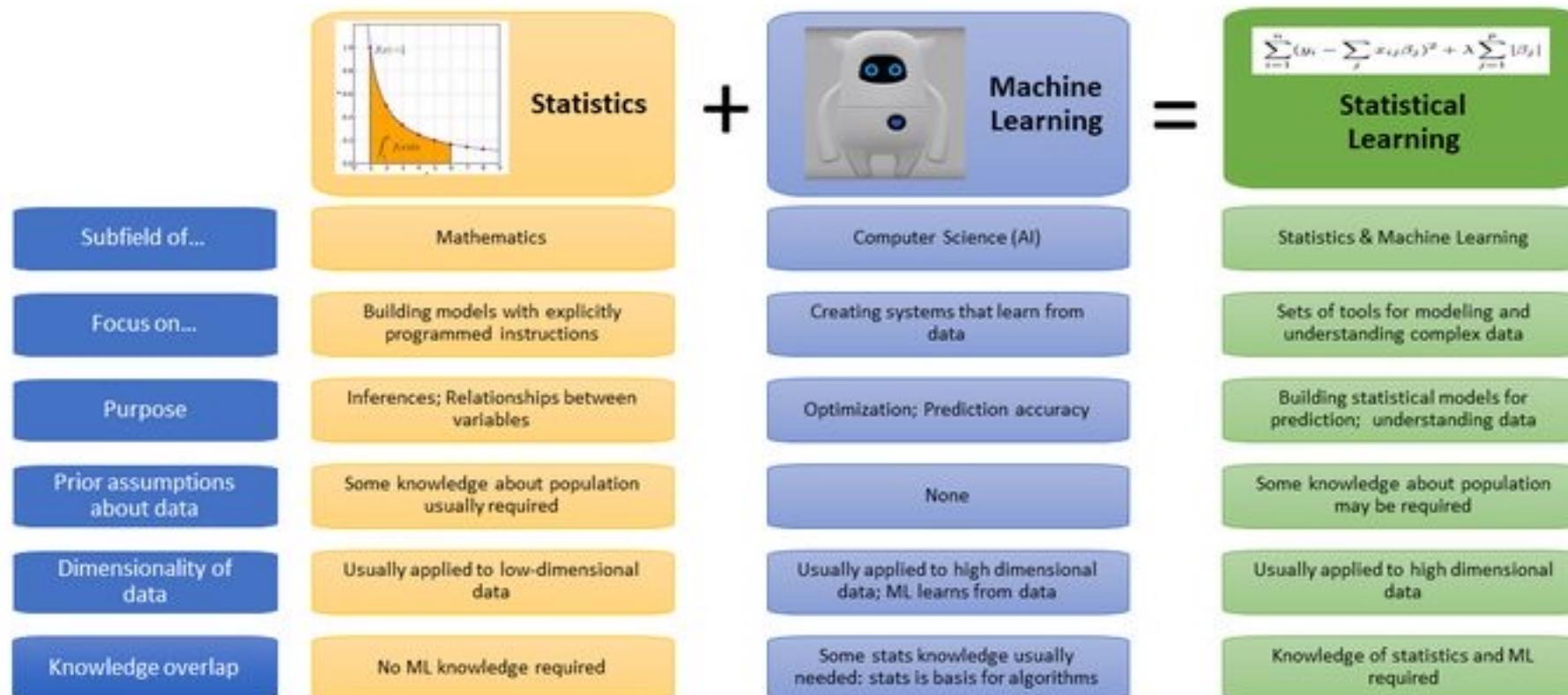
(a) Unadjusted log-scaled P values from statistical differential expression analysis as a function of effect size, measured by fold change in expression. (b) Log-scaled P values from a as a function of gene importance from random forest classification. In a and b, red circles identify the ten differentially expressed genes from **Figure 1**; the remaining genes are indicated by open circles. (c) Distribution of the number of dysregulated genes correctly identified in 1,000 simulations by inference (gray fill) and random forest (black line).

jika kita melakukan simulasi 1.000 kali dan menghitung jumlah gen disregulasi yang diidentifikasi dengan benar oleh kedua pendekatan (a dan b)—berdasarkan uji hipotesis klasik atau generalisasi pola prediktif dengan RF dan sepuluh peringkat feature importance— maka diperoleh bahwa kedua metode menghasilkan hasil yang serupa.

Jumlah rata-rata gen disregulasi yang diidentifikasi adalah 7,4/10 untuk inferensi klasik dan 7,7/10 untuk RF

- Machine learning adalah semua tentang hasil. Namun, pemodelan statistik lebih tentang menemukan hubungan antara variabel dan signifikansi hubungan tersebut, dan dapat juga untuk menghasilkan prediksi.
- Statistics menarik kesimpulan populasi dari sampel, sedangkan machine learning menemukan pola prediktif yang dapat digeneralisasikan.

Statistics + Machine Learning = Statistical Learning



Musio image: Akawikipic [CC BY-SA 4.0 (<https://creativecommons.org/licenses/by-sa/4.0/>)]

<https://www.datasciencecentral.com/wp-content/uploads/2021/10/3541473617.jpg>

Statistical Learning

- Pembelajaran statistik mengacu pada seperangkat alat yang luas untuk memahami data.
- Alat-alat ini dapat diklasifikasikan sebagai: **supervised learning** dan **unsupervised learning**.
- Secara umum, statistical supervised learning melibatkan pembangunan model statistik untuk memprediksi, atau memperkirakan *outputs* berdasarkan satu atau lebih *inputs*. Masalah alam ini terjadi di berbagai bidang seperti bisnis, kedokteran, astrofisika, dan kebijakan publik.
- Dengan statistical unsupervised learning, ada *inputs* tetapi tidak ada *supervising outputs*; namun kita dapat mempelajari hubungan dan struktur dari data tersebut.

Beberapa Metode

Classical Machine Learning

Task Driven

Supervised Learning

(Pre Categorized Data)



Classification

(Divide the socks by Color)

Eg. Identity Fraud Detection



Regression

(Divide the Ties by Length)

Eg. Market Forecasting

Data Driven

Unsupervised Learning

(Unlabelled Data)



Clustering

(Divide by Similarity)

Eg. Targeted Marketing



Association

(Identify Sequences)

Eg. Customer Recommendation



Dimensionality Reduction

(Wider Dependencies)

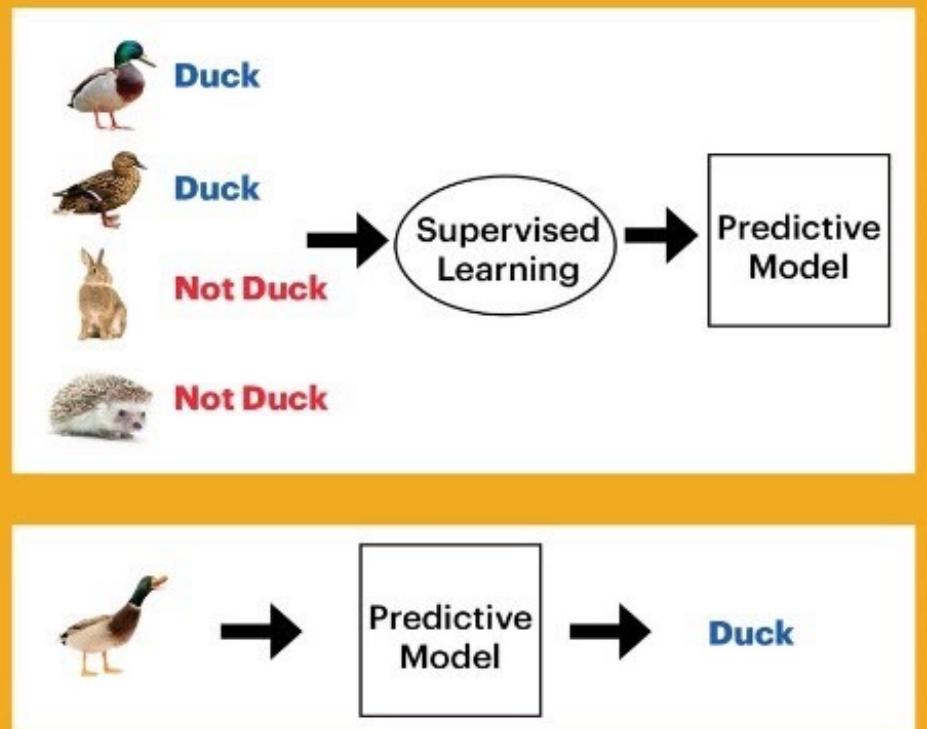
Eg. Big Data Visualization

Obj: Predictions & Predictive Models

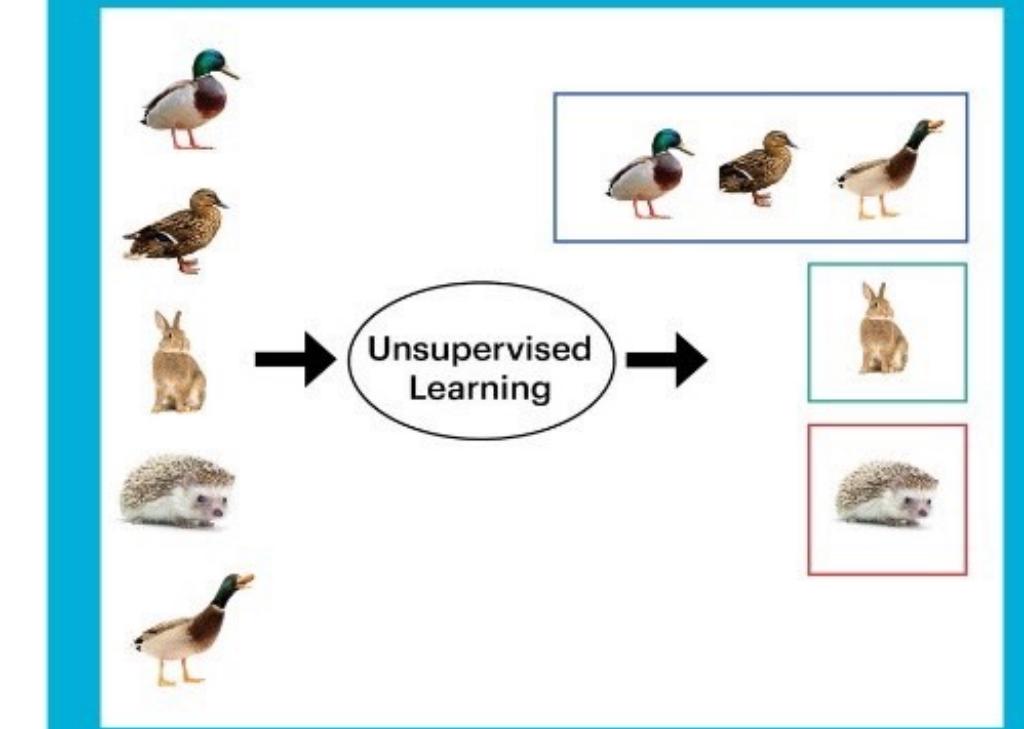
Pattern/ Structure Recognition



Supervised Learning (Classification Algorithm)



Unsupervised Learning (Clustering Algorithm)



Western Digital.

https://miro.medium.com/max/1111/0*4q_X_xQxevOQY_0u

Beberapa Metode: Supervised Learning

- Supervised Learning:
 - Misalkan diketahui peubah respon Y dengan skala numerik dan beberapa peubah prediktor X_1, X_2, \dots, X_p .
 - Ingin diketahui:
 1. Peubah prediktor mana yang mempengaruhi perubahan rata-rata peubah respon
 2. Prediksi peubah respon
 - Dengan ukuran data yang cukup besar, seringkali metode-metode klasik tidak cocok untuk digunakan.
 - Terdapat metode dalam statistical learning yang dapat digunakan:
 - Ridge regression
 - Lasso Regression
 - Model Averaging

Ridge Regression

Ridge regression (Hoerl & Kennard 1988)

→ meminimalkan jumlah kuadrat galat yang terikat pada regularisasi L_2 dari koefisiennya.

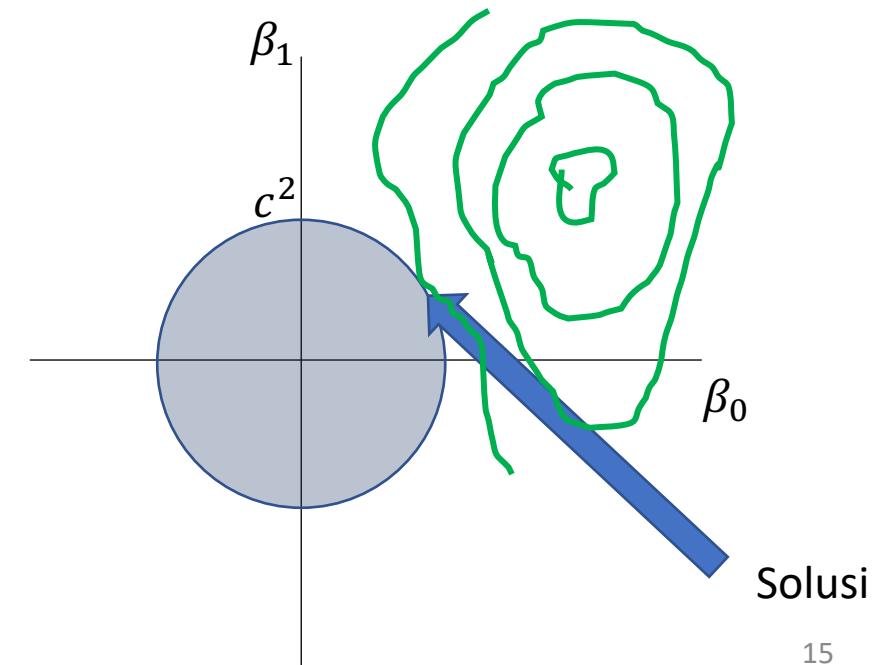
$$\begin{aligned}\hat{\beta}^{Ridge} &= \arg \min_{\beta} \{ \|y - X\beta\|_2^2 + \lambda \|\beta\|_2^2 \} \\ &= \arg \min_{\beta} \{ (y - X\beta)^T (y - X\beta) + \lambda \beta^T \beta \}\end{aligned}$$

→ $\hat{\beta}^{Ridge} = (X^T X + \lambda I)^{-1} X^T y$

Pada kasus dengan parameter β_0 and β_1

$$\hat{\beta}^{Ridge} = \arg \min_{\beta} \|y - X\beta\|_2^2 \text{ s.t. } \|\beta\|_2^2 \leq c^2$$

$$\beta_0^2 + \beta_1^2 \leq c^2$$



Kegunaan Ridge Regression

- Menduga koefisien regresi dengan peubah prediktornya saling berkorelasi (multikolinieritas tinggi)
- Digunakan pada pemodelan regresi dengan peubah prediktor sangat banyak (bahkan $p \gg n$)

Sifat dugaan parameternya:

- Dugaan parameternya berbias
- Model ridge selalu mempertahankan semua prediktornya

Lasso Regression

(Tibshirani 1996)

