UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

Supervised Learning

#2 Meeting

OLS, WLS dan GLS

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001

Regresi Linier - Review OLS

Sejauh ini, kita telah mempelajari metode Ordinary Least Squares (OLS) sebagai fondasi dari model regresi. Prinsip utama OLS adalah menemukan garis yang meminimalkan jumlah dari kuadrat residual ($\sum$ $error^2$). Di balik kesederhanaan ini, OLS memiliki sebuah asumsi implisit yang sangat kuat: setiap observasi data sama pentingnya dan sama andalnya.

Bayangkan OLS sebagai sebuah sistem demokrasi yang sempurna, di mana setiap titik data memiliki hak suara yang sama persis (satu suara) dalam menentukan di mana garis regresi "terbaik" akan diletakkan. Untuk banyak kasus, pendekatan ini bekerja dengan sangat baik.

Regresi Linier - Heteroscedasticity

Heteroscedasticity adalah kondisi di mana tingkat penyebaran (varians) dari error tidak konstan di seluruh rentang data. Artinya, tingkat ketidakpastian prediksi model kita berubah-ubah.

Analogi Sederhana: Memprediksi Harga Rumah, Bayangkan membuat model untuk memprediksi harga rumah berdasarkan luas bangunannya.

1. Untuk Rumah Kecil (misal, 50 m²): Harga rumah di kisaran ini cenderung seragam. Prediksi mungkin meleset sekitar ± Rp 50 juta. Data di area ini akan bergerombol rapat di sekitar garis regresi. Observasi ini bisa kita anggap sangat andal.
2. Untuk Rumah Mewah (misal, 500 m²): Harga di kisaran ini bisa sangat bervariasi tergantung pada kualitas marmer, desain interior, merek lift, dll. Prediksi bisa meleset sangat jauh, mungkin ± Rp 2 Miliar. Data di area ini akan tersebar sangat lebar di sekitar garis regresi. Observasi ini kurang andal.

Regresi Linier - Heteroscedasticity

Dalam situasi ini, OLS akan menghadapi masalah. Karena OLS meminimalkan kuadrat error, observasi rumah mewah yang memiliki potensi error sangat besar akan memiliki pengaruh yang tidak proporsional (suara yang jauh lebih "bising"). Satu atau dua data rumah mewah yang outlier bisa "menarik" garis regresi secara signifikan, sehingga merusak akurasi prediksi untuk mayoritas data (rumah-rumah kecil dan menengah).

Solusi: **Weighted Least Squares (WLS)**

Regresi Linier - WLS

Weighted Least Squares (WLS) adalah solusi elegan untuk masalah heteroscedasticity. Prinsip utamanya adalah menolak gagasan bahwa semua data sama pentingnya. Sebaliknya, WLS memperkenalkan sistem "demokrasi yang diboboti".

Cara kerjanya adalah dengan memberikan bobot (weight) pada setiap observasi sebelum menjumlahkan kuadrat errornya. Aturan pembobotannya sangat intuitif:

- Observasi yang dianggap lebih andal (memiliki varians error yang kecil, seperti data rumah kecil) akan diberi bobot yang TINGGI.
- Observasi yang kurang andal (memiliki varians error yang BESAR, seperti data rumah mewah) akan diberi bobot yang RENDAH.

WLS memastikan bahwa observasi tersebut tidak memiliki pengaruh berlebihan dalam menentukan posisi akhir garis regresi. Garis tersebut akan lebih "tertarik" ke arah titik-titik data yang lebih andal dan konsisten.

Regresi Linier - WLS

- Tujuan OLS: Menemukan koefisien (β) yang meminimalkan:

$$RSS_{OLS} = \sum(\text{error}_i)^2$$

- Tujuan WLS: Menemukan koefisien (β) yang meminimalkan:

$$RSS_{WLS} = \sum w_i \cdot (\text{error}_i)^2$$

Di mana wi adalah bobot yang ditetapkan untuk observasi ke-i. Dengan pendekatan ini, WLS menghasilkan estimasi koefisien yang secara statistik lebih efisien dan dapat diandalkan ketika asumsi homoscedasticity dilanggar.

Regresi Linier - WLS

Minimalisasi RSS (Residual Sum of Square) pada Regresi Linier Berganda

Bobot berbanding terbalik dengan varians error dari observasi

$w_i = 1/\sigma_i^2$

Buat sebuah matriks diagonal (W) berukuran (n x n)

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ . & . & \dots & . \\ 0 & 0 & \dots & w_n \end{bmatrix}$$

Jika

$\sum e_i^2 = e^T e$

Maka, untuk terboboti menjadi :

$\sum w_i e_i^2 = e^T W e$

RSS_wls = $e^T W e$

Karena

$e = y - X\beta$

Maka

RSS_wls = $(y - X\beta)^T W (y - X\beta)$

Jika semua observasi dianggap sama penting (kasus OLS), maka semua w_i akan bernilai 1, dan matriks W akan menjadi matriks identitas (I).

