Nama : Nana Oktaviana
NIM : G1401201006

Ujian Akhir Semester STA1372 - Metode Simulasi Resampling

A. Latar Belakang

Menurut Gujarati (2004), analisis regresi adalah suatu metode statistik yang digunakan untuk mengukur dan memodelkan hubungan antara variabel dependen (variabel yang ingin diprediksi) dan satu atau lebih variabel independen (variabel penjelas). Persamaan regresi linear dapat ditulis sebagai:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_\mathrm{p} x_\mathrm{p} + \varepsilon$$

dengan y adalah variabel dependen, $x_1, x_2, \ldots, x_\mathrm{p}$ adalah variabel independen, $\beta_0, \beta_1, \beta_2, \ldots, \beta_\mathrm{p}$ adalah parameter yang ingin diestimasi, dan ε adalah galat atau sisaan (residual) yang merupakan perbedaan antara nilai observasi dan nilai prediksi (Montgomery 2012).

Metode kuadrat terkecil bekerja dengan mencari estimasi parameter $\beta_0, \beta_1, \beta_2, \ldots, \beta_\mathrm{p}$ yang meminimalkan jumlah kuadrat galat (Kutner 2004). MKT dapat digunakan apabila asumsi-asumsi pada analisis regresi terpenuhi. Salah satu asumsi penting yang harus dipenuhi adalah asumsi kenormalan (Astari 2014).

Dalam prakteknya, model regresi seringkali dihadapkan pada situasi sebaran sisaan yang bervariasi. Berbagai metode digunakan untuk mengatasi masalah tersebut, salah satunya dengan menggunakan Simulasi Monte Carlo. Alijoyo et al. (2019) mengemukakan bahwa Monte Carlo Simulation (MCS) adalah suatu metode simulasi kuantitatif yang digunakan untuk mengevaluasi risiko dengan menghitung probabilitas hasil akhir dalam menghadapi ketidakpastian. Metode ini melibatkan penggunaan variabel acak (random variable) berdasarkan karakteristik distribusi input atau data yang sedang dianalisis. Menurut Gentle (2005), untuk melakukan prediksi menggunakan Monte Carlo, diperlukan pengujian berulang-ulang pada data yang sama dengan menggunakan bilangan acak yang berbeda tapi memiliki keseragaman sehingga informasi dapat dihasilkan lebih efisien.

B. Tujuan

Simulasi ini bertujuan memeriksa performa pendugaan dengan metode kuadrat terkecil pada model regresi dengan berbagai situasi sebaran sisaan.

C. Data

Data yang digunakan dalam simulasi ini merupakan data hasil bangkitan menggunakan aplikasi perangkat lunak R.

D. Metodologi

Langkah-langkah simulasi yang dilakukan adalah sebagai berikut.

1. Menghitung estimasi koefisien, SSE, MSE, dan R-squared.

2. Menentukan jumlah simulasi dan ukuran sampel.
3. Menentukan skenario distribusi residual dengan variansi residual tertentu.
4. Membuat matriks "results" untuk menyimpan hasil perhitungan.
5. Melakukan simulasi Monte Carlo dengan menghasilkan data x dan residual, menghitung estimasi koefisien, SSE, MSE, dan R-squared, dan menyimpan hasilnya dalam matriks "results".
6. Menampilkan hasil rata-rata dan variansi dari matriks "results" untuk evaluasi.

E. Hasil dan Pembahasan

```
> mean.results
       Normal SSE        Normal MSE  Normal R-squared       Poisson SSE
       24.2384081         0.2423841         0.9411726        24.6982047
      Poisson MSE Poisson R-squared  Normal Intercept      Normal Slope
        0.2469820         0.9402911         4.9956549         1.9992493
Poisson Intercept     Poisson Slope
        1.9999028                NA
> var.results
       Normal SSE        Normal MSE  Normal R-squared       Poisson SSE
     1.132635e+01      1.132635e-03      1.316957e-04      3.691526e+01
      Poisson MSE Poisson R-squared  Normal Intercept      Normal Slope
     3.691526e-03      2.696019e-04      2.427499e-03      2.820613e-03
Poisson Intercept     Poisson Slope
     2.842840e-03                NA
```

Berdasarkan hasil tersebut, dapat dilihat bahwa performa pendugaan dengan metode kuadrat terkecil pada model regresi dapat bervariasi tergantung pada situasi sebaran sisaan yang digunakan.

1. SSE (Sum of Squared Errors):

   Model regresi dengan distribusi residual normal memiliki SSE yang lebih rendah (24.2384081) dibandingkan dengan model regresi dengan distribusi residual Poisson (24.6982047). Hal ini menunjukkan bahwa model regresi dengan distribusi residual normal memiliki kesalahan prediksi yang lebih kecil.

2. MSE (Mean Squared Error):

   MSE dari model regresi dengan distribusi residual normal (0.2423841) lebih rendah dibandingkan dengan MSE dari model regresi dengan distribusi residual Poisson (0.2469820). Hal ini menunjukkan bahwa model regresi dengan distribusi residual normal memiliki rata-rata kesalahan prediksi yang lebih kecil.

3. R-squared:

   R-squared dari model regresi dengan distribusi residual normal (0.9411726) lebih tinggi dibandingkan dengan R-squared dari model regresi dengan distribusi residual Poisson (0.9402911). Hal ini menunjukkan bahwa model regresi dengan distribusi residual normal lebih baik dalam menjelaskan variasi dalam data.