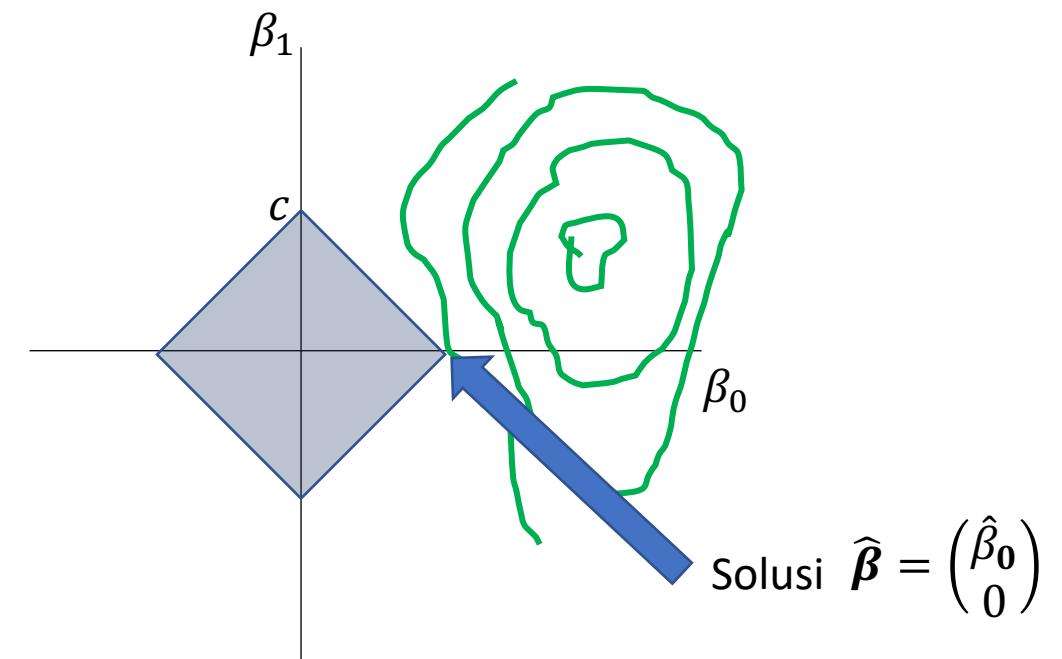
→ lasso melakukan penyusutan berkelanjutan (shrinkage) dan pemilihan variabel otomatis secara bersamaan

$$\hat{\beta}^{lasso} = \arg \min_{\beta} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \quad \longrightarrow \quad \|\alpha\|_1 = \sum_i |a_i|$$

→ *least angle regression (LARS) algorithm*

Pada kasus dengan parameter β_0 and β_1

$$\hat{\beta}^{lasso} = \arg \min_{\beta} \|y - X\beta\|_2^2 \text{ s.t. } \|\beta\|_1 \leq c$$
$$|\beta_0| + |\beta_1| \leq c$$



Kegunaan Lasso Regression

- Lasso berguna dalam seleksi peubah prediktor dalam model
- Digunakan pada pemodelan regresi dengan peubah prediktor sangat banyak (bahkan $p \gg n$)

Sifat dugaan parameternya:

- Dugaan parameternya berbias
- Pada model lasso memungkinkan tidak semua peubah prediktor dipilih (memiliki koefisien tidak sama dengan 0)

Model Averaging

- Membangun beberapa kandidat model untuk dikombinasikan menjadi sebuah model final
- Umumnya model averaging diterapkan dalam rangka untuk memperoleh nilai prediksi peubah respon

High dimensional regression data ($p \gg n$)

Y	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	...	X_p
y_1	x_{11}	x_{21}	x_{31}	x_{41}	x_{51}	x_{61}	x_{71}	x_{81}		x_{p1}
y_2	x_{12}	x_{22}	x_{32}	x_{42}	x_{52}	x_{62}	x_{72}	x_{82}		x_{p2}
:										
y_n	x_{1n}	x_{2n}	x_{3n}	x_{4n}	x_{5n}	x_{6n}	x_{7n}	x_{8n}		x_{pn}

Model Candidate Construction

$$1 \quad Y \sim X_1 + X_3 + X_6 \longrightarrow \hat{Y}_1$$

$$2 \quad Y \sim X_3 + X_4 + X_8 \longrightarrow \hat{Y}_2$$

:

$$k \quad Y \sim X_2 + X_6 + X_p \longrightarrow \hat{Y}_k$$

Model Averaging

$$\hat{Y} = \frac{\sum_{i=1}^k w_i \hat{Y}_i}{\sum_{i=1}^k w_i}$$

Ilustrasi di R

```
set.seed(123)
x <- cbind(1,matrix(rnorm(100*20,2,1),100,20))
e <- matrix(rnorm(100),100,1)
b <- c(1,rep(0:4,each=4))
y <- x%*%b+e

dt.all <- data.frame(y,x[,-1])
str(dt.all)

#regresi linier
mod1 <- lm(y~.,data=dt.all)
coef(mod1)

library(glmnet)
#regresi ridge
mod2 <- cv.glmnet(x[,-1],y,alpha=0) #pemilihan lambda dgn cv untuk ridge
mod2
coef(mod2,s="lambda.min")

#lasso
mod3 <- cv.glmnet(x[,-1],y,alpha=1) #pemilihan lambda dgn cv untuk lambda
mod3
coef(mod3,s="lambda.min")

#model averaging
library(MuMIn)
mod1 <- lm(y~.,data=dt.all, na.action = na.fail)
mod4 <- dredge(global.model=mod1,m.lim=c(18,20))
mod5 <- model.avg(mod4,delta<4)
mod5
summary(mod5)
```

Terima kasih 😊

Variable Selection

Kuliah 11 – STA1381

Pengantar Sains Data

Septian Rahardiantoro



Outline

- Pengantar
- Beberapa metode dalam seleksi peubah (Variable Selection)
- Subset Selection
 - Best subset selection
 - Stepwise selection
 - Pemilihan model optimal

Pengantar

- Misalkan dalam model linear

$$Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \varepsilon$$

yang umumnya digunakan untuk menggambarkan hubungan antara peubah respon Y dan sekumpulan peubah prediktor X_1, X_2, \dots, X_p .

- Umumnya, model linier ini dapat diduga dengan metode kuadrat terkecil (OLS).
- Dugaan model dengan OLS berprinsip untuk meminimumkan jumlah kuadrat galat.

Matriks $\varepsilon = Y - \hat{X}\beta$
 $Y = X\beta + \varepsilon \rightarrow MKT$

$\min (\varepsilon' \varepsilon)$
 $\rightarrow \hat{\beta} = (X'X)^{-1} X' Y$

- Sifat penduga OLS:
 - Nilai dugaan parameternya bersifat tidak berbias $E(\hat{\beta}) = \beta$
 - Nilai dugaan parameternya memiliki ragam terkecil jika dibandingkan dengan penduga lainnya $\text{Var}(\hat{\beta}) \rightarrow \underline{\text{min}}$
- Oleh karenanya, penduga OLS pada model regresi linier merupakan penduga parameter tak bias terbaik.

$$y = \underbrace{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}_{OLS \rightarrow \hat{\beta}_0 \dots \hat{\beta}_p} + \epsilon$$

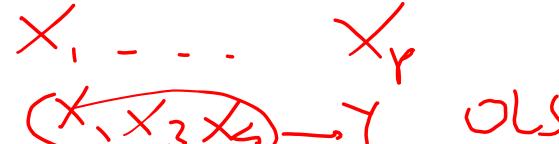
- Namun, banyak orang yang beralih tidak menggunakan OLS dalam pendugaan parameter untuk pemodelan regresi linier.
- Beberapa alasan:
 - **Keakuratan Prediksi:** penduga OLS tidak lagi bersifat tak bias terbaik ketika $p \gg n$, bahkan OLS tidak dapat digunakan sama sekali \rightarrow gagal dalam konteks prediksi.
 - **Kemudahan Interpretasi Model:** penduga OLS akan menghasilkan dugaan setiap parameter di dalam model, meskipun pada peubah prediktor yang tidak berhubungan dengan respon. Adanya dugaan yang tidak relevan tersebut, akan membuat interpretasi model lebih kompleks (rumit). ✓

- Dalam rangka meningkatkan kemudahan dalam interpretasi modelnya, metode yang digunakan adalah: *feature selection* atau *variable selection*
 - *feature selection* atau *variable selection* → menyisihkan peubah prediktor yang tidak relevan pada model regresi linier berganda

Beberapa metode dalam seleksi peubah (Variable Selection)

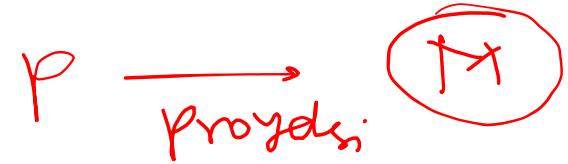
1. Subset Selection

Pendekatan ini melibatkan pengidentifikasi subset dari p buah prediktor yang diyakini terkait dengan respon. Kemudian, dugaan modelnya diperoleh menggunakan OLS pada sebagian prediktor yang telah dipilih.



2. Shrinkage

Pendekatan ini melibatkan pembuatan model yang melibatkan semua p buah prediktor. Namun, koefisien ini diduga dengan penyusutan menuju nol. Penyusutan ini (juga dikenal sebagai regularisasi) memiliki efek mengurangi keragaman. Oleh karena itu, metode penyusutan juga dapat melakukan pemilihan peubah prediktor.



3. Dimension Reduction → PCA

Pendekatan ini memproyeksikan p buah prediktor ke dalam subruang M -dimensi, di mana $M < p$. Hal ini diperoleh dengan menghitung M buah kombinasi linier yang berbeda, atau proyeksi, dari variabel. Kemudian proyeksi M ini digunakan sebagai prediktor agar sesuai dengan model regresi linier dengan OLS.

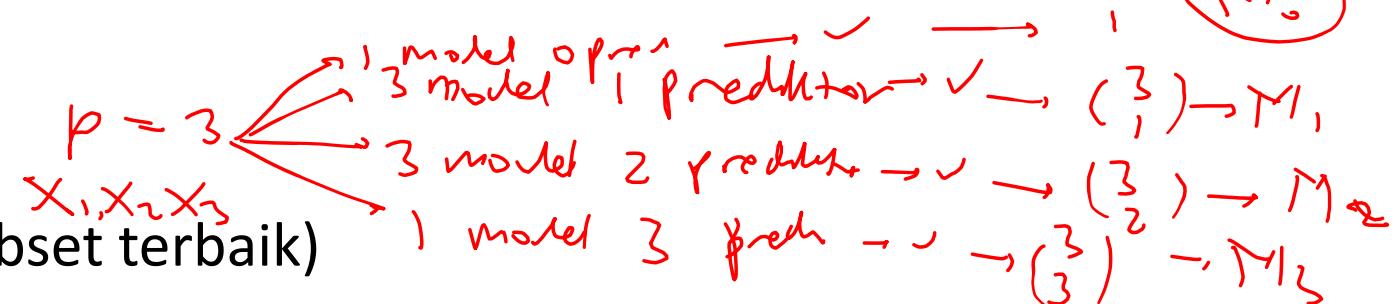
Subset Selection

1. Best Subset Selection (pemilihan subset terbaik)

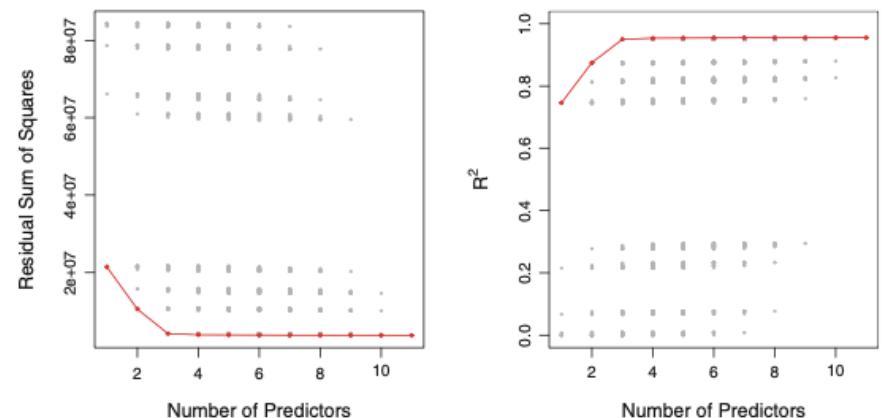
- Untuk melakukan pemilihan subset terbaik, dilakukan OLS yang terpisah untuk setiap kemungkinan kombinasi dari p prediktor.
- Yaitu, membangun p buah model yang berisi sebuah prediktor, membangun $\binom{p}{2} = \frac{p(p-1)}{2}$ model yang berisi tepat dua prediktor, dan seterusnya sampai ada model yang berisi semua prediktor
- Kemudian, dipilihlah model terbaiknya

Algorithm 6.1 Best subset selection

- Let \mathcal{M}_0 denote the *null model*, which contains no predictors. This model simply predicts the sample mean for each observation.
 - For $k = 1, 2, \dots, p$:
 - Fit all $\binom{p}{k}$ models that contain exactly k predictors.
 - Pick the best among these $\binom{p}{k}$ models, and call it \mathcal{M}_k . Here *best* is defined as having the smallest RSS, or equivalently largest R^2 .
 - Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-



Untuk perbandingan $p + 1$ model ini tidak disarankan menggunakan RSS dan R^2



- Pemilihan subset terbaik adalah pendekatan yang sederhana dan menarik secara konseptual, namun sangat mahal dalam komputasinya.

$$P = 3 \rightarrow 2^P - 2^S = 8$$

- Jumlah model yang mungkin yang harus dipertimbangkan, meningkat dengan cepat sejalan dengan meningkatnya p . Secara umum, ada sebanyak 2^p model yang melibatkan himpunan bagian dari p prediktor.

$$2^{10}$$

- Akibatnya, dengan $p = 10$, maka akan ada 1000 kemungkinan model yang akan dipertimbangkan. Dan $p = 20$, akan ada lebih dari 1 juta kemungkinan model.
- Untuk alasan komputasi, pemilihan subset terbaik tidak dapat diterapkan dengan p yang sangat besar. Pemilihan subset terbaik juga dapat mengalami masalah statistik ketika p besar. Semakin besar ruang pencarian, semakin tinggi peluang untuk menemukan model yang terlihat bagus pada data pelatihan, meskipun model tersebut mungkin tidak memiliki kekuatan prediktif pada data masa depan. Jadi ruang pencarian yang sangat besar dapat menyebabkan overfitting dan ragam yang tinggi dari dugaan koefisien.