Regresi Linier - WLS

Prosesnya sangat mirip dengan OLS, tetapi sekarang matriks W ada di tengah.

$$RSS\_wls = (y^T - (X\beta)^T)\, W\, (y - X\beta)$$
$$RSS\_wls = (y^T - \beta^T X^T)\, W\, (y - X\beta)$$
$$RSS\_wls = (y^T W - \beta^T X^T W)\, (y - X\beta)$$
$$RSS\_wls = y^T W y - y^T W X\beta - \beta^T X^T W y + \beta^T X^T W X\beta$$

Sama seperti OLS, dua suku di tengah adalah skalar yang nilainya sama, jadi bisa digabungkan:

$$RSS\_wls = y^T W y - 2\beta^T X^T W y + \beta^T X^T W X\beta$$

Turunan terhadap β dan Atur Sama Dengan Nol:

$$\partial(RSS\_wls)/\partial\beta = 0 - 2X^T W y + 2X^T W X\beta$$

$$2X^T W X\beta - 2X^T W y = 0$$
$$2X^T W X\beta = 2X^T W y$$
$$X^T W X\beta = X^T W y$$

Selesaikan Persamaan untuk Mendapatkan β_wls

$$(X^T W X)^{-1} (X^T W X)\, \beta = (X^T W X)^{-1} X^T W y$$

$$\hat{\beta}\_wls = (X^T W X)^{-1} X^T W y$$

Regresi Linier - Autokorelasi

WLS baru menyelesaikan separuh dari potensi masalah yang ada pada error. WLS memperbaiki masalah varians, tetapi ia masih bekerja di bawah asumsi bahwa setiap error adalah kejadian yang independen dan tidak berhubungan satu sama lain. Bagaimana jika asumsi ini juga dilanggar?

Regresi Linier - Autokorelasi

Autokorelasi adalah kondisi di mana error dari satu observasi berkorelasi atau berhubungan dengan error dari observasi lainnya. Artinya, nilai error pada satu titik data memberi kita petunjuk tentang kemungkinan nilai error pada titik data lain.

Analogi Sederhana: Memprediksi Penjualan Es Krim Harian, membuat model regresi untuk memprediksi penjualan es krim berdasarkan suhu harian.

- Pada hari Senin, suhu tinggi, dan model memprediksi penjualan 100 cup. Kenyataannya, terjual 120 cup karena ada festival kejutan di dekat toko. Model memiliki error positif yang besar (+20).
- Pada hari Selasa, festival tersebut masih berlangsung. Meskipun suhu sedikit berbeda, faktor tak terduga (festival) yang menyebabkan error besar kemarin kemungkinan besar masih ada. Jadi, sangat mungkin model akan kembali melakukan under-prediction (kesalahan positif lagi).

Regresi Linier - Autokorelasi

Dalam kasus ini, error pada hari Senin tidak independen dari error pada hari Selasa. Error seolah-olah memiliki "ingatan". Inilah yang disebut autokorelasi. Masalah ini sangat umum terjadi pada data yang memiliki urutan, seperti data deret waktu (time-series) atau data geografis.

Jika kita mengabaikan autokorelasi dan tetap menggunakan OLS atau WLS, koefisien (β) yang kita hasilkan mungkin tidak bias, tetapi standar errornya akan salah. Akibatnya, seluruh uji hipotesis (p-value, interval kepercayaan) menjadi tidak dapat diandalkan.

Solusi: **Generalized Least Squares (GLS)**

Regresi Linier - GLS

Generalized Least Squares (GLS) adalah metode yang dirancang untuk menjadi solusi komprehensif yang menangani struktur error yang kompleks. GLS dapat mengatasi keduanya secara bersamaan:

1. Heteroscedasticity (varians error tidak konstan).
2. Autokorelasi (error saling berhubungan).

Jika WLS menggunakan bobot individual untuk setiap observasi, GLS menggunakan pendekatan yang lebih holistik. Ia memodelkan keseluruhan struktur varians-kovarians dari error menggunakan sebuah matriks khusus yang disebut Omega (Ω).

- Diagonal dari matriks Ω ini menangani varians setiap error (seperti WLS).
- Elemen di luar diagonal (off-diagonal) dari matriks Ω menangani kovarians (korelasi) antara error yang berbeda.

Regresi Linier - GLS

Definisikan matriks varians-kovarians dari error (Ω, dibaca Omega). Ini adalah matriks berukuran (n x n)

Elemen Diagonal (Ω_{ii}): Mewakili varians dari error untuk observasi ke-i. Bagian ini menangani heteroscedasticity.
Elemen Luar Diagonal (Ω_{ij}): Mewakili kovarians antara error dari observasi ke-i dan observasi ke-j.

Bagian inilah yang menangani autokorelasi. Jika tidak ada autokorelasi, semua elemen ini akan bernilai nol.

RSS_gls = $e^T\Omega^{-1}e$

Karena

$e = y - X\beta$

Maka

RSS_gls = $(y - X\beta)^T \Omega^{-1} (y - X\beta)$

Sama seperti WLS.Hanya perlu mengganti setiap W dengan Ω^{-1}.