4. Estimasi Koefisien:

Estimasi koefisien intercept dan slope pada model regresi dengan distribusi residual normal dan Poisson cukup dekat. Namun, perlu diperhatikan bahwa estimasi koefisien slope pada model regresi dengan distribusi residual Poisson tidak tersedia (NA).

5. Variansi:

   Variansi SSE, MSE, dan R-squared pada kedua model menunjukkan variasi yang signifikan selama simulasi Monte Carlo. Namun, variansi dari SSE dan MSE pada model regresi dengan distribusi residual Poisson cenderung lebih tinggi dibandingkan dengan model regresi dengan distribusi residual normal.

F. Simpulan

Model regresi dengan distribusi residual normal cenderung memberikan performa pendugaan yang lebih baik, dengan SSE, MSE, dan R-squared yang lebih rendah dibandingkan dengan model regresi dengan distribusi residual Poisson. Namun, perlu diperhatikan bahwa pilihan distribusi residual tergantung pada karakteristik data dan asumsi model yang digunakan.

G. Daftar Pustaka

Alijoyo A, Wijaya B, Jacob I. 2019. *Monte Carlo Simulation*. Jakarta (ID) : CRMS.

Astari NMM. Suciptawati NLP. Sukarsa IKG. 2014. Penerapan metode bootstrap residual dalam mengatasi bias pada penduga parameter analisis regresi. *E-Jurnal Matematika*. 3(4):130-137.

Gentle JE. 2005. *Random Number Generation and Monte Carlo Methods*. New York (US): Springer Science Business Media, Inc.

Gujarati DN, Porter DC. 2009. *Basic Econometrics (5th Edition)*. New York (US): McGraw-Hill.

Kutner MH. Nachtsheim CJ. Neter J. Li W. 2004. *Applied Linear Statistical Models (5th Edition)*. New York (US): McGraw-Hill.

Montgomery DC. Peck EA. Vining GG. 2012. *Introduction to Linear Regression Analysis (5th Edition)*. John Wiley & Sons.

H. Lampiran

```
library(MASS)


# Fungsi untuk menghitung estimasi koefisien menggunakan MKT

est.coeff <- function(x, y) {

  lm_model <- lm(y ~ x)

  coefficients(lm_model)

}


# menghitung SSE
```

```
cal.SSE <- function(x, y, coefficients) {
  fitted_values <- coefficients[1] + coefficients[2] * x
  SSE <- sum((y - fitted_values)^2)
  SSE
}

# Menghitung MSE
cal.MSE <- function(x, y, coefficients) {
  SSE <- cal.SSE(x, y, coefficients)
  MSE <- SSE / length(y)
  MSE
}

# Menghitung R-squared
cal.R.sq <- function(x, y, coefficients) {
  fitted_values <- coefficients[1] + coefficients[2] * x
  SSE <- cal.SSE(x, y, coefficients)
  SST <- sum((y - mean(y))^2)
  R.squared <- 1 - SSE / SST
  R.squared
}

# Mengatur jumlah simulasi dan ukuran sampel
num.simul <- 1000
sample.size <- 100

# Menetapkan skenario distribusi residual dengan variansi residual
res.var <- 0.25

# Membuat matriks untuk menyimpan hasil SSE, MSE, R-squared, dan koefisien.
```

```
results <- matrix(0, nrow = num.simul, ncol = 10)

# Simulasi Monte Carlo
for (i in 1:num.simul) {
  # Membangkitkan data
  x <- rnorm(sample.size)

  # Membangkitkan sisaan dengan distribusi normal
  res.norm <- rnorm(sample.size, mean = 0, sd = sqrt(res.var))

  # Membangkitkan sisaan dengan distribusi Poisson
  res.pois <- rpois(sample.size, lambda = res.var)

  # Membangkitkan respon data
  y.norm <- 5 + 2 * x + res.norm
  y.poiss <- 5 + 2 * x + res.pois

  # Menghitung koefisien estimasi
  coeff.nor <- est.coeff(x, y.norm)
  coeff.poiss <- est.coeff(x, y.poiss)

  # Menghitung SSE, MSE, dan R-squared
  SSE.normal <- cal.SSE(x, y.norm, coeff.nor)
  MSE.normal <- cal.MSE(x, y.norm, coeff.nor)
  R.squared.normal <- cal.R.sq(x, y.norm, coeff.nor)

  SSE.poisson <- cal.SSE(x, y.poiss, coeff.poiss)
  MSE.poisson <- cal.MSE(x, y.poiss, coeff.poiss)
  R.squared.poisson <- cal.R.sq(x, y.poiss, coeff.poiss)
```

```
  # Menyimpan hasil SSE, MSE, dan R-squared
  results[i, 1] <- SSE.normal
  results[i, 2] <- MSE.normal
  results[i, 3] <- R.squared.normal
  results[i, 4] <- SSE.poisson
  results[i, 5] <- MSE.poisson
  results[i, 6] <- R.squared.poisson

  # Menyimpan hasil koefisien
  results[i, 7] <- coeff.nor[1] # Normal intercept
  results[i, 8] <- coeff.nor[2] # Normal slope
  results[i, 9] <-  coeff.poiss[2] # Poisson intercept
  results[i, 10] <-  coeff.poiss[3] # Poisson slope
}

# Menampilkan hasil
colnames(results) <- c("Normal SSE", "Normal MSE", "Normal R-squared", "Poisson SSE", "Poisson MSE", "Poisson R-squared", "Normal Intercept", "Normal Slope", "Poisson Intercept", "Poisson Slope")
mean.results <- colMeans(results)
var.results <- apply(results, 2, var)

mean.results
var.results
```