2. Stepwise Selection

a. Forward Stepwise Selection

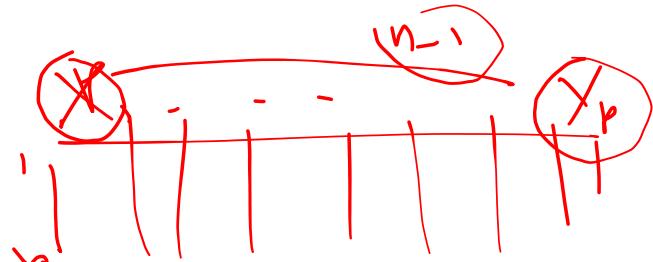
- Forward Stepwise Selection dimulai dengan model yang tidak mengandung prediktor, dan kemudian menambahkan prediktor ke model, satu per satu, sampai semua prediktor berada dalam model.
- Secara khusus, pada setiap langkah, variabel yang memberikan peningkatan terbesar pada kebaikan modellah yang ditambahkan ke model.

$$\begin{array}{l} p=3 \\ \text{① } M_0 : \text{tidak ada } X \\ \text{② } \leftarrow \begin{array}{c} X_1 \\ X_2 \\ X_3 \end{array} \right\} M_1 = X_1 \\ \text{③ } X_1 \leftarrow \begin{array}{c} X_1 \\ X_2 \end{array} \right\} M_2 = X_1 + X_2 \\ \text{④ } X_1 + X_2 \leftarrow X_3 - M_3 \end{array}$$

Algorithm 6.2 Forward stepwise selection

- Let M_0 denote the *null* model, which contains no predictors.
 - For $k = 0, \dots, p - 1$:
 - Consider all $p - k$ models that augment the predictors in M_k with one additional predictor.
 - Choose the *best* among these $p - k$ models, and call it M_{k+1} . Here *best* is defined as having smallest RSS or highest R^2 .
 - Select a single best model from among M_0, \dots, M_p using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

- Tidak seperti pemilihan subset terbaik, yang melibatkan identifikasi sebanyak 2^p model, Forward Stepwise Selection melibatkan pembentukan model yang dalam iterasi ke- k , untuk $k = 0, \dots, p - 1$, yang berjumlah total $1 + \sum_{k=0}^{p-1} (p - k) = 1 + p(p + 1)/2$ model. $p=3 \rightarrow 5$ model $2^{10} = 4096$
- Ketika $p = 20$, pemilihan subset terbaik akan mempertimbangkan sebanyak $1,048,576$ models, namun dengan Forward Stepwise Selection hanya membentuk model sebanyak 211. $p=10 \rightarrow 56$ model
- Namun, Forward Stepwise Selection tidak selalu menjamin memperoleh model yang paling baik. Seperti contoh, diketahui $p = 3$, dan model terbaik untuk $p = 1$ adalah model dengan X_1 , dan model terbaik untuk $p = 2$ adalah model dengan X_2 dan X_3 . Apabila dengan menggunakan Forward Stepwise Selection jika \mathcal{M}_1 di dalamnya terdapat X_1 , maka pada \mathcal{M}_2 juga di dalamnya terdapat X_1 .
- Forward Stepwise Selection dapat diterapkan bahkan dalam data berdimensi tinggi di mana $n < p$; namun, dalam kasus ini, dimungkinkan untuk membangun submodel $\mathcal{M}_0, \dots, \mathcal{M}_{n-1}$ saja, karena setiap submodel diduga menggunakan OLS, yang tidak akan menghasilkan solusi unik jika $p > n$.



$$p = 10 \rightarrow 49$$

$p = 50$

b. Backward Stepwise Selection

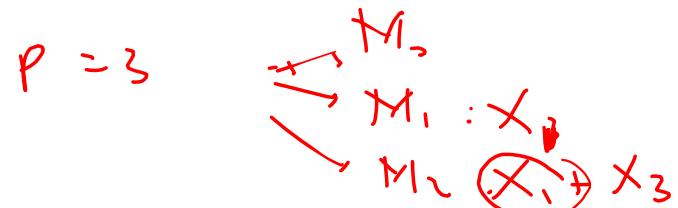
- Dimulai dengan model penuh yang berisi semua p prediktor, dan kemudian secara iteratif menghilangkan prediktor yang paling tidak berguna, satu per satu.

$$\begin{array}{l} P = 3 \\ \cancel{x_1} \cancel{x_2} \cancel{x_3} \\ \textcircled{1} M_3 = x_1, x_2, x_3 \\ \textcircled{2} \{x_1, x_2\} M_2 = x_1, x_2 \\ \textcircled{3} \{x_1, x_2, x_3\} M_1 = x_3 \\ \textcircled{4} M_0 = \text{tak ada } x \end{array}$$

Algorithm 6.3 Backward stepwise selection

- Let \mathcal{M}_p denote the *full* model, which contains all p predictors.
 - For $k = p, p - 1, \dots, 1$:
 - Consider all k models that contain all but one of the predictors in \mathcal{M}_k , for a total of $k - 1$ predictors.
 - Choose the *best* among these k models, and call it \mathcal{M}_{k-1} . Here *best* is defined as having smallest RSS or highest R^2 .
 - Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, C_p (AIC), BIC, or adjusted R^2 .
-

- Seperti Forward Stepwise Selection, pendekatan Backward Stepwise Selection hanya mencari sebanyak $1 + p(p + 1)/2$ model dan dengan demikian dapat diterapkan dalam data dengan p terlalu besar untuk menerapkan pemilihan subset terbaik.
- Juga seperti Forward Stepwise Selection, Backward Stepwise Selection tidak dijamin untuk menghasilkan model terbaik yang berisi subset dari p prediktor.
 $n > p$ $(r > n) \times$.
- Backward Stepwise Selection mensyaratkan bahwa jumlah sampel n lebih besar dari jumlah variabel p . Sebaliknya, Forward Stepwise Selection dapat digunakan bahkan ketika $n < p$, dan merupakan satu-satunya metode subset yang layak ketika p sangat besar.



c. Metode Hybrid

- Pendekatan seleksi subset terbaik, Forward Stepwise Selection, dan Backward Stepwise Selection umumnya memberikan model yang serupa tetapi tidak identik.
- Sebagai alternatif lain, tersedia versi hybrid dari Forward Stepwise Selection dan Backward Stepwise Selection, di mana variabel ditambahkan ke model secara berurutan, dalam analogi Forward Stepwise Selection. Namun, setelah menambahkan setiap variabel baru, metode tersebut juga dapat menghapus variabel apa pun yang tidak lagi memberikan peningkatan pada kebaikan model. Pendekatan semacam itu mencoba untuk lebih meniru pemilihan subset terbaik sambil mempertahankan keunggulan komputasi Forward Stepwise Selection dan Backward Stepwise Selection.

3. Pemilihan Model Optimal

a. Model optimal dipilih berdasarkan nilai C_p , AIC , BIC , dan $Adjusted R^2$

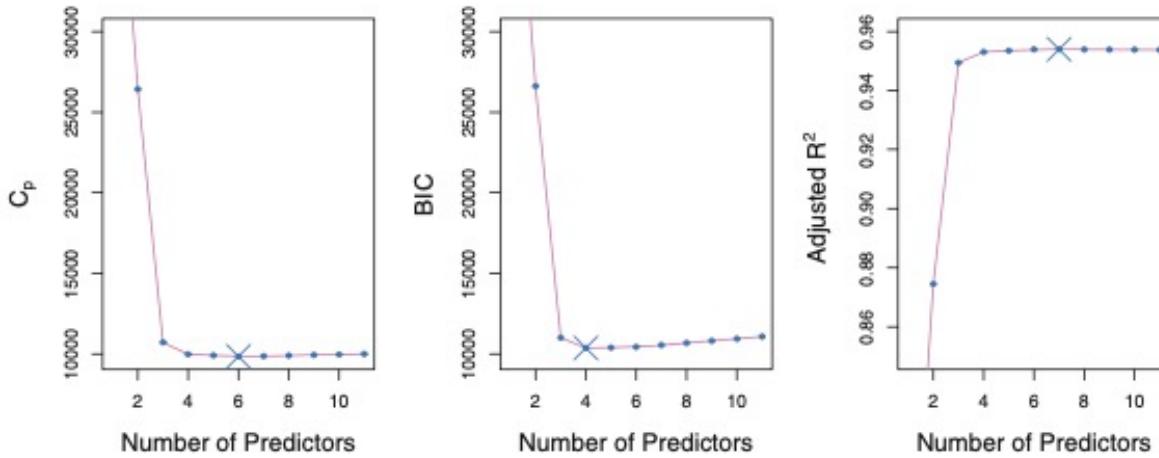
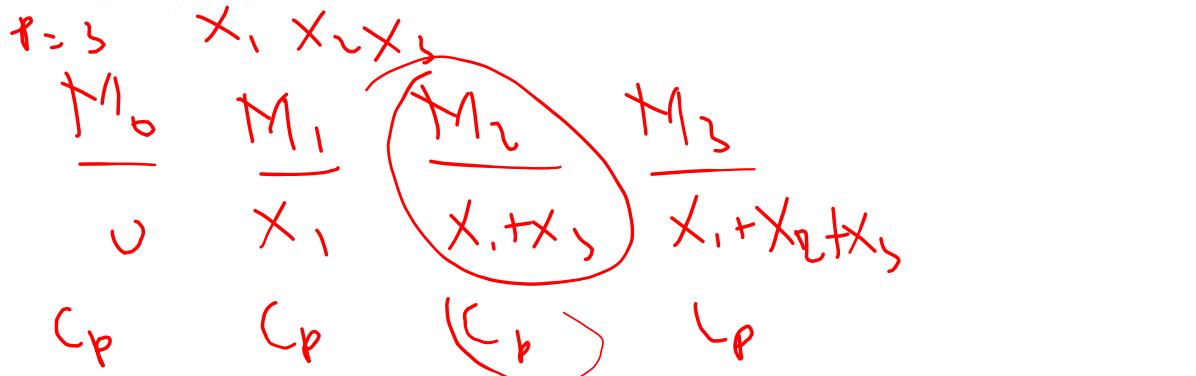


FIGURE 6.2. C_p , BIC , and adjusted R^2 are shown for the best models of each size for the Credit data set (the lower frontier in Figure 6.1). C_p and BIC are estimates of test MSE. In the middle plot we see that the BIC estimate of test error shows an increase after four variables are selected. The other two plots are rather flat after four variables are included.



$$C_p = \frac{1}{n} (RSS + 2d\hat{\sigma}^2)$$

C_p penduga tak bias bagi MSE → akibatnya memilih model dengan C_p terkecil

$$AIC = \frac{1}{n\hat{\sigma}^2} (RSS + 2d\hat{\sigma}^2)$$

C_p dan AIC sebanding satu sama lain

$$BIC = \frac{1}{n} (RSS + \log(n)d\hat{\sigma}^2)$$

→ memilih model dengan BIC terkecil

$$Adjusted R^2 = 1 - \frac{RSS/(n - d - 1)}{TSS/(n - 1)}$$

→ memilih model dengan $Adjusted R^2$ terbesar

b. Model optimal dipilih berdasarkan Validasi dan Validasi Silang

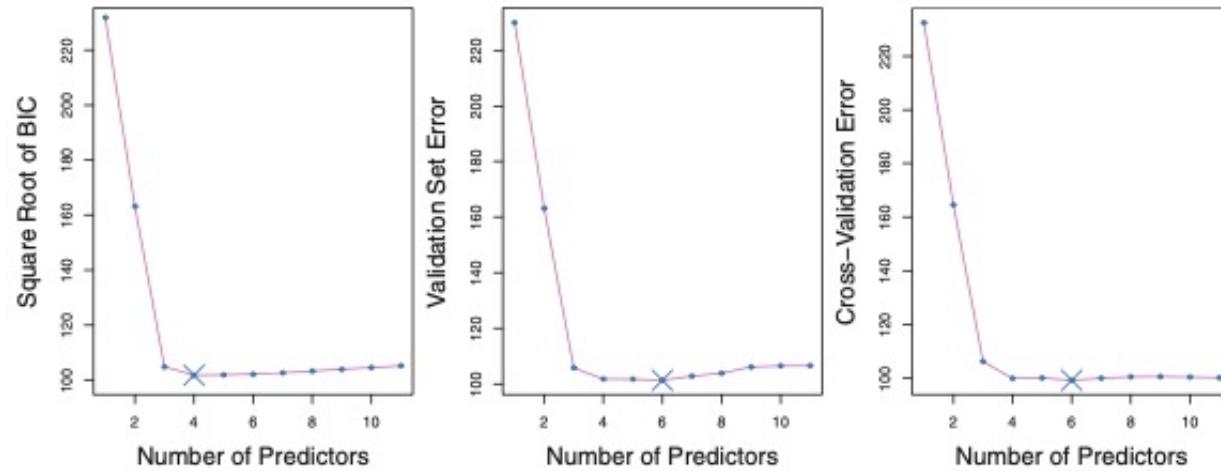


FIGURE 6.3. For the Credit data set, three quantities are displayed for the best model containing d predictors, for d ranging from 1 to 11. The overall best model, based on each of these quantities, is shown as a blue cross. Left: Square root of BIC. Center: Validation set errors. Right: Cross-validation errors.

- Kita dapat langsung memperkirakan kesalahan pengujian menggunakan metode validasi dan validasi silang.
- Kita dapat menghitung kesalahan validasi atau kesalahan validasi silang untuk setiap model yang dipertimbangkan, lalu pilih model yang menghasilkan estimasi kesalahan pengujian terkecil.
- Prosedur ini memiliki keuntungan relatif terhadap AIC , BIC , C_p , dan $Adjusted R^2$, karena memberikan perkiraan langsung dari kesalahan pengujian, dan membuat lebih sedikit asumsi tentang model dasar yang sebenarnya, serta dapat digunakan pada kasus seleksi model yang lebih luas.

$$p = 3$$

$$x_1, x_2, x_3$$

→ forward stepwise selection

① $M_0: \beta_0$

② $M_1 \leftarrow \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \right\} \text{backward} \rightarrow M_1 = x_1$

③ $x_1 + \left\langle \begin{array}{c} x_2 \\ x_3 \end{array} \right\rangle \left\{ \begin{array}{c} x_1 + x_2 \\ x_1 + x_3 \end{array} \right\} \text{backward} \rightarrow M_2 = x_1 + x_3$

④ $M_3 = x_1 + x_2 + x_3$

$M_0 = \text{base } X$
 $M_1 = x_1$
 $\boxed{M_2 = x_1 + x_3}$
 $M_3 = x_1 + x_2 + x_3$
 $C_p, BIC, AIC - R^2$
5 fold CV

Aplikasi di R

```
install.packages("ISLR")
library(ISLR)
fix(Hitters)
names(Hitters)
dim(Hitters)
sum(is.na(Hitters$Salary)) #ada 59 pemain yg gajinya missing

Hitters=na.omit(Hitters)
dim(Hitters)
sum(is.na(Hitters$Salary))

library(leaps)
regfit.full=regsubsets(Salary~.,Hitters)
summary(regfit.full)
regfit.full=regsubsets(Salary~.,data=Hitters ,nvmax=19)
reg.summary=summary(regfit.full)
reg.summary

reg.summary$adjr2 #model11
which.max(reg.summary$adjr2)
plot(reg.summary$adjr2 ,xlab="Number of Variables " ,
     ylab="Adjusted RSq",type="l")
points(11,reg.summary$adjr2[11], col="red",cex=2,pch=20)
#dan seterusnya

#Forward
regfit.fwd=regsubsets(Salary~.,data=Hitters ,nvmax=19, method ="forward")
summary(regfit.fwd)
#Backward
regfit.bwd=regsubsets (Salary~.,data=Hitters ,nvmax=19,method ="backward")
summary(regfit.bwd)

coef(regfit.full,7)
coef(regfit.fwd,7)
coef(regfit.bwd,7)
```



Departemen Statistika
Fakultas Matematika dan Pengetahuan Alam
IPB University

PROGRAM STUDI SARJANA, MAGISTER, DOKTOR

STATISTIKA DAN SAINS DATA

KULIAH UMUM

Pengembangan Profesi Sains Data

PEMATERI



Ir. Hedi M. Idris, M.Sc., Ph.D

Kepala Pusat Pengembangan Profesi dan Sertifikasi
Kementerian Komunikasi dan Informatika RI