RSS_gls = $(y^T - (X\beta)^T) \Omega^{-1} (y - X\beta)$
RSS_gls = $(y^T - \beta^TX^T) \Omega^{-1} (y - X\beta)$
RSS_gls =
$y^T\Omega^{-1}y - y^T\Omega^{-1}X\beta - \beta^TX^T\Omega^{-1}y + \beta^TX^T\Omega^{-1}X\beta$
RSS_gls =
$y^T\Omega^{-1}y - 2\beta^TX^T\Omega^{-1}y + \beta^TX^T\Omega^{-1}X\beta$

Regresi Linier - GLS

Turunan terhadap β dan Atur Sama Dengan Nol:

$$\partial(RSS\_gls)/\partial\beta = 0 - 2X^T\Omega^{-1}y + 2X^T\Omega^{-1}X\beta$$

$$2X^T\Omega^{-1}X\beta - 2X^T\Omega^{-1}y = 0$$

$$2X^T\Omega^{-1}X\beta = 2X^T\Omega^{-1}y$$

$$X^T\Omega^{-1}X\beta = X^T\Omega^{-1}y$$

Selesaikan Persamaan untuk Mendapatkan β_gls

$$(X^T\Omega^{-1}X)^{-1} (X^T\Omega^{-1}X) \beta = (X^T\Omega^{-1}X)^{-1} X^T\Omega^{-1}y$$

Solusi akhir untuk koefisien GLS:

$$\hat{\beta}\_gls = (X^T\Omega^{-1}X)^{-1} X^T\Omega^{-1}y$$

Regresi Linier - GLS

1.GLS adalah bentuk yang paling umum.

2. WLS adalah kasus spesial dari GLS.
Ini terjadi ketika tidak ada autokorelasi, sehingga semua elemen di luar diagonal matriks Ω adalah nol. Dalam kasus ini, Ω menjadi matriks diagonal, dan inversnya (Ω^{-1}) adalah matriks diagonal yang sama persis dengan matriks W pada WLS. Oleh karena itu, formula GLS berubah menjadi formula WLS.

3. OLS adalah kasus spesial dari GLS (dan WLS). Ini terjadi ketika error "sempurna": tidak ada autokorelasi DAN tidak ada heteroscedasticity (homoscedastic). Dalam kasus ini, matriks Ω menjadi matriks identitas I (dikalikan sebuah skalar varians σ^2 yang akan saling meniadakan). Ketika Anda mengganti Ω^{-1} dengan I dalam formula GLS, Maka akan mendapatkan kembali formula OLS.

Regresi Linier - Implementasi OLS, WLS, GLS

y	x0 (Intercept)	x1	x2	x3
8.1	1	1	2	5
9.5	1	2	3	4
10.2	1	3	2	6
13.5	1	4	4	5
12.9	1	5	3	7
17.8	1	6	5	6
18.1	1	7	4	8
22	1	8	6	7
21.5	1	9	5	9
25.4	1	10	7	8

Regresi Linier - Implementasi OLS

$$\mathbf{y} = \begin{bmatrix} 8.1 \\ 9.5 \\ 10.2 \\ 13.5 \\ 12.9 \\ 17.8 \\ 18.1 \\ 22.0 \\ 21.5 \\ 25.4 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & 1 & 2 & 5 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 6 \\ 1 & 4 & 4 & 5 \\ 1 & 5 & 3 & 7 \\ 1 & 6 & 5 & 6 \\ 1 & 7 & 4 & 8 \\ 1 & 8 & 6 & 7 \\ 1 & 9 & 5 & 9 \\ 1 & 10 & 7 & 8 \end{bmatrix}$$

$$\hat{\beta}_{OLS} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$$

$$(X'X) = \begin{bmatrix} 10 & 55 & 41 & 65 \\ 55 & 385 & 266 & 395 \\ 41 & 266 & 193 & 280 \\ 65 & 395 & 280 & 445 \end{bmatrix}$$

$$(X'X)^{-1} = \begin{bmatrix} 45.07 & 7.76 & -8.33 & -8.23 \\ 7.76 & 1.40 & -1.50 & -1.43 \\ -8.33 & -1.50 & 1.67 & 1.50 \\ -8.23 & -1.43 & 1.50 & 1.53 \end{bmatrix}$$

$$(X'y) = \begin{bmatrix} 159.0 \\ 1033.2 \\ 736.5 \\ 1099.8 \end{bmatrix}$$

$$\hat{\beta} = \begin{bmatrix} -2.40 \\ 0.25 \\ 2.40 \\ 1.09 \end{bmatrix}$$

Persamaan Regresi dengan OLS

$$\hat{y} = -2.40 + 0.25\,x1 + 2.40\,x2 + 1.09\,x3$$

Regresi Linier - Implementasi OLS - Syntax R

```
# --- Menyiapkan Data ---
# Vektor variabel dependen (y)
y_dependent <- c(8.1, 9.5, 10.2, 13.5, 12.9, 17.8, 18.1, 22.0, 21.5, 25.4)

# Vektor-vektor variabel independen (x)
# Diambil dari kolom 2, 3, dan 4 pada matriks X sebelumnya
x1 <- 1:10
x2 <- c(2, 3, 2, 4, 3, 5, 4, 6, 5, 7)
x3 <- c(5, 4, 6, 5, 7, 6, 8, 7, 9, 8)

# Membuat data frame
data_regresi <- data.frame(y_dependent, x1, x2, x3)

# Menampilkan beberapa baris pertama data
print("Data yang digunakan:")
print(head(data_regresi))
```

```
# --- Menjalankan Model Regresi Linear ---
# Formula: y_dependent ~ x1 + x2 + x3 berarti kita memprediksi y_dependent
# berdasarkan kombinasi linear dari x1, x2, dan x3.
# R akan otomatis menambahkan intercept.
model <- lm(y_dependent ~ x1 + x2 + x3, data = data_regresi)


# --- Menampilkan Ringkasan Hasil Model ---
# summary() memberikan output yang jauh lebih lengkap daripada coef()
summary(model)
```

Regresi Linier - Implementasi OLS - Output R

```
Call:
lm(formula = y_dependent ~ x1 + x2 + x3, data = data_regresi)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.7800 -0.1600 -0.0750  0.3175  0.5100 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -2.4000     3.1702  -0.757  0.47768   
x1            0.2500     0.5587   0.447  0.67027   
x2            2.4000     0.6096   3.937  0.00765 **
x3            1.0900     0.5848   1.864  0.11160   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4722 on 6 degrees of freedom
Multiple R-squared:  0.9958,	Adjusted R-squared:  0.9937 
F-statistic: 470.8 on 3 and 6 DF,  p-value: 1.653e-07
```

Regresi Linier - Implementasi WLS

Pertanyaan	Langkah
Model apa yang harus saya mulai?	Selalu mulai dengan OLS.
Bagaimana cara memeriksa masalah?	Dapatkan sisaan dari model OLS.
Bagaimana cara menemukan sumber masalah?	Plot sisaan melawan setiap variabel X secara terpisah.
Variabel mana yang saya pilih untuk WLS?	Pilih variabel X yang plotnya menunjukkan pola corong/kipas yang jelas.
Apa yang harus dilakukan jika tidak ada yang jelas?	Plot sisaan vs. $\hat{y}$. Jika berpola, gunakan $\hat{y}$ sebagai dasar bobot. Jika tidak, pertimbangkan Robust Standard Errors.

Regresi Linier - Implementasi WLS

UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

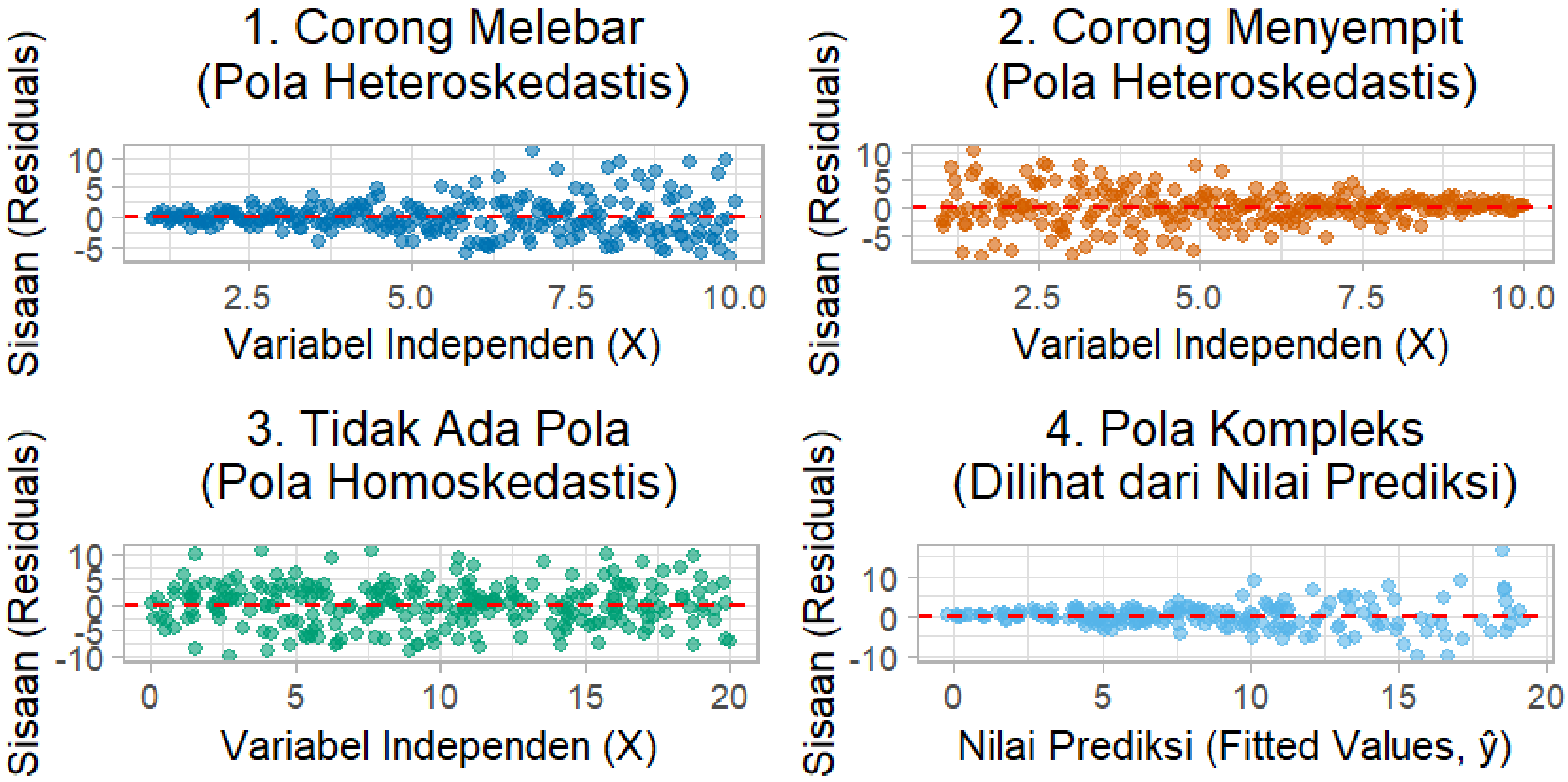
Jujur Adil Wibawa Amanah Religius Akuntabel

Bagaimana cara menentukan bobot matrix W?