MODERATOR



La Ode Abdul Rahman, M.Si

Dosen PS Statistika dan Sains Data
IPB University



Jumat, 11 Nov 2022
13.30 - 15.30 WIB



LIVE ON:
Departement of Statistics,
IPB University



AUDITORIUM FMIPA
IPB University
Dramaga - Bogor



statistikaipb



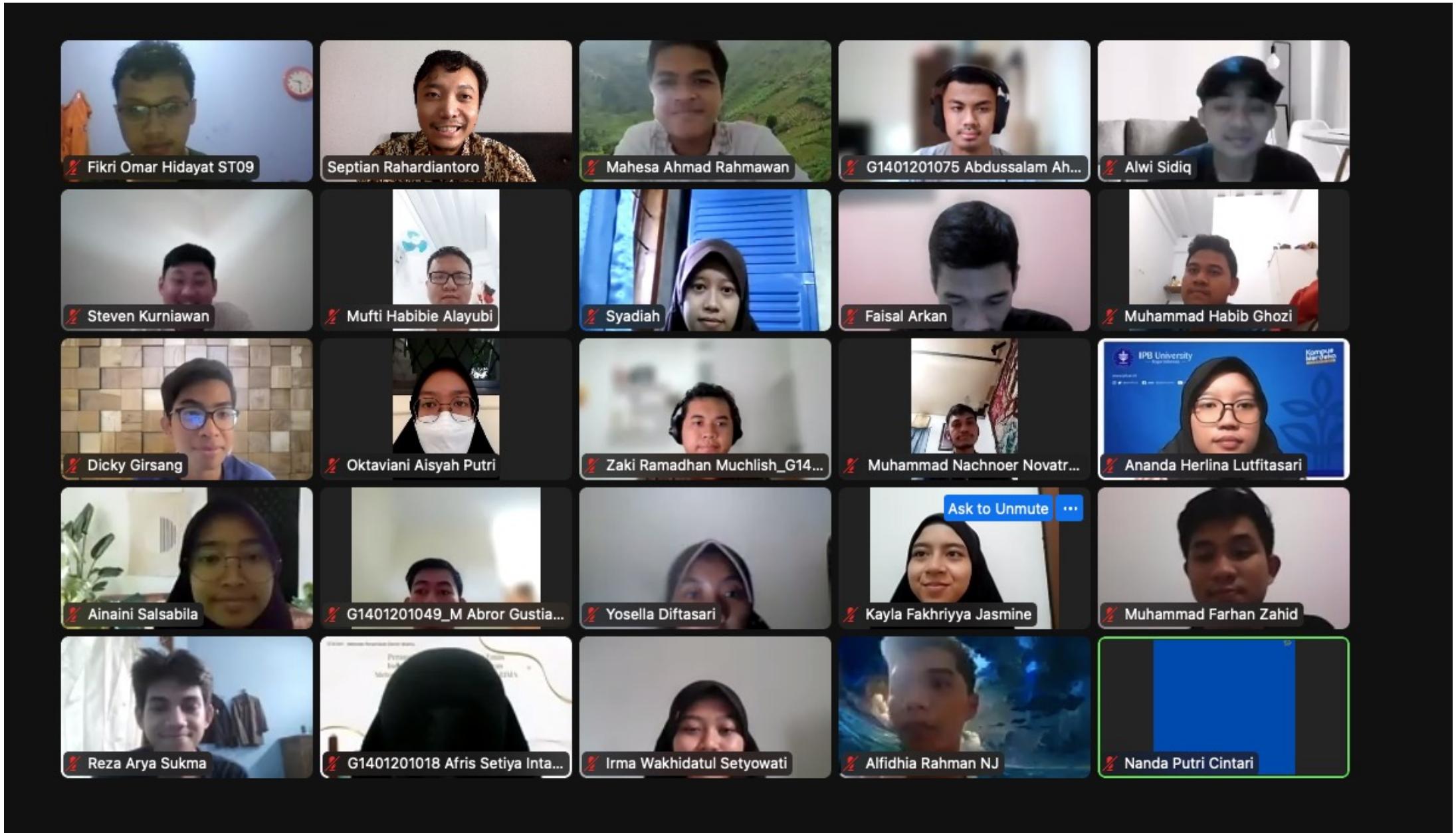
stkipb

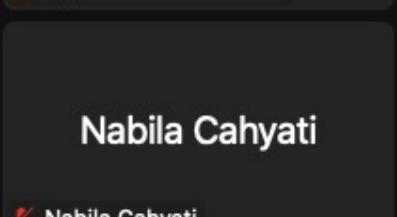
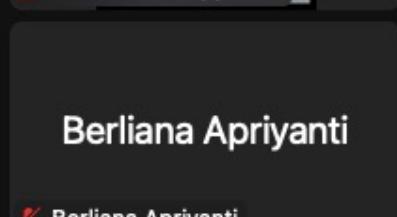
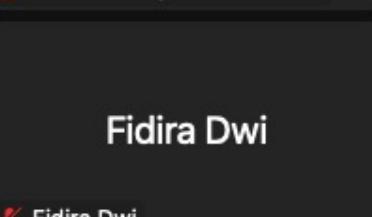
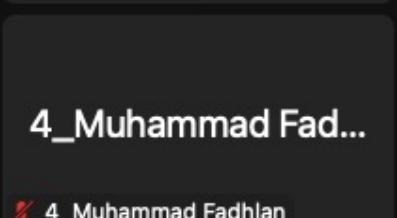
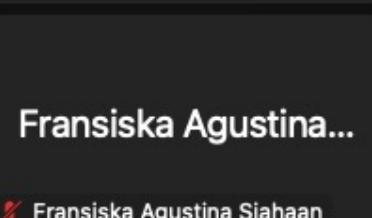
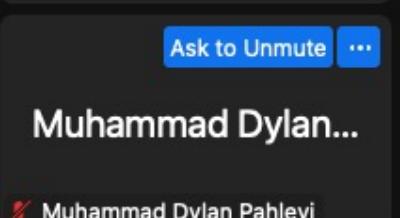
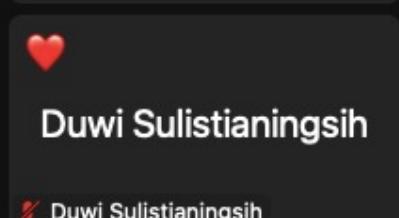
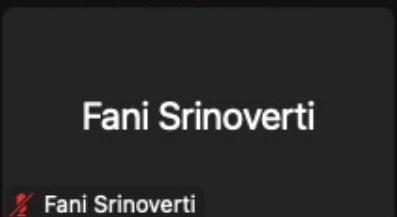
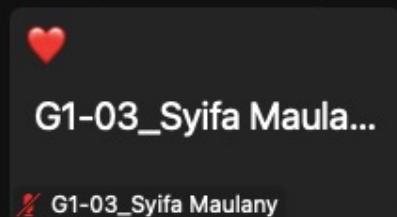
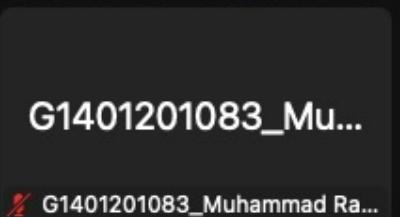
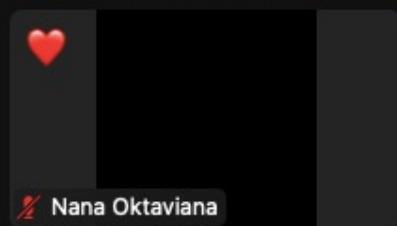
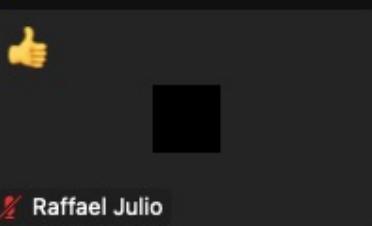


statistika@apps.ipb.ac.id



stat.ipb.ac.id







Maulana Ahsan Fa...

Maulana Ahsan Fadillah_G140...



G1401201083_Mu...

G1401201083_Muhammad Ra...

Faadiyah Ramadhani

Faadiyah Ramadhani

G1-03_Syifa Maula...

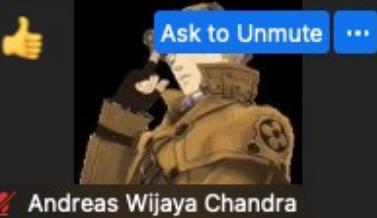
G1-03_Syifa Maulany

Fani Srinoverti

Fani Srinoverti

Duwi Sulistianingsih

Duwi Sulistianingsih



Muhammad Dylan...

Muhammad Dylan Pahlevi



Fransiska Agustina...

Fransiska Agustina Siahaan



4_Muhammad Fad...

4_Muhammad Fadlan



Fidira Dwi

Fidira Dwi

Berliana Apriyanti

Berliana Apriyanti



Febrian Adhitya Ca...

Febrian Adhitya Cahya Belardi



Akmal Riza Wibiso...

Akmal Riza Wibisono



Agsyhan Muhammad Sayidar



Arsyfia Chairunnisa



G1-39_Siti Aisyah

G1-39_Siti Aisyah

Aprilia Permata Putri



.

Dhea Puspita Adinda

Rosy Rosita

Rosy Rosita



Tantri Gustina Dewi



Naura Tirza

Terima kasih 😊



Unsupervised Learning: Cluster Analysis

Kuliah 12 – STA1381

Pengantar Sains Data

Septian Rahardiantoro

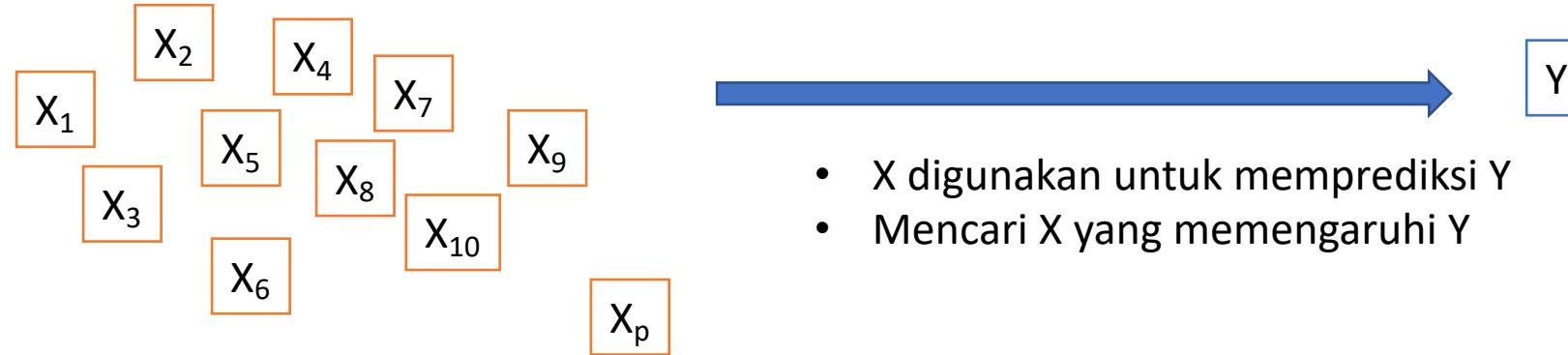


Outline

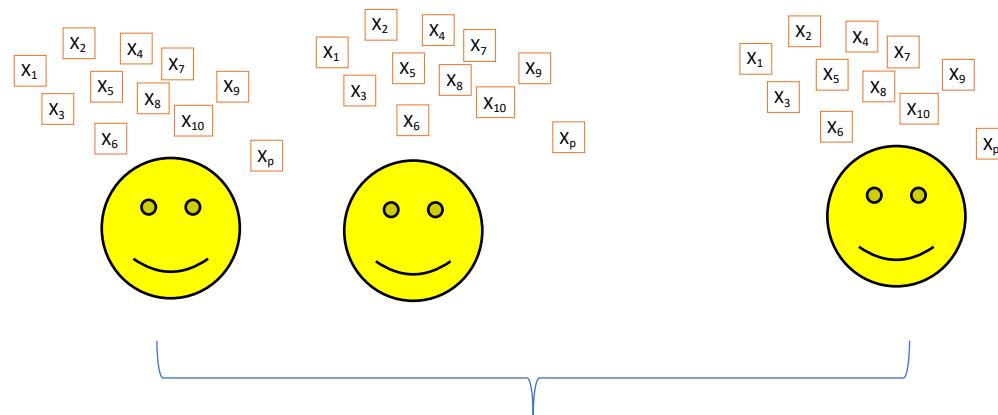
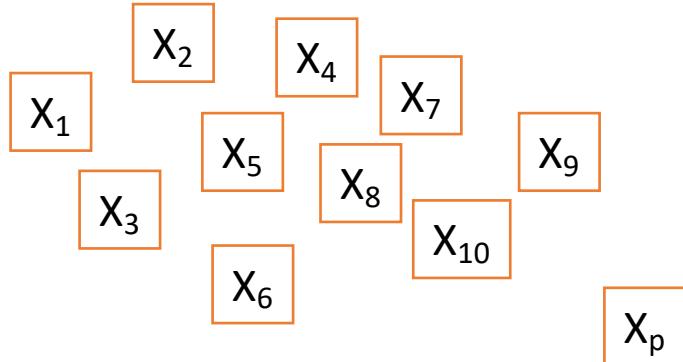
- Pengantar
- Analisis Gerombol (Cluster Analysis)
 - Analisis Gerombol tak berhierarki (Non-hierarchical Clustering)
 - Analisis Gerombol berhierarki (Hierarchical Clustering)
- Isu Praktis dalam Analisis Gerombol

Pengantar

Supervised learning



Unsupervised learning



- Mengelompokkan objek berdasarkan kesamaan karakteristik X

Analisis Gerombol (Cluster Analysis)

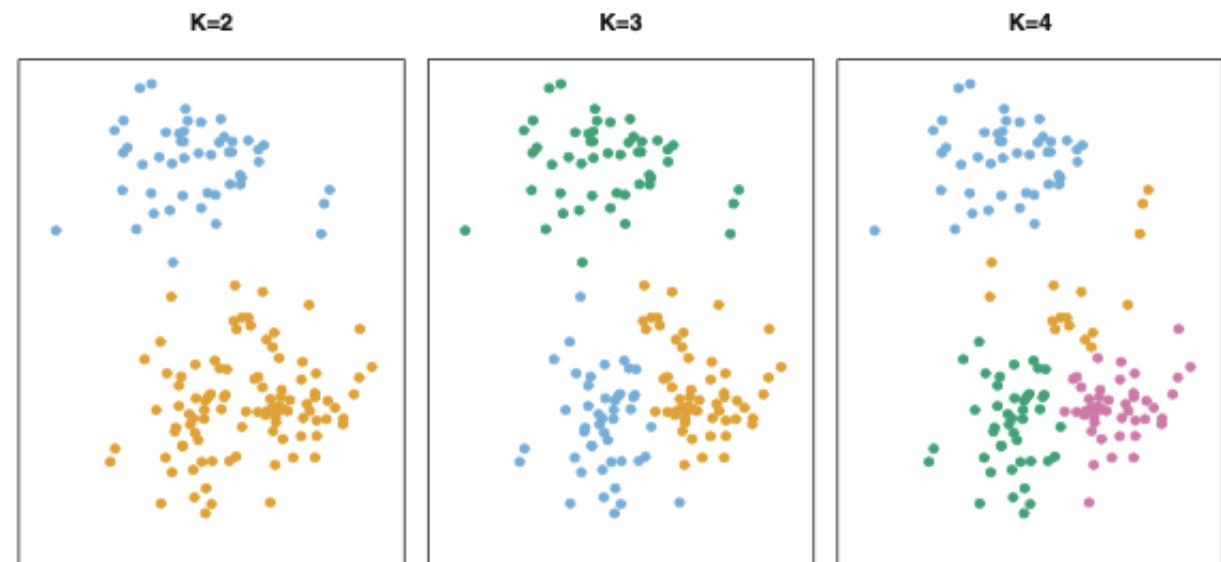
- Analisis gerombol bertujuan untuk menemukan subkelompok yang homogen di antara pengamatan.
- Misalkan kita mungkin memiliki akses ke sejumlah besar pengukuran (mis. pendapatan rumah tangga rata-rata, pekerjaan, jarak dari daerah perkotaan terdekat, dan sebagainya) untuk sejumlah besar pengelompokan orang.
- Sasarannya adalah melakukan segmentasi pasar dengan mengidentifikasi subkelompok orang yang mungkin lebih mudah menerima bentuk iklan tertentu, atau lebih cenderung membeli produk tertentu. Tugas melakukan segmentasi pasar sama dengan mengelompokkan orang-orang dalam kumpulan data (aplikasi analisis gerombol).
- Pada kuliah ini, kita fokus pada dua pendekatan metode penggerombolan yang paling terkenal:
 - Analisis Gerombol tak berhierarki (Non-hierarchical Clustering): K -means
 - Analisis Gerombol berhierarki (Hierarchical Clustering)

- Dalam metode K -means, kita berusaha untuk mempartisi pengamatan menjadi sejumlah pengelompokan yang telah ditentukan sebelumnya (sebanyak K).
- Di sisi lain, dalam analisis gerombol berhierarkhi, kita tidak mengetahui sebelumnya berapa banyak gerombol yang diinginkan; pada kenyataannya, dimanfaatkan representasi visual pengamatan seperti pohon, yang disebut dendrogram, yang memungkinkan untuk melihat sekaligus pengelompokan yang diperoleh untuk setiap kemungkinan jumlah gerombol, dari 1 hingga n . Ada kelebihan dan kekurangan masing-masing pendekatan pengelompokan ini, yang nanti akan dibahas pada kuliah ini.