Pola Visual pada Plot Sisaan	Deskripsi & Interpretasi	Kemungkinan Penyebab Utama	Rekomendasi Bobot W
Corong Melebar (Klasik): Sebaran sisaan melebar ke kanan.	Varians sisaan (σ^2) meningkat seiring dengan meningkatnya nilai variabel di sumbu X.Ketidakpastian model bertambah besar pada nilai X yang tinggi.	Satu variabel independen (misal, X1) adalah penyebab utamanya. (Plot sisaan vs. X lain terlihat acak).	Invers dari variabel penyebab. w = 1 / X_penyebab Contoh: w = 1 / df$x1^2 atau w = 1 / df$x1
Corong Menyempit (Terbalik): Sebaran sisaan menyempit ke kanan.	Varians sisaan (σ^2) menurun seiring dengan meningkatnya nilai variabel di sumbu X. Model menjadi lebih presisi pada nilai X yang tinggi.	Satu variabel independen (misal, X2) adalah penyebabnya. (Contoh nyata: Pengalaman kerja).	Proporsional dengan variabel penyebab. w = X_penyebab. Contoh: w = df$x2 atau w = df$x2^2
Corong Kompleks / Ambigu: Pola samar di beberapa plot X, TAPI JELAS di plot Sisaan vs. Nilai Prediksi ($\hat{y}$).	Varians sisaan (σ^2) adalah fungsi dari kombinasi beberapa variabel X, bukan hanya satu. Pola ini paling baik ditangkap oleh nilai prediksi.	Efek gabungan dari beberapa variabel independen (X1, X2, dst.) secara bersamaan.	Invers dari kuadrat nilai prediksi ($\hat{y}^2$). w = 1 / $\hat{y}^2$ Contoh: w = 1 / fitted(model_ols)^2
Tidak Ada Pola (Awan Acak): Sebaran titik terlihat konstan dan acak di semua plot.	Tidak ada bukti visual adanya heteroskedastisitas. Asumsi homoskedastisitas kemungkinan besar terpenuhi.	Tidak ada masalah heteroskedastisitas yang perlu dikhawatirkan.	Tidak perlu WLS. Tetap gunakan OLS (Ordinary Least Squares).

Regresi Linier - Implementasi WLS

Bagaimana cara menentukan bobot matrix W?



Regresi Linier - Implementasi WLS

Anggap pola sisaan Corong Melebar
terhadap x1. Secara matematis,

$$Var(e_i) = \sigma^2 x_{i1}.$$

Maka,

$$w_i = \frac{1}{x_{i1}}.$$

$$\mathbf{W} = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.333 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.167 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.143 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.125 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.111 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}$$

Observasi (i)	Nilai xi1	Bobot wi=1/xi1
1	1	1/1 = 1.0
2	2	1/2 = 0.5
3	3	1/3 ≈ 0.333
4	4	1/4 = 0.25
5	5	1/5 = 0.2
6	6	1/6 ≈ 0.167
7	7	1/7 ≈ 0.143
8	8	1/8 = 0.125
9	9	1/9 ≈ 0.111
10	10	1/10 = 0.1

Regresi Linier - Implementasi WLS

$$\mathbf{y} = \begin{bmatrix} 8.1 \\ 9.5 \\ 10.2 \\ 13.5 \\ 12.9 \\ 17.8 \\ 18.1 \\ 22.0 \\ 21.5 \\ 25.4 \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} 1 & 1 & 2 & 5 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 2 & 6 \\ 1 & 4 & 4 & 5 \\ 1 & 5 & 3 & 7 \\ 1 & 6 & 5 & 6 \\ 1 & 7 & 4 & 8 \\ 1 & 8 & 6 & 7 \\ 1 & 9 & 5 & 9 \\ 1 & 10 & 7 & 8 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.333 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.25 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.167 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.143 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.125 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.111 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}$$

$$\hat{\beta}_{WLS} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}(\mathbf{X}'\mathbf{W}\mathbf{y})$$

$$(X'WX) = \begin{bmatrix} 2.93 & 10 & 9.18 & 16.47 \\ 10.00 & 55 & 41.00 & 65.00 \\ 9.18 & 41 & 34.26 & 54.62 \\ 16.47 & 65 & 54.62 & 97.72 \end{bmatrix} \qquad (X'Wy) = \begin{bmatrix} 35.44 \\ 159.00 \\ 129.89 \\ 214.39 \end{bmatrix}$$

$$(X'WX)^{-1} = \begin{bmatrix} 90.72 & 14.07 & -16.91 & -15.19 \\ 14.07 & 2.35 & -2.83 & -2.35 \\ -16.91 & -2.83 & 3.68 & 2.68 \\ -15.19 & -2.35 & 2.68 & 2.63 \end{bmatrix} \qquad \hat{\beta} = \begin{bmatrix} -2.3527 \\ 0.2797 \\ 2.3263 \\ 1.1041 \end{bmatrix}$$

Regresi Linier - Implementasi WLS - Syntax R

```r
# --- Menyiapkan Data ---
# Vektor variabel dependen (y)
y_dependent <- c(8.1, 9.5, 10.2, 13.5, 12.9, 17.8, 18.1, 22.0, 21.5, 25.4)

# Vektor-vektor variabel independen (x)
# Diambil dari kolom 2, 3, dan 4 pada matriks X sebelumnya
x1 <- 1:10
x2 <- c(2, 3, 2, 4, 3, 5, 4, 6, 5, 7)
x3 <- c(5, 4, 6, 5, 7, 6, 8, 7, 9, 8)

# Membuat data frame
data_regresi <- data.frame(y_dependent, x1, x2, x3)

# Menampilkan beberapa baris pertama data
print("Data yang digunakan:")
print(head(data_regresi))
```

```r
# Membuat vektor bobot (bukan matriks)
bobot_vector <- 1 / data_regresi$x1
# Menjalankan regresi WLS menggunakan lm() dengan argumen
'weights'
model_WLS <- lm(y ~ x1 + x2 + x3, data = data_regresi, weights =
bobot_vector)
summary(model_WLS)
```

Regresi Linier - Implementasi WLS - Output R


```
Call:
lm(formula = y ~ x1 + x2 + x3, data = data_regresi, weights = bobot_vector)

Weighted Residuals:
     Min       1Q   Median       3Q      Max 
-0.38158 -0.06680 -0.02293  0.12350  0.25204 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)  -2.3527     2.0361  -1.156   0.2918   
x1            0.2797     0.3277   0.854   0.4261   
x2            2.3263     0.4103   5.670   0.0013 **
x3            1.1041     0.3472   3.179   0.0191 * 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2138 on 6 degrees of freedom
Multiple R-squared:  0.9962,	Adjusted R-squared:  0.9942 
F-statistic: 519.7 on 3 and 6 DF,  p-value: 1.231e-07
```

Regresi Linier - Implementasi GLS

Anggap pola sisaan berAutokorelasi

AR(1) dengan ρ rho=0.7.

$$\Omega_{ij} = \rho^{|i-j|}$$

Di mana:

rho adalah koefisien autokorelasi (kita gunakan 0.7).

|i-j| adalah jarak absolut (selalu positif) antara observasi ke-i dan ke-j.

Ini berarti:

Diagonalnya (i=j) akan selalu $0.7^0 = 1$

Elemen di sebelah diagonal (∣ i−j ∣ =1) akan bernilai $0.7^1 = 0.7$

Elemen yang terpisah 2 langkah (∣ i−j ∣ =2) akan bernilai $0.7^2 = 0.49$ dan seterusnya.

UNIVERSITAS SULTAN AGENG TIRTAYASA

Regresi Linier - Implementasi GLS

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

Matrix Omega Ω dengan rho=0.7.

$$\hat{\beta}_{GLS} = (\mathbf{X}'\mathbf{\Omega}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{\Omega}^{-1}\mathbf{y}$$

Ω **(n x n)**

1.000	0.700	0.490	0.343	0.240	0.168	0.118	0.082	0.058	0.040
0.700	1.000	0.700	0.490	0.343	0.240	0.168	0.118	0.082	0.058
0.490	0.700	1.000	0.700	0.490	0.343	0.240	0.168	0.118	0.082
0.343	0.490	0.700	1.000	0.700	0.490	0.343	0.240	0.168	0.118
0.240	0.343	0.490	0.700	1.000	0.700	0.490	0.343	0.240	0.168
0.168	0.240	0.343	0.490	0.700	1.000	0.700	0.490	0.343	0.240
0.118	0.168	0.240	0.343	0.490	0.700	1.000	0.700	0.490	0.343
0.082	0.118	0.168	0.240	0.343	0.490	0.700	1.000	0.700	0.490
0.058	0.082	0.118	0.168	0.240	0.343	0.490	0.700	1.000	0.700
0.040	0.058	0.082	0.118	0.168	0.240	0.343	0.490	0.700	1.000

Ω^{-1} **(n x n)**

1.961	-1.373	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-1.373	2.922	-1.373	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	-1.373	2.922	-1.373	0.000	0.000	0.000	0.000	0.000	0.000
0.000	0.000	-1.373	2.922	-1.373	0.000	0.000	0.000	0.000	0.000
0.000	0.000	0.000	-1.373	2.922	-1.373	0.000	0.000	0.000	0.000
0.000	0.000	0.000	0.000	-1.373	2.922	-1.373	0.000	0.000	0.000
0.000	0.000	0.000	0.000	0.000	-1.373	2.922	-1.373	0.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	-1.373	2.922	-1.373	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.373	2.922	-1.373
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	-1.373	1.961

$$(X'\Omega^{-1}X) = \begin{bmatrix} 2.588 & 14.235 & 10.941 & 16.824 \\ 14.235 & 121.882 & 83.450 & 108.824 \\ 10.941 & 83.451 & 84.706 & 53.255 \\ 16.824 & 108.824 & 53.255 & 144.000 \end{bmatrix}$$

$$(X'\Omega^{-1}X)^{-1} = \begin{bmatrix} 24.107 & 3.562 & -4.118 & -3.986 \\ 3.562 & 0.618 & -0.669 & -0.636 \\ -4.118 & -0.669 & 0.759 & 0.706 \\ -3.986 & -0.636 & 0.706 & 0.692 \end{bmatrix}$$

$$(X'\Omega^{-1}y) = \begin{bmatrix} 41.853 \\ 313.998 \\ 255.833 \\ 270.273 \end{bmatrix}$$

$$\hat{\beta} = \begin{bmatrix} -3.128 \\ 0.1008 \\ 2.558 \\ 1.220 \end{bmatrix}$$