K -means

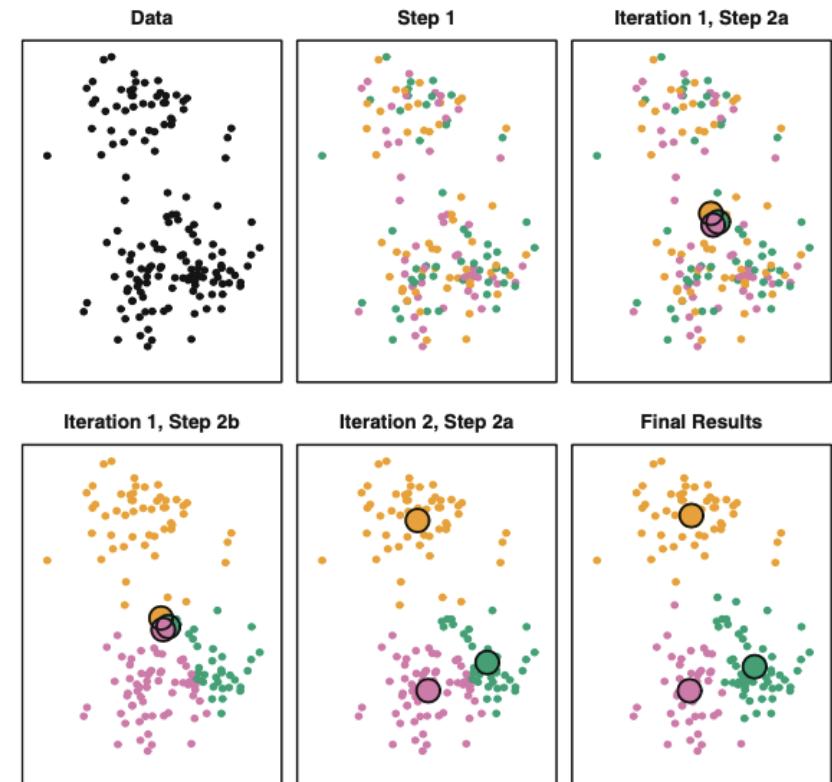
- Metode K -means adalah pendekatan yang sederhana dan elegan untuk mempartisi kumpulan data ke dalam K gerombol yang berbeda dan tidak tumpang tindih.
- Untuk melakukan penggerombolan K -means, pertama-tama kita harus menentukan jumlah gerombol K yang diinginkan; maka algoritma K -means akan menugaskan setiap observasi tepat ke salah satu K gerombol.

Ilustrasi



Algorithm 10.1 K-Means Clustering

1. Randomly assign a number, from 1 to K , to each of the observations. These serve as initial cluster assignments for the observations.
 2. Iterate until the cluster assignments stop changing:
 - (a) For each of the K clusters, compute the cluster *centroid*. The k th cluster centroid is the vector of the p feature means for the observations in the k th cluster.
 - (b) Assign each observation to the cluster whose centroid is closest (where *closest* is defined using Euclidean distance).
-



Penjelasan:

- Kita mulai dengan mendefinisikan beberapa notasi. Misalkan $C_1, C_2 \dots, C_K$ menunjukkan himpunan yang berisi indeks pengamatan di setiap gerombol. Himpunan ini memenuhi dua sifat:
 - $C_1 \cup C_2 \cup \dots \cup C_K = \{1, 2, \dots, n\}$
 - $C_k \cap C_{k'} = \emptyset$ untuk setiap $k \neq k'$
- Gagasan di balik penggerombolan k -means adalah bahwa penggerombolan yang baik adalah penggerombolan yang variasi (keragaman) di dalamnya sekecil mungkin (minimum within-cluster variation).
- Within-cluster variation untuk gerombol C_k diukur oleh $W(C_k)$, sehingga kita ingin menyelesaikan masalah

$$\min_{C_1, C_2, \dots, C_K} \left\{ \sum_{k=1}^K W(C_k) \right\}$$

- Dengan kata lain, rumus ini menyatakan bahwa kita ingin mempartisi observasi ke dalam K gerombol sedemikian rupa sehingga total variasi dalam gerombol, yang dijumlahkan untuk semua K gerombol, sekecil mungkin.

- Umumnya within-cluster variation dihitung dengan squared Euclidean distance (jarak Euclidean kuadrat), dengan rumus:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2$$

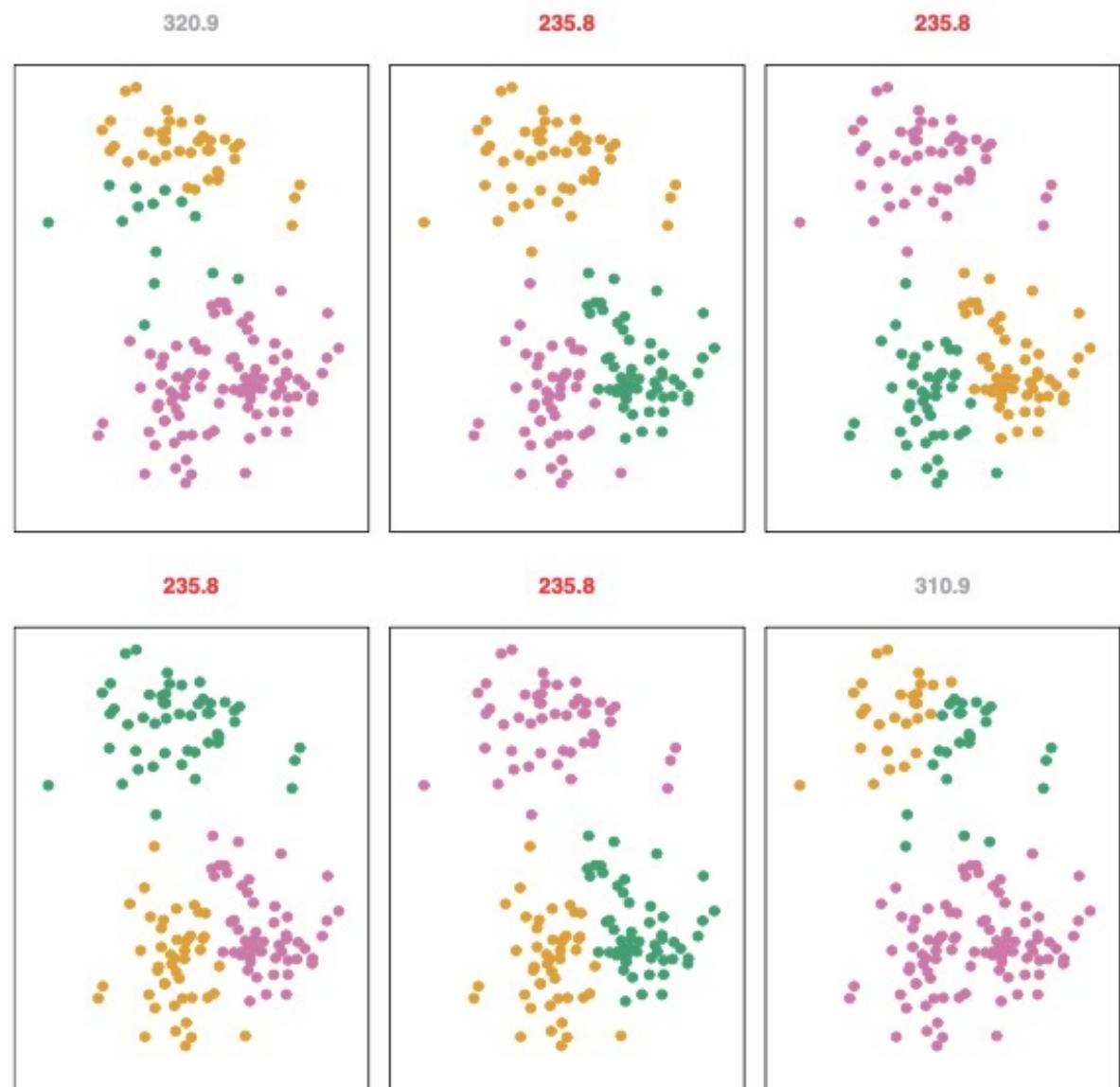
- dengan $|C_k|$ adalah banyaknya pengamatan pada gerombol ke- k .
- Akibatnya, penggerombolan k -means berupaya untuk menyelesaikan masalah pengoptimuman:

$$\min_{c_1, c_2, \dots, c_K} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$$

Catatan:

- Karena algoritma K -means menemukan solusi optimal lokal daripada optimal global, maka hasil yang diperoleh akan bergantung pada inisiasi gerombol awal (acak) dari setiap pengamatan pada Langkah 1 Algoritma 10.1
- Untuk alasan ini, penting untuk menjalankan algoritma beberapa kali dari inisiasi awal dengan acak yang berbeda. Kemudian dipilih solusi terbaik yang memiliki minimum:

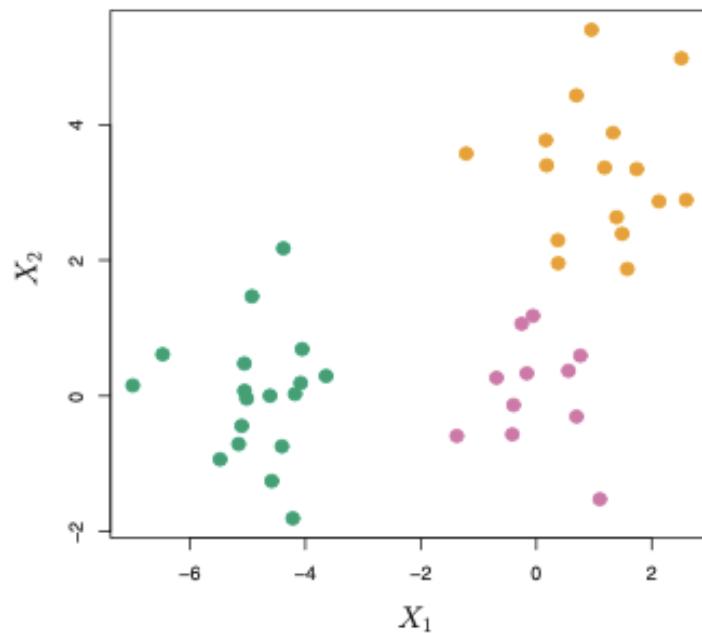
$$\min_{c_1, c_2, \dots, c_K} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$$



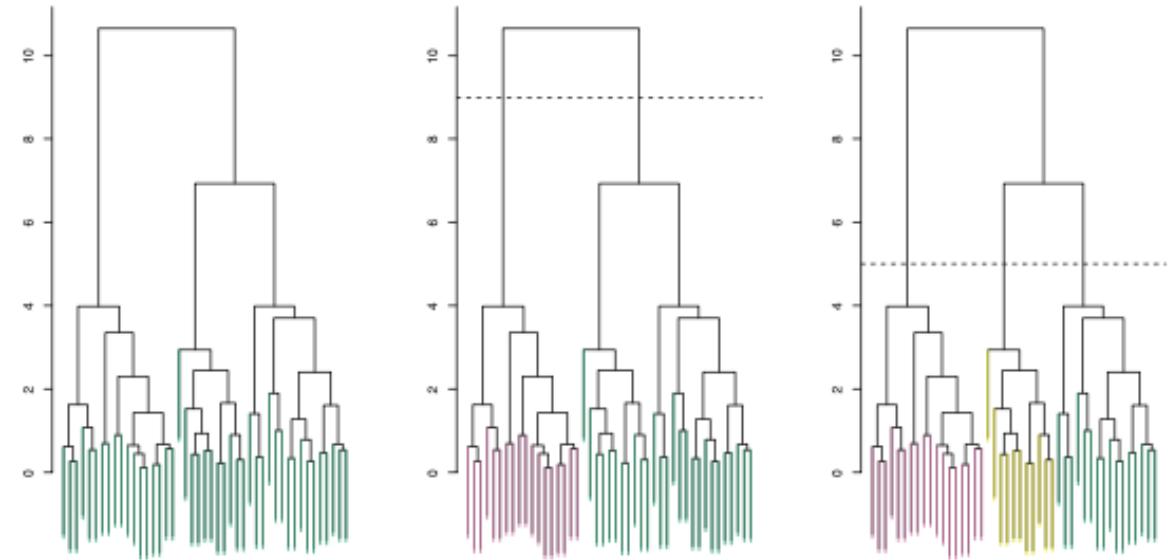
Analisis Gerombol berhierarki

- Salah satu kelemahan dari penggerombolan K -means adalah bahwa kita harus menentukan terlebih dahulu banyaknya gerombol K .
- Penggerombolan berhierarki adalah pendekatan alternatif yang tidak mengharuskan kita berkomitmen pada pilihan K tertentu.
- Penggerombolan berhierarki memiliki nilai tambah keunggulan dibandingkan penggerombolan K -means karena menghasilkan representasi pengamatan berbasis pohon yang menarik, yang disebut dendrogram.

- Pada bagian ini, akan dijelaskan penggerombolan bottom-up atau agglomerative.
- Ini adalah jenis penggerombolan berhierarki yang paling umum, dan mengacu pada fakta bahwa dendrogram dibangun mulai dari daun dan menggabungkan kelompok hingga ke batang.
- Kita akan mulai dengan diskusi tentang bagaimana menginterpretasikan dendrogram dan kemudian mendiskusikan bagaimana sebenarnya pengelompokan hierarki dilakukan—yaitu, bagaimana dendrogram dibangun.



Metode complete linkage and Euclidean distance.



ketinggian potongan ke dendrogram memiliki peran yang sama dengan K dalam pengelompokan K -means: ia mengontrol jumlah kelompok yang diperoleh.