Regresi Linier - Implementasi GLS - Syntax R

```
# --- Menyiapkan Data ---
# Vektor variabel dependen (y)
y_dependent <- c(8.1, 9.5, 10.2, 13.5, 12.9, 17.8, 18.1, 22.0, 21.5, 25.4)

# Vektor-vektor variabel independen (x)
# Diambil dari kolom 2, 3, dan 4 pada matriks X sebelumnya
x1 <- 1:10
x2 <- c(2, 3, 2, 4, 3, 5, 4, 6, 5, 7)
x3 <- c(5, 4, 6, 5, 7, 6, 8, 7, 9, 8)

# Membuat data frame
data_regresi <- data.frame(y_dependent, x1, x2, x3)

# Menampilkan beberapa baris pertama data
print("Data yang digunakan:")
print(head(data_regresi))
```

```
# Membuat dataframe, termasuk kolom 'time' untuk urutan korelasi
data_model <- data.frame(y = y, x1 = X[,"X1"], x2 = X[,"X2"], x3 = X[,"X3"],
time = 1:nrow(X))

# Membuat Matriks Omega (Ω)
n <- nrow(X)
rho <- 0.7
Omega <- outer(1:n, 1:n, function(i, j) rho^(abs(i - j)))

# Ekstrak nilai korelasi dari segitiga bawah matriks Omega
initial_corrs <- Omega[lower.tri(Omega)]

# Buat struktur korelasi kustom dengan corSymm
# fixed = TRUE adalah kuncinya: ini memaksa gls menggunakan nilai kita
custom_correlation <- corSymm(value = initial_corrs, form = ~time, fixed =
TRUE)

# Jalankan model gls
model_GLS <- gls(y ~ x1 + x2 + x3, data = data_model,
                 correlation = custom_correlation)
summary(model_GLS)
```

Regresi Linier - Implementasi GLS - Output R

```
Generalized least squares fit by REML
  Model: y ~ x1 + x2 + x3
  Data: data_model
       AIC      BIC    logLik
  26.42444 25.38324 -8.212222

Correlation Structure: General
 Formula: ~time
 Parameter estimate(s):
 Correlation:
   1     2     3     4     5     6     7     8     9
2  0.700
3  0.490 0.700
4  0.343 0.490 0.700
5  0.240 0.343 0.490 0.700
6  0.168 0.240 0.343 0.490 0.700
7  0.118 0.168 0.240 0.343 0.490 0.700
8  0.082 0.118 0.168 0.240 0.343 0.490 0.700
9  0.058 0.082 0.118 0.168 0.240 0.343 0.490 0.700
10 0.040 0.058 0.082 0.118 0.168 0.240 0.343 0.490 0.700

Coefficients:
                 Value Std.Error   t-value p-value
(Intercept) -3.1275584  3.857149 -0.810847  0.4484
x1           0.1008031  0.617545  0.163232  0.8757
x2           2.5577973  0.684450  3.737014  0.0097
x3           1.2201675  0.653436  1.867309  0.1111
```

Regresi Linier - Perbandingan OLS, WLS, GLS

Kita **BELUM BISA** membandingkan ke tiga model tersebut dengan menggunakan Set Data Contoh yang sudah diberikan sebelumnya..

Kenapa???

Karena daritadi kita menyelesaikan jawaban dengan SET ASUMSI, BUKAN berdasarkan Kenyataan diagnostik Sisaan.

Regresi Linier - Perbandingan OLS, WLS, GLS

Silakan Buka Raw Data pada :

LINK

Regresi Linier - Perbandingan OLS, WLS, GLS

```r
# Memuat Semua Paket yang Diperlukan
# install.packages(c("lmtest", "car", "nlme",
"readxl")) # Jalankan jika belum terinstal
library(lmtest)
library(car)
library(nlme)
library(readxl)

# MENGIMPOR DATA DARI FILE EXCEL
cat("Mengimpor data dari file Excel...\n")
# Tentukan path lengkap ke file Excel Anda
file_path <- "C:/Users/user/OneDrive -
untirta.ac.id/R Script/data_simulasi_gls.xlsx"
# Membaca data dari file Excel dan
menyimpannya ke dataframe 'data_final'
data_final <- read_excel(file_path)

cat("Data berhasil diimpor dari:", file_path, "\n")
cat("Berikut adalah 6 baris pertama data yang
diimpor:\n")
print(head(data_final))
cat("\n\n")
```

Regresi Linier - Perbandingan OLS, WLS, GLS

```
# (UJI DIAGNOSTIK PADA OLS)
cat("Menjalankan 4 Uji Diagnostik pada Model OLS...\n")
model_ols_final <- lm(y ~ x1 + x2 + x3, data = data_final)
cat("\n--- Uji Normalitas (Shapiro-Wilk) ---\n")
print(shapiro.test(residuals(model_ols_final)))
cat("\n--- Uji Homoskedastisitas (Breusch-Pagan) ---\n")
print(bptest(model_ols_final))
cat("\n--- Uji Autokorelasi (Durbin-Watson) ---\n")
print(dwtest(model_ols_final))
cat("\n--- Uji Multikolinearitas (VIF) ---\n")
print(vif(model_ols_final))
cat("\nPembuktian selesai.\n\n")
```

```
# ANALISIS DENGAN OLS, WLS, DAN
GLS
# MODEL WLS
bobot <- 1 / data_final$x1^2 # Ambil x1
dari dataframe yang diimpor
model_wls_final <- lm(y ~ x1 + x2 + x3,
data = data_final, weights = bobot)
# MODEL GLS
model_gls_final <- gls(y ~ x1 + x2 + x3,
data = data_final,
                weights = varFixed(~x1),
                correlation = corAR1(form
= ~time))
```

Regresi Linier - Perbandingan OLS, WLS, GLS

```
# PERBANDINGAN AKHIR DAN KESIMPULAN
cat("--- 1. Hasil Model OLS (Baseline, Kesimpulan Tidak Valid) ---\n")
cat("----------------------------------------------------------------------\n")
print(summary(model_ols_final))
cat("--- 2. Hasil Model WLS (Perbaikan Parsial, Masih Kurang Tepat) ---\n")
cat("----------------------------------------------------------------------\n")
print(summary(model_wls_final))
cat("--- 3. Hasil Model GLS (Metode Paling Tepat dan Akurat) ---\n")
cat("----------------------------------------------------------------------\n")
print(summary(model_gls_final))
```

Regresi Linier - Perbandingan OLS, WLS, GLS

UNIVERSITAS SULTAN AGENG TIRTAYASA

UNTIRTA Jawara

Jujur Adil Wibawa Amanah Religius Akuntabel

```
cat("\n\n--- Perbandingan Model Menggunakan AIC & BIC (Nilai lebih kecil
lebih baik) ---\n")
cat("\n--- Model OLS ---\n")
print(paste("AIC:", round(AIC(model_ols_final), 2)))
print(paste("BIC:", round(BIC(model_ols_final), 2)))
cat("\n--- Model WLS ---\n")
print(paste("AIC:", round(AIC(model_wls_final), 2)))
print(paste("BIC:", round(BIC(model_wls_final), 2)))
cat("\n--- Model GLS ---\n")
print(paste("AIC:", round(AIC(model_gls_final), 2)))
print(paste("BIC:", round(BIC(model_gls_final), 2)))
```

Regresi Linier - Perbandingan OLS, WLS, GLS

Output OLS

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = data_final)

Residuals:
    Min      1Q  Median      3Q     Max 
-331.73  -52.72    3.07   72.89  220.79 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   0.8581    70.5060   0.012    0.990    
x1            1.8305     0.3819   4.793    6e-06 ***
x2            0.4335     0.6377   0.680    0.498    
x3            1.2326     1.1471   1.075    0.285    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 109.8 on 96 degrees of freedom
Multiple R-squared:  0.1986,	Adjusted R-squared:  0.1735 
F-statistic: 7.929 on 3 and 96 DF,  p-value: 8.86e-05
```

Regresi Linier - Perbandingan OLS, WLS, GLS

Output WLS

```
Call:
lm(formula = y ~ x1 + x2 + x3, data = data_final, weights = bobot)

Weighted Residuals:
    Min      1Q  Median      3Q     Max 
-5.1444 -1.5416  0.1138  1.6125  4.7730 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   7.2586    14.7750   0.491 0.624354    
x1            2.9778     0.2481  12.005  < 2e-16 ***
x2           -0.4109     0.2392  -1.718 0.089029 .  
x3            1.1321     0.2969   3.814 0.000242 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.188 on 96 degrees of freedom
Multiple R-squared:  0.6423,    Adjusted R-squared:  0.6311 
F-statistic: 57.45 on 3 and 96 DF,  p-value: < 2.2e-16
```

Regresi Linier - Perbandingan OLS, WLS, GLS

Output GLS

```
Generalized least squares fit by REML
  Model: y ~ x1 + x2 + x3
  Data: data_final
       AIC      BIC    logLik
  1114.512 1129.898 -551.2558

Correlation Structure: AR(1)
 Formula: ~time
 Parameter estimate(s):
      Phi
0.7411009
Variance function:
 Structure: fixed weights
 Formula: ~x1
```

```
Coefficients:
                Value Std.Error   t-value p-value
(Intercept) -13.134945  31.86651 -0.412187  0.6811
x1            2.704892   0.61493  4.398704  0.0000
x2           -0.579660   0.24559 -2.360311  0.0203
x3            1.764696   0.42832  4.120035  0.0001

 Correlation:
   (Intr) x1     x2
x1 -0.331
x2 -0.492  0.103
x3 -0.740  0.098 -0.003

Standardized residuals:
       Min         Q1        Med         Q3        Max
-2.6228241 -0.4967329  0.1501300  0.7665890  1.9164083

Residual standard error: 15.24417
Degrees of freedom: 100 total; 96 residual
```

Regresi Linier - Perbandingan OLS, WLS, GLS

Perbandingan Kebaikan Model dengan AIC/ BIC

```
--- Model OLS ---
> print(paste("AIC:", round(AIC(model_ols_final), 2)))
[1] "AIC: 1229.46"
> print(paste("BIC:", round(BIC(model_ols_final), 2)))
[1] "BIC: 1242.48"
> cat("\n--- Model WLS ---\n")

--- Model WLS ---
> print(paste("AIC:", round(AIC(model_wls_final), 2)))
[1] "AIC: 1173.75"
> print(paste("BIC:", round(BIC(model_wls_final), 2)))
[1] "BIC: 1186.77"
> cat("\n--- Model GLS ---\n")

--- Model GLS ---
> print(paste("AIC:", round(AIC(model_gls_final), 2)))
[1] "AIC: 1114.51"
> print(paste("BIC:", round(BIC(model_gls_final), 2)))
[1] "BIC: 1129.9"
```

SEE YOU NEXT WEEK !

Ferdian Bangkit Wijaya, S.Stat., M.Si
NIP. 199005202024061001
ferdian.bangkit@untirta.ac.id