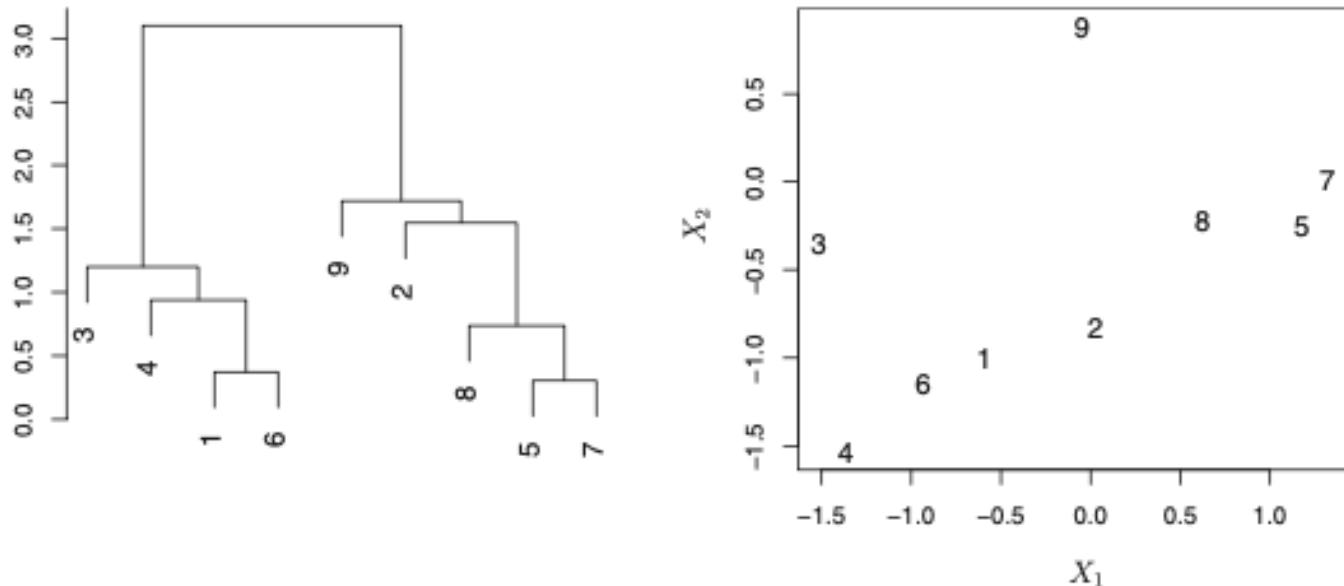


FIGURE 10.10. An illustration of how to properly interpret a dendrogram with nine observations in two-dimensional space. Left: a dendrogram generated using Euclidean distance and complete linkage. Observations 5 and 7 are quite similar to each other, as are observations 1 and 6. However, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7, even though observations 9 and 2 are close together in terms of horizontal distance. This is because observations 2, 8, 5, and 7 all fuse with observation 9 at the same height, approximately 1.8. Right: the raw data used to generate the dendrogram can be used to confirm that indeed, observation 9 is no more similar to observation 2 than it is to observations 8, 5, and 7.

Catatan:

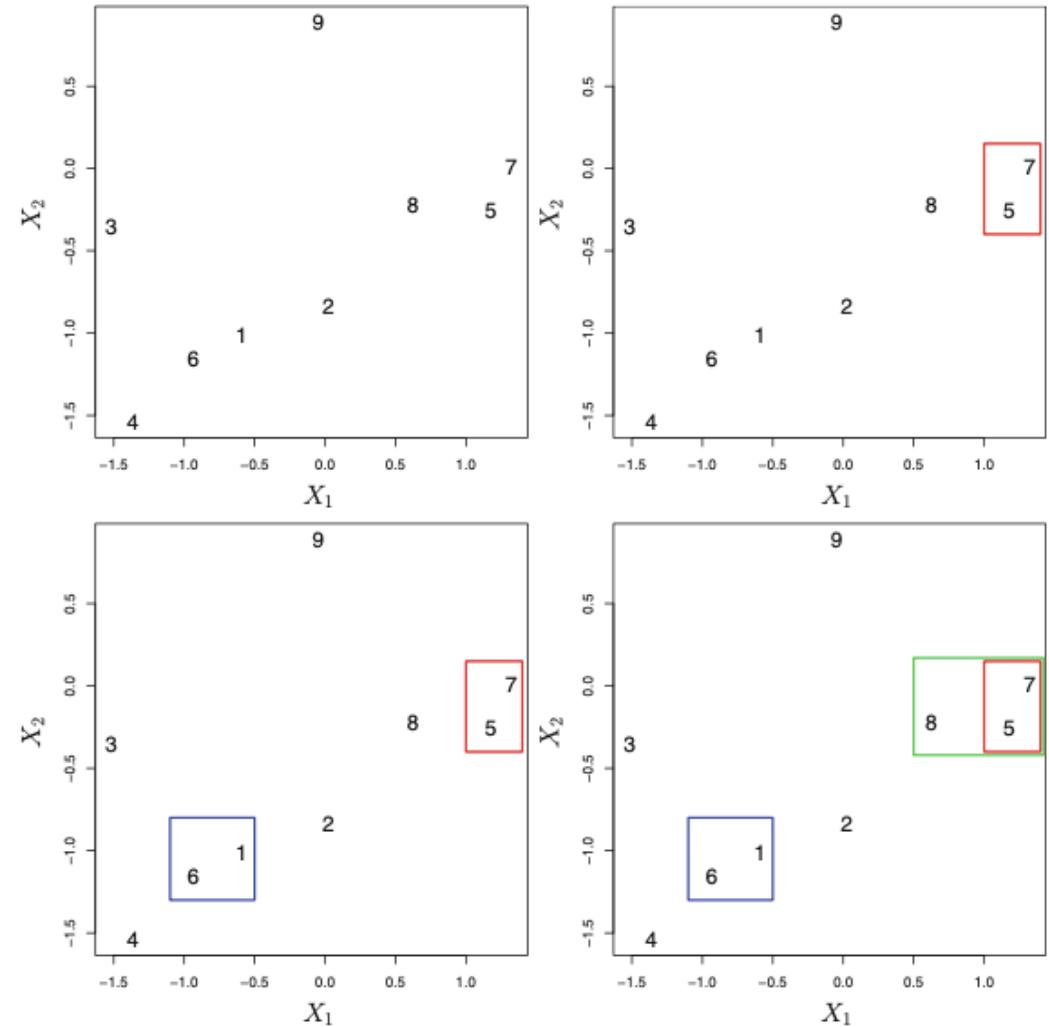
- Dalam praktiknya, orang sering melihat dendrogram dan memilih sejumlah gerombol yang masuk akal, berdasarkan ketinggian penggabungan (fusion) dan jumlah gerombol yang diinginkan.
- Namun, seringkali pilihan tempat pemotongan dendrogram tidak begitu jelas.
- Istilah hierarki mengacu pada fakta bahwa gerombol yang diperoleh dengan memotong dendrogram pada ketinggian tertentu harus bersarang di dalam gerombol yang diperoleh dengan memotong dendrogram pada ketinggian yang lebih tinggi. Namun, pada kumpulan data secara umum, asumsi struktur hierarki ini mungkin tidak realistik.
- Misalnya, anggaplah bahwa pengamatan kita berhubungan dengan sekelompok orang dengan pembagian pria dan wanita 50–50, terbagi rata antara orang Amerika, Jepang, dan Prancis. Kita dapat membayangkan sebuah skenario di mana pembagian terbaik menjadi dua gerombol mungkin membagi orang-orang ini berdasarkan jenis kelamin, dan pembagian terbaik menjadi tiga gerombol mungkin memisahkan mereka berdasarkan kebangsaan. Dalam hal ini, cluster yang sebenarnya tidak bersarang, dalam arti bahwa pembagian terbaik menjadi tiga gerombol tidak dihasilkan dari mengambil pembagian terbaik menjadi dua gerombol dan memisahkan salah satu dari gerombol tersebut.
- Akibatnya, situasi ini tidak dapat terwakili dengan baik oleh penggerombolan berhierarki. Karena situasi seperti ini, penggerombolan berhierarki terkadang dapat menghasilkan hasil yang lebih buruk (kurang akurat) daripada penggerombolan K -means untuk sejumlah kelompok tertentu.

Algoritma Penggerombolan Berhierarki

- Dendrogram penggerombolan berhierarki diperoleh melalui algoritma yang sangat sederhana.
- Kita mulai dengan mendefinisikan semacam ukuran perbedaan (dissimilarity) antara masing-masing pasangan pengamatan. Paling sering, jarak Euclidean digunakan.
- Algoritma berjalan secara iteratif. Dimulai dari bagian bawah dendrogram, masing-masing n pengamatan diperlakukan sebagai gerombol tersendiri. Dua gerombol yang paling mirip satu sama lain kemudian digabungkan sehingga menjadi $n - 1$ gerombol. Selanjutnya dua gerombol yang paling mirip satu sama lain dilebur kembali, sehingga menjadi $n - 2$ gerombol. Algoritma berjalan dengan cara ini sampai semua pengamatan termasuk dalam satu gerombol tunggal, dan dendrogram selesai.
- Konsep perbedaan (dissimilarity) antara sepasang pengamatan perlu diperluas menjadi sepasang kelompok pengamatan. Perluasan ini dicapai dengan mengembangkan gagasan keterkaitan (linkage), yang mendefinisikan perbedaan (dissimilarity) antara dua kelompok pengamatan. Umumnya, jenis linkage: complete, average, single, and centroid

Algorithm 10.2 Hierarchical Clustering

1. Begin with n observations and a measure (such as Euclidean distance) of all the $\binom{n}{2} = n(n - 1)/2$ pairwise dissimilarities. Treat each observation as its own cluster.
 2. For $i = n, n - 1, \dots, 2$:
 - (a) Examine all pairwise inter-cluster dissimilarities among the i clusters and identify the pair of clusters that are least dissimilar (that is, most similar). Fuse these two clusters. The dissimilarity between these two clusters indicates the height in the dendrogram at which the fusion should be placed.
 - (b) Compute the new pairwise inter-cluster dissimilarities among the $i - 1$ remaining clusters.
-



Complete linkage and Euclidean distance

<i>Linkage</i>	<i>Description</i>
Complete	Maximal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the <i>largest</i> of these dissimilarities.
Single	Minimal intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the <i>smallest</i> of these dissimilarities. Single linkage can result in extended, trailing clusters in which single observations are fused one-at-a-time.
Average	Mean intercluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the <i>average</i> of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable <i>inversions</i> .

$$d(i \cup j, k) = \max\{d(i, k), d(j, k)\}$$

$$d(i \cup j, k) = \min\{d(i, k), d(j, k)\}$$

$$d(i \cup j, k) = (1/2)\{d(i, k) + d(j, k)\}$$

$$d(i \cup j, k) = \{\bar{x}_{ij}, \bar{y}_k\}$$

Pemilihan Ukuran dissimilarity

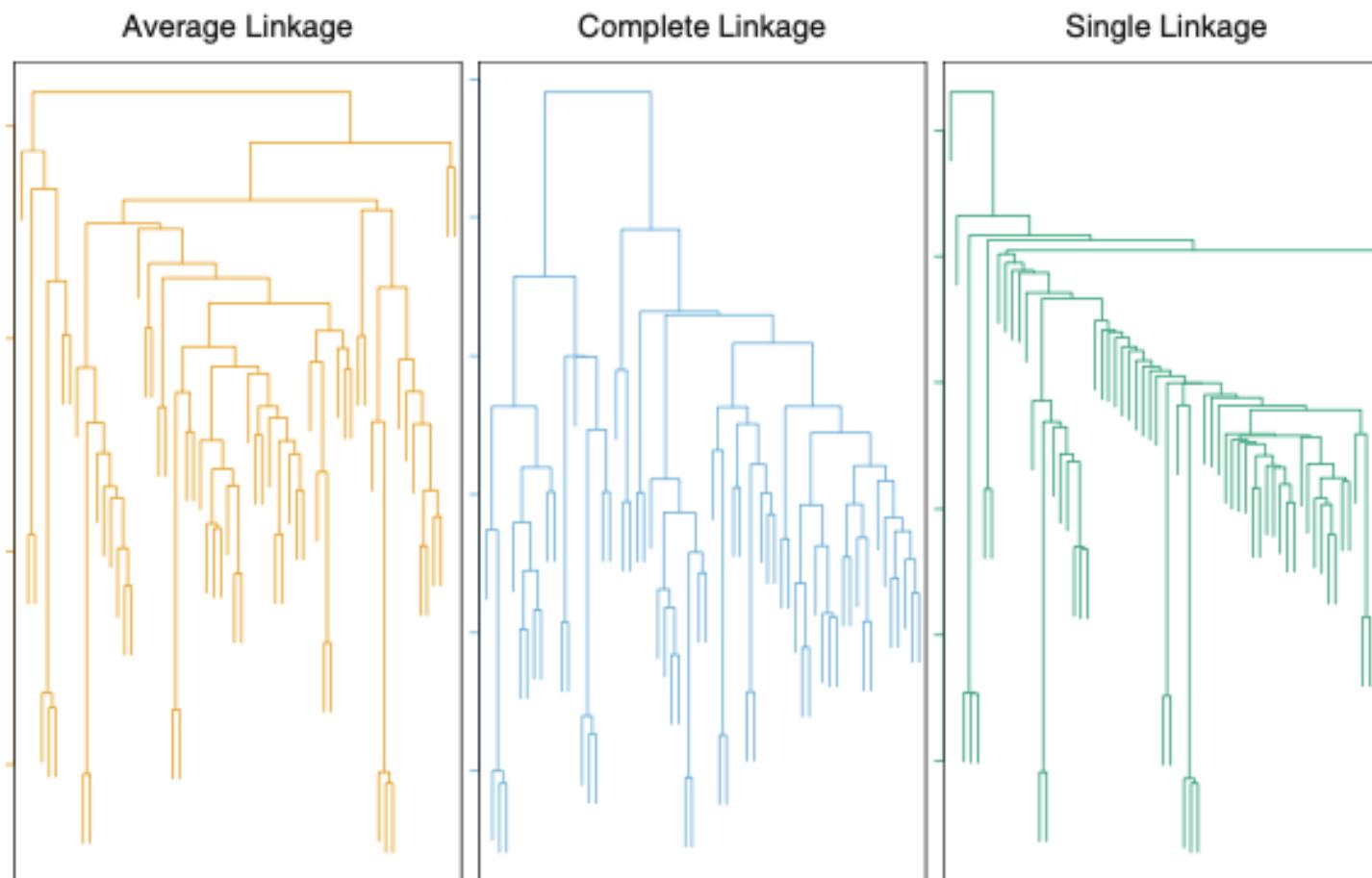


FIGURE 10.12. Average, complete, and single linkage applied to an example data set. Average and complete linkage tend to yield more balanced clusters.

- Pemilihan ukuran dissimilarity sangat penting, karena memiliki efek yang kuat pada dendrogram yang dihasilkan.
- Secara umum, perhatian yang cermat harus diberikan pada jenis data yang digerombolkan dan pertanyaan ilmiah yang ada.
- Pertimbangan ini harus menentukan jenis ukuran ketidaksamaan (dissimilarity) apa yang digunakan untuk penggerombolan berhierarki.

Isu Praktis dalam Analisis Gerombol

- Keputusan Kecil dengan Konsekuensi Besar
 - Haruskah pengamatan atau peubah perlu distandarisasi?
 - Pada kasus penggerombolan berhierarki: ukuran dissimilarity apa yang harus digunakan? Jenis linkage apa yang digunakan? Dimana kita harus memotong dendogram untuk memperoleh sejumlah gerombol?
 - Pada kasus penggerombolan *K*-means, berapa banyak gerombol yang akan kita tentukan pada data?
- Memvalidasi Gerombol yang diperoleh
 - Kita benar-benar ingin tahu apakah gerombol yang ditemukan mewakili subgrup sebenarnya dalam data, atau apakah itu hanya hasil dari penggerombolan noise.

- Pertimbangan Lain dalam Penggerombolan
 - metode penggerombolan umumnya tidak terlalu kuat terhadap gangguan pada data (hasil gerombol bisa sangat berbeda dengan data sama namun banyaknya berbeda)
- Menafsirkan Hasil Penggerombolan
 - Yang terpenting, kita harus berhati-hati tentang bagaimana hasil analisis penggerombolan diinterpretasikan. Hasil ini tidak boleh dianggap sebagai kebenaran mutlak tentang kumpulan data. Sebaliknya, mereka harus merupakan titik awal untuk pengembangan hipotesis ilmiah dan studi lebih lanjut.

Aplikasi di R

```
#k-means
set.seed (2)
x=matrix(rnorm(50*2), ncol=2)
x[1:25,1]=x[1:25,1]+3
x[1:25,2]=x[1:25,2]-4

##k=2
km.out1=kmeans(x,2,nstart=20)
km.out1$cluster
plot(x, col=(km.out1$cluster +1),
      main="K-Means Clustering Results with K=2",
      xlab="", ylab="", pch=20, cex=2)

##k=3
set.seed(4)
km.out2=kmeans(x,3,nstart=20)
km.out2$cluster
plot(x, col=(km.out2$cluster +1),
      main="K-Means Clustering Results with K=3",
      xlab="", ylab="", pch=20, cex=2)

km.out1$tot.withinss
km.out2$tot.withinss
```

```
#Hierarki
hc.complete=hclust(dist(x), method="complete")
hc.average=hclust(dist(x), method="average")
hc.single=hclust(dist(x), method="single")

par(mfrow=c(1,3))
plot(hc.complete,main="Complete Linkage", xlab="", sub="",
      cex =.9)
plot(hc.average , main="Average Linkage", xlab="", sub="",
      cex =.9)
plot(hc.single , main="Single Linkage", xlab="", sub="",
      cex =.9)

###2 cluster
cutree(hc.complete, 2)
cutree(hc.average, 2)
cutree(hc.single, 2)
cutree(hc.single , 4)
```

Reminder Tugas Kelompok

- Gunakan link
<https://www.bps.go.id/site/pilihdata>
- Pilihlah:
 - Minimal 4 peubah yang akan dianalisis (boleh terdapat 1 peubah respon dan minimal 3 predictor)
 - Gunakan satu titik waktu yakni pada suatu tahun tertentu (optional)
 - Gunakan semua peubah dengan level pengamatan dalam kabupaten
- Analisis (pilih):
 - Gunakan pemodelan regresi (dan atau pengembangannya) terhadap peubah-peubah yang dipilih untuk tujuan mencari peubah-peubah prediktor yang signifikan atau untuk tujuan prediksi
 - Gunakan metode unsupervised learning

ISI POSTER: Latar Belakang, Metodologi (Data & Analisis Data), Hasil & Pembahasan, dan Kesimpulan

Deadline waktu pengumpulan: Rabu/ 30 Nov '22

Presentasi hasil poster pada pertemuan 14: Jumat/ 2 Des '22

Tugas Individu

- **Tugas Individu → submit di NewLMS**
 - Deadline waktu pengumpulan: Jumat/ 25 Nov '22 (sebelum waktu kuliah)
 - Penilaian: orisinalitas, kesesuaian jawaban, kecepatan pengiriman
- Diketahui data Mall_Customers.csv yang ingin dikelompokkan berdasarkan peubah Age, Annual.Income, dan Spending.Score, untuk dapat memberikan gambaran strategi marketing yang baik untuk dilakukan.
- Tentukanlah gerombol customer tersebut dengan menggunakan:
 - Analisis gerombol berhierarkhi
 - K-means
- Pilihlah gerombol terbaik antara gerombol-gerombol yang terbentuk!

Terima kasih 😊



Aplikasi Analisis Gerombol

Kuliah 13 – STA1381 Pengantar Sains Data

Septian Rahardiantoro



Penggunaan R

- Diketahui data `Mall_Customers.csv` yang ingin dikelompokkan berdasarkan peubah `Age`, `Annual.Income`, dan `Spending.Score`, untuk dapat memberikan gambaran strategi marketing yang baik untuk dilakukan.
- Tentukanlah gerombol customer tersebut dengan menggunakan:
 - Analisis gerombol berhierarkhi
 - K-means

Persiapan data

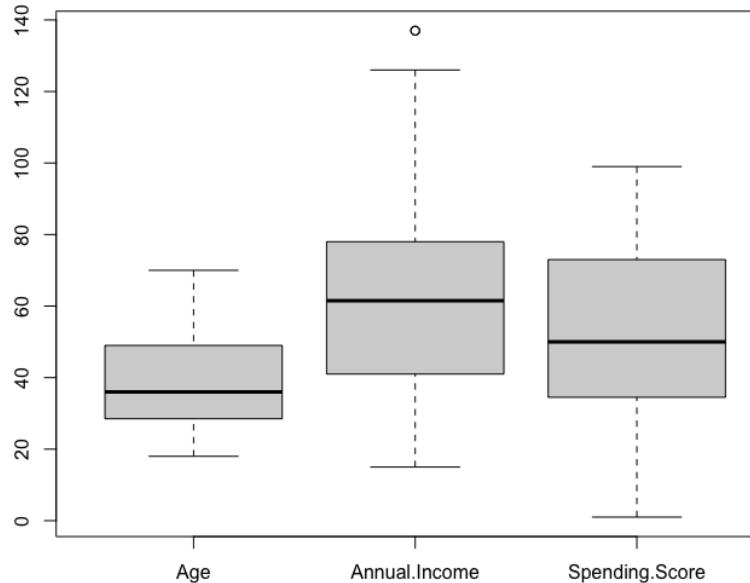
```
#persiapan data
data.mall <- read.csv("Mall_Customers.csv")
str(data.mall)
head(data.mall)

> data.mall <- read.csv("Mall_Customers.csv")
> str(data.mall)
'data.frame': 200 obs. of 5 variables:
 $ CustomerID : int 1 2 3 4 5 6 7 8 9 10 ...
 $ Genre       : chr "Male" "Male" "Female" "Female" ...
 $ Age         : int 19 21 20 23 31 22 35 23 64 30 ...
 $ Annual.Income: int 15 15 16 16 17 17 18 18 19 19 ...
 $ Spending.Score: int 39 81 6 77 40 76 6 94 3 72 ...
> head(data.mall)
  CustomerID Genre Age Annual.Income Spending.Score
1            1 Male  19           15          39
2            2 Male  21           15          81
3            3 Female 20           16           6
4            4 Female 23           16          77
5            5 Female 31           17          40
6            6 Female 22           17          76
```

```
#data yang digunakan
data.mall.OK <- data.mall[,3:5]
str(data.mall.OK)

> data.mall.OK <- data.mall[,3:5]
> str(data.mall.OK)
'data.frame': 200 obs. of 3 variables:
 $ Age       : int  19 21 20 23 31 22 35 23 64 30 ...
 $ Annual.Income: int  15 15 16 16 17 17 18 18 19 19 ...
 $ Spending.Score: int  39 81 6 77 40 76 6 94 3 72 ...
```

```
boxplot(data.mall.OK)
```



```
#standarisasi peubah
data.mall.stdz <- scale(data.mall.OK)
apply(data.mall.stdz,2,mean)          #rataan 0
apply(data.mall.stdz,2,sd)           #sd 1

> data.mall.stdz <- scale(data.mall.OK)
> apply(data.mall.stdz,2,mean)
    Age  Annual.Income Spending.Score
-1.016906e-16 -8.144310e-17 -1.096708e-16
> apply(data.mall.stdz,2,sd)
    Age  Annual.Income Spending.Score
    1      1            1
```

Analisis gerombol berhierarkhi

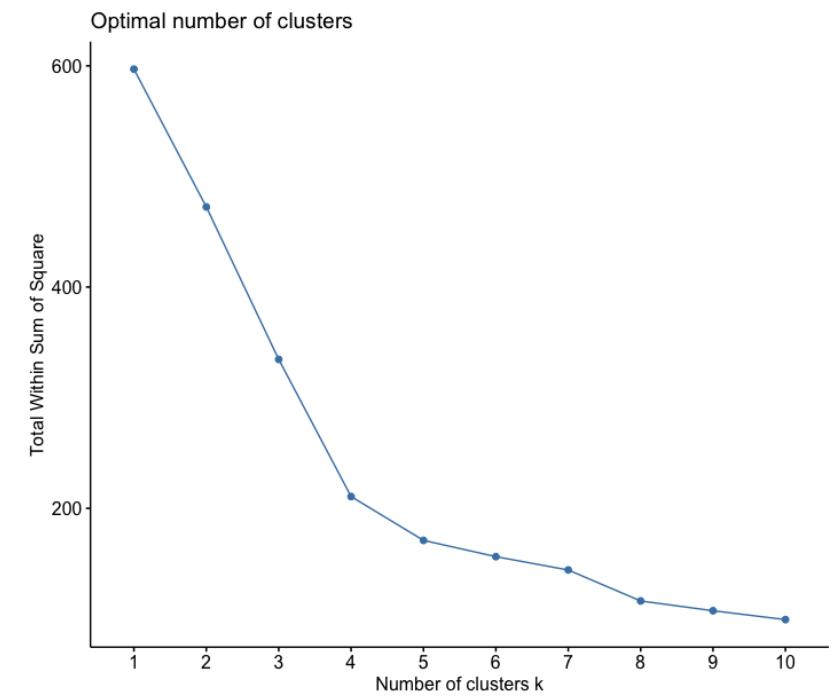
Memilih metode linkage dan banyaknya cluster

Pada kasus ini menggunakan package “factoextra”

Menggunakan kriteria wss yang sudah relative tidak berubah

```
#Analisis gerombol berhierarkhi
##Memilih metode linkage dan banyaknya cluster
install.packages("factoextra")
library(factoextra)

##complete
a1 <- fviz_nbclust(data.mall.stdz,FUNcluster =
hcut,method = "wss",hc_method = "complete",hc_metric =
"euclidean")
a1
a1$data
```



```

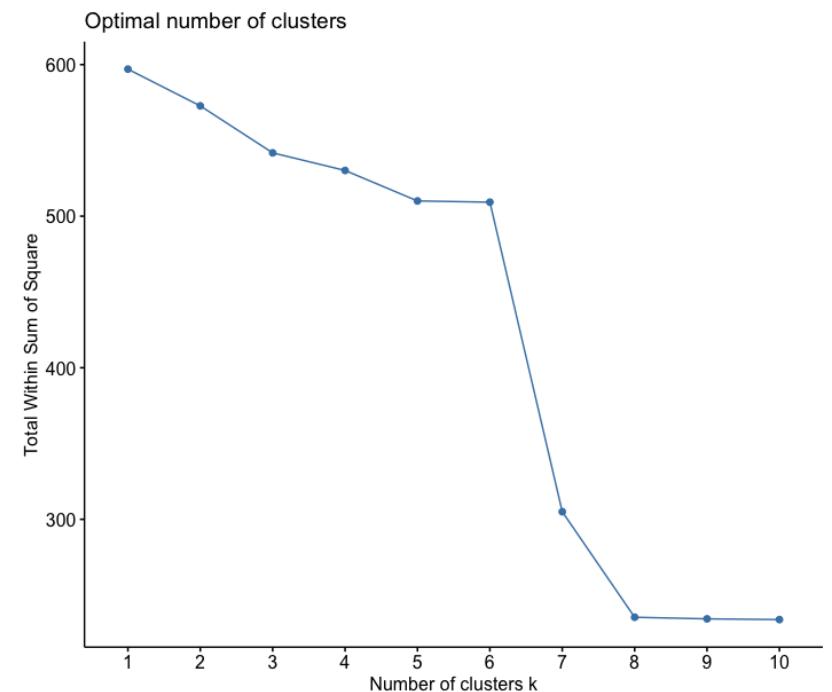
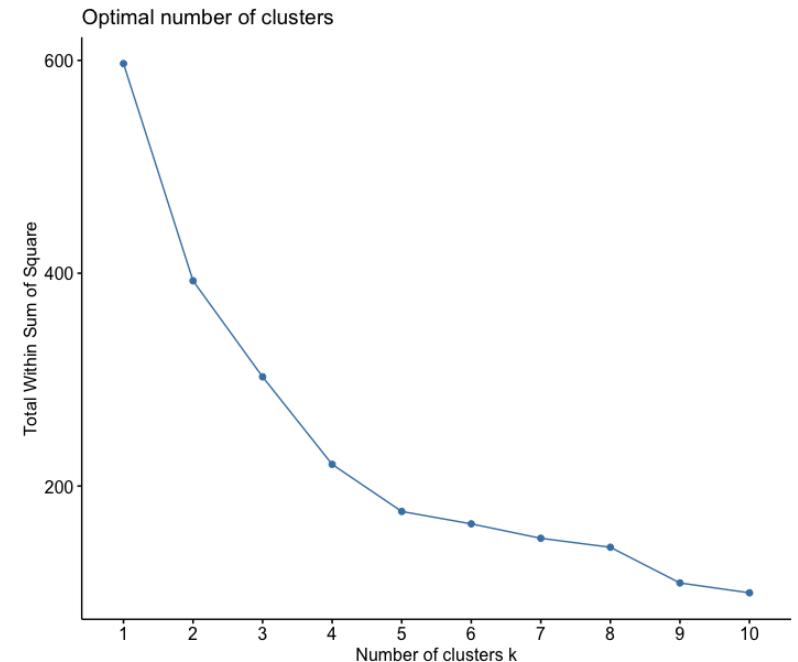
a2 <- fviz_nbclust(data.mall.stdz, FUNcluster =
hcut, method = "wss", hc_method = "average", hc_metric
= "euclidean")
a2
a2$data

```

```

##centroid
a3 <- fviz_nbclust(data.mall.stdz, FUNcluster =
hcut, method = "wss", hc_method = "centroid", hc_metric
= "euclidean")
a3
a3$data

```

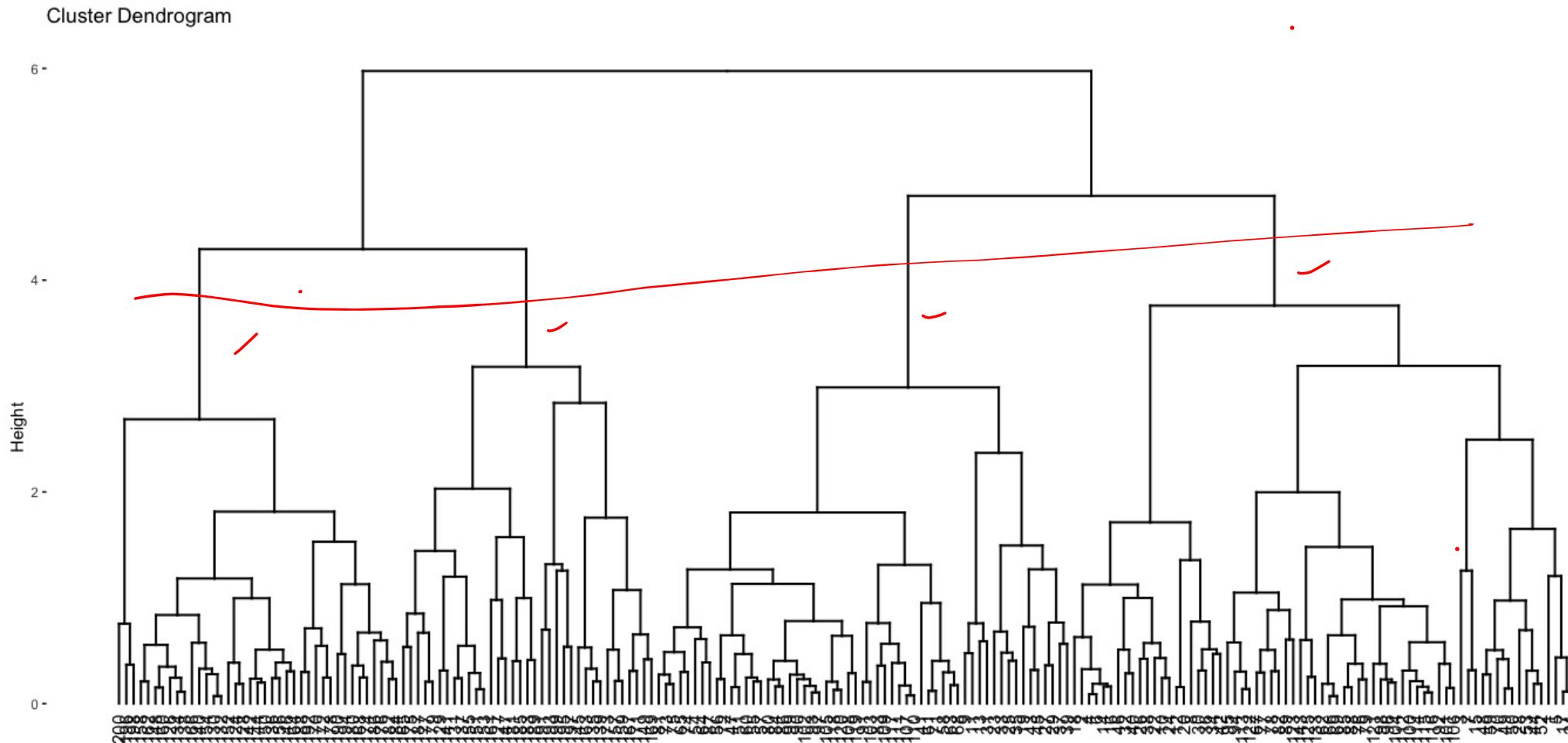


```
cbind(clusters=a1$data[,1], complete=a1$data[,2], average=a2$data[,2],  
centroid=a3$data[,2])
```

	clusters	complete	average	centroid
[1,]	1	597.00000	597.00000	597.0000
[2,]	2	472.44364	392.89818	572.7536
[3,]	3	334.64043	302.62546	541.7832
[4,]	4	210.61574	220.40815	530.1611
[5,]	5	171.00969	176.13829	510.0553
[6,]	6	156.28341	164.47095	509.2355
[7,]	7	144.22360	150.90845	305.0162
[8,]	8	116.22958	142.48203	235.3848
[9,]	9	107.39757	108.93658	234.3236
[10,]	10	99.36778	99.67111	233.8285

Dendogram

```
##misalkan dipilih metode linkage = complete  
fviz_dend(hclust(dist(data.mall.stdz),method = 'euclidean'),method = "complete"))
```



Interpretasi Gerombol yang terbentuk

```
#interpretasi  
hc.mall <- eclust(data.mall.OK, stand = TRUE, FUNcluster = "hclust", k=4, hc_method  
= "average", hc_metric = "euclidean", graph = F)  
hc.mall$cluster
```

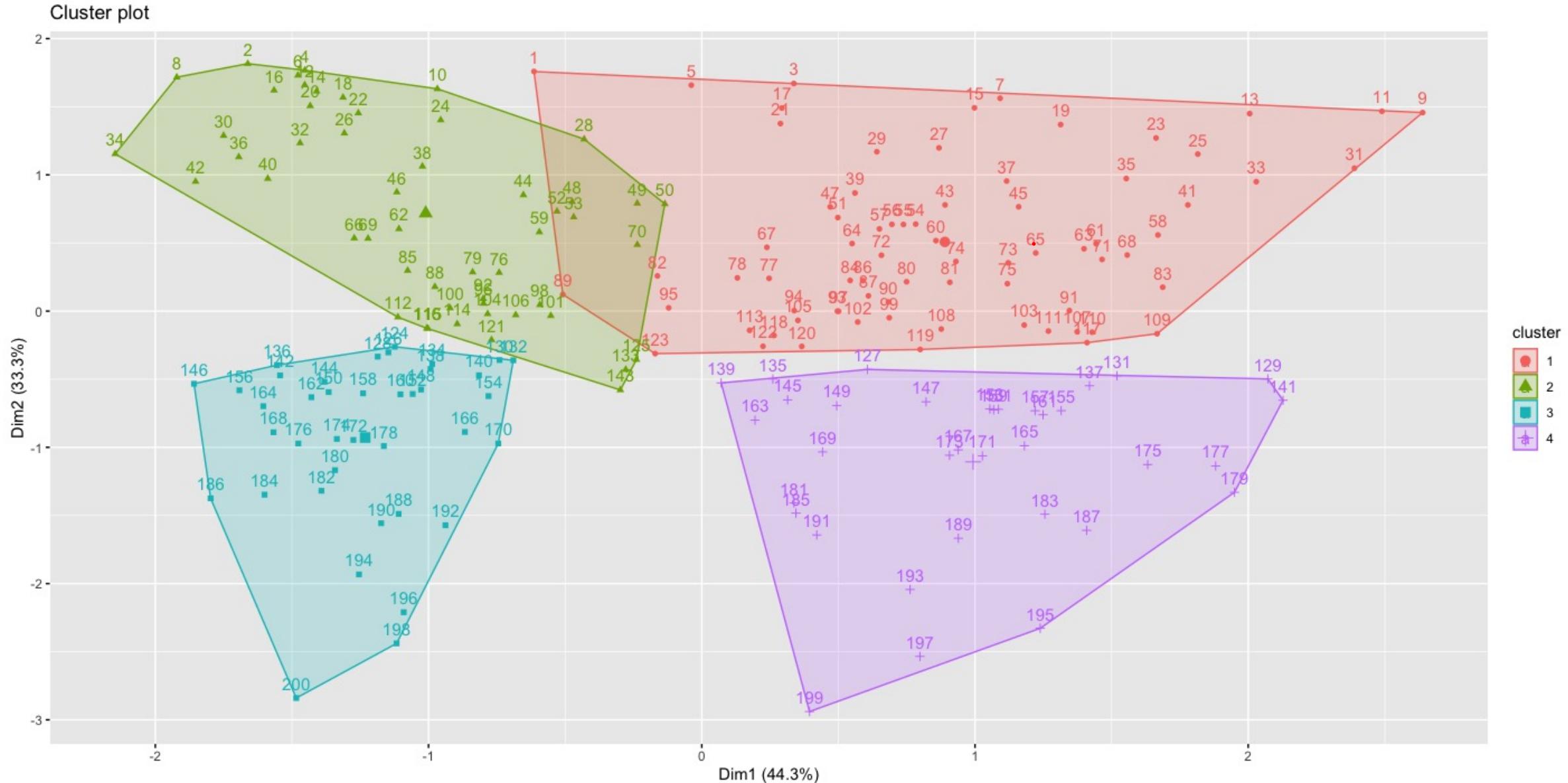
```
> hc.mall$cluster
```

```
[1] 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2  
[35] 1 2 1 2 1 2 1 2 1 2 1 2 2 2 1 2 2 1 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1  
[69] 2 2 1 1 1 1 2 1 1 2 1 1 1 1 2 1 1 2 1 1 1 2 1 1 2 1 1 2 1 2 1 2 2 1  
[103] 1 2 1 2 1 1 1 1 2 1 2 2 2 1 1 1 2 1 1 3 2 3 4 3 4 3 4 3 2 3 4 3  
[137] 4 3 4 3 4 3 2 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3  
[171] 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3 4 3
```

```
aggregate(data.mall.OK, by=list(cluster=hc.mall$cluster), FUN = mean)
```

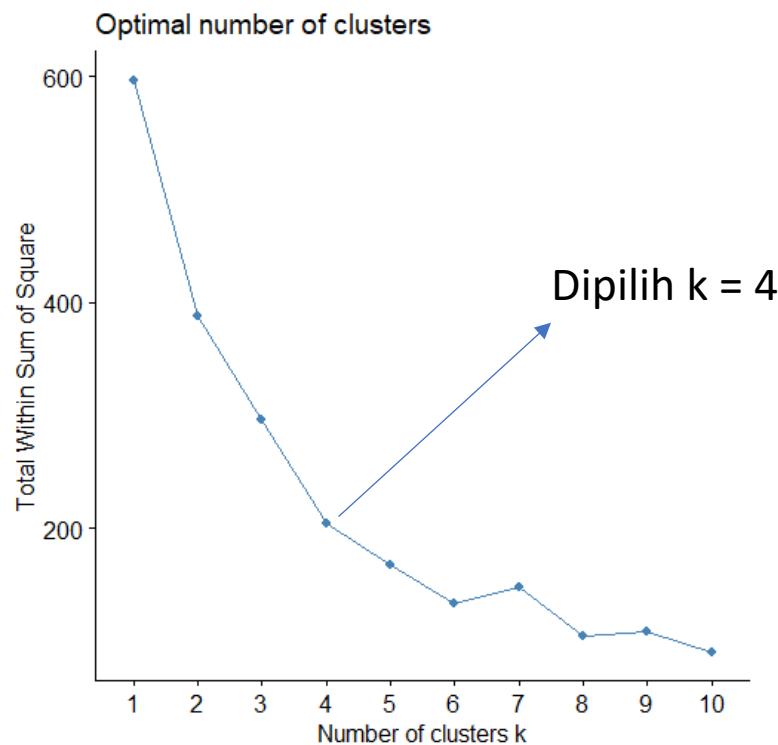
	cluster	Age	Annual.Income	Spending.Score
1	1	50.72973	46.16216	40.59459
2	2	24.65385	42.94231	62.07692
3	3	32.69231	86.53846	82.12821
4	4	41.68571	88.22857	17.28571

```
#scatterplot  
fviz_cluster(hc.mall)
```



Analisis gerombol tak berhierarkhi: k-means

```
#k-means  
##penentuan k dengan wss  
b1 <- fviz_nbclust(data.mall.stdz, FUNcluster = kmeans, method = "wss")  
b1  
b1$data
```



```
kmeans.mall <- eclust(data.mall.OK, stand = TRUE, FUNcluster = "kmeans", k=4, graph = F)
kmeans.mall$cluster
```

```
> kmeans.mall$cluster
```

```
[1] 3 3 3 3 3 3 2 3 2 3 2 3 2 3 3 3 2 3 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2  
[38] 3 2 3 2 3 2 3 2 3 3 3 2 3 3 2 2 2 2 2 3 2 2 3 2 2 2 3 2 2 3 2 2 3 3 3 2 2 2  
[75] 2 3 2 2 3 2 2 3 2 2 3 3 2 2 3 2 2 3 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2 2 3 3 3 2 2 2  
[112] 3 1 3 3 3 2 2 2 2 3 1 4 4 1 4 1 4 2 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4  
[149] 1 4 1 4 1 4 1 4 1 4 2 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1  
[186] 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4
```

#interpretasi

```
kmeans.mall$centers
```

```
aggregate(data.mall.OK, by=list(cluster=kmeans.mall$cluster), FUN = mean)
```

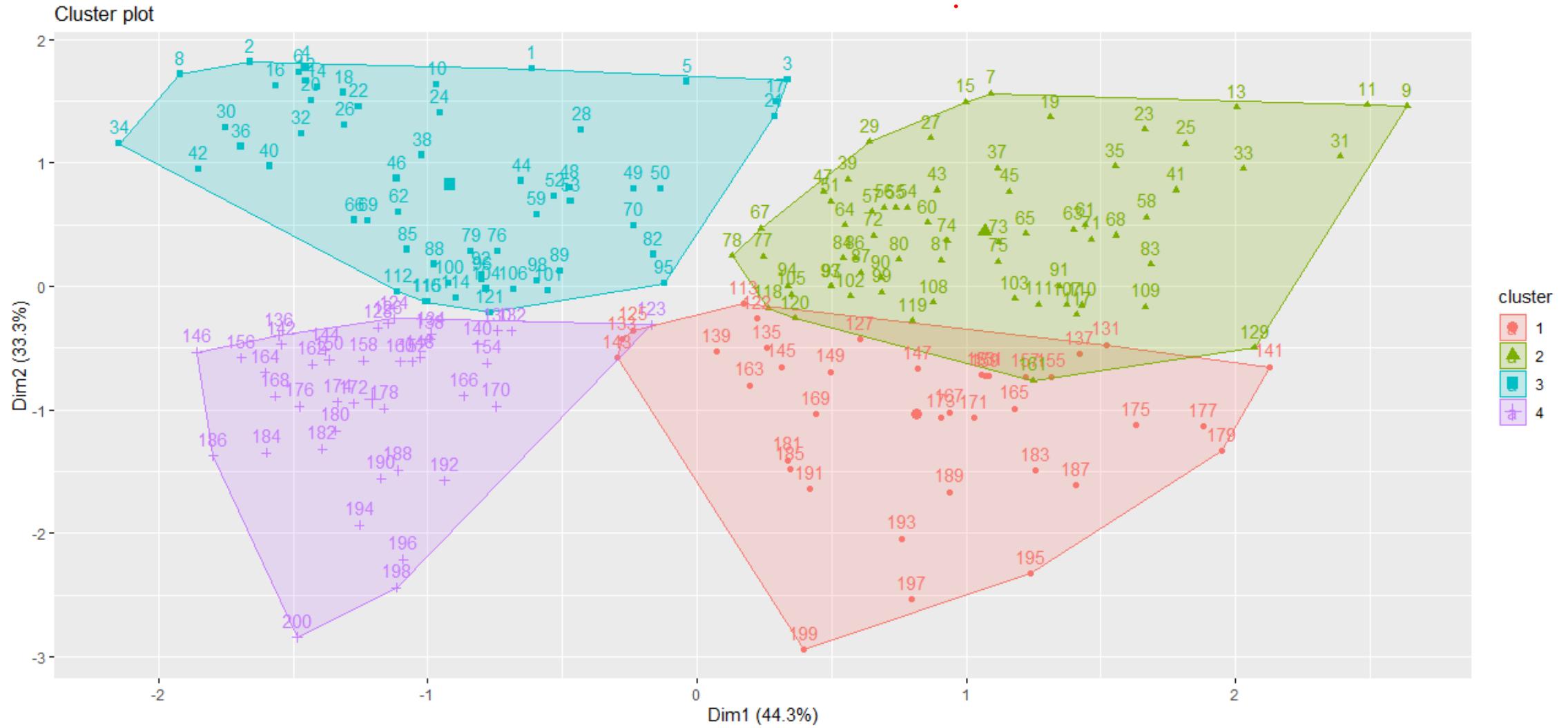
```
> kmeans.mall$centers
```

	Age	Annual.Income	Spending.Score
1	0.03711223	0.9876366	-1.1857814
2	1.08344244	-0.4893373	-0.3961802
3	-0.96008279	-0.7827991	0.3910484
4	-0.42773261	0.9724070	1.2130414

```
> aggregate(data.mall.OK, by=list(cluster=kmeans.mall$cluster), FUN = mean)
```

cluster	Age	Annual.Income	Spending.Score
1	39.36842	86.50000	19.57895
2	53.98462	47.70769	39.96923
3	25.43860	40.00000	60.29825
4	32.87500	86.10000	81.52500

```
#scatterplot  
fviz_cluster(kmeans.mall)
```



```
##perbandingan nilai wss complete vs k-means  
cbind(clusters=a1$data[,1],complete=a1$data[,2],kmeans=b1$data[,2])
```

	clusters	complete	kmeans
[1,]	1	597.00000	597.00000
[2,]	2	472.44364	387.43926
[3,]	3	334.64043	295.51779
[4,]	4	210.61574	204.19902
[5,]	5	171.00969	167.85741
[6,]	6	156.28341	133.19899
[7,]	7	144.22360	147.86523
[8,]	8	116.22958	104.68068
[9,]	9	107.39757	107.61878
[10,]	10	99.36778	89.69673

Dipilih k-means dengan k=4

Terima kasih😊

Sumber data: <https://gerrydito.github.io/Analisis-Clustering-menggunakan-R/>