PRAKTIKUM KOMSTAT MODUL 3

ANDRIAN AGUTINUS LUMBAN GAOL

2023-11-08

# Kerjakan Latihan dibawah ini dengan menggunakan metode diatas dan bandingkan hasilnya dengan menggunaka integrate

## nilai sebenarnya

x<-function(x) x^4\*exp(-x)  
integrate(x,0,5)$value

## [1] 13.42816

##TRAPEZOID

trapezoid<-function(f,a,b){  
 hasil <- ((b-a)/2) \* (f(a) + f(b))  
 return (hasil)  
}  
  
f<-function(x) x^4\*exp(-x)  
trapezoid(f,0,5)

## [1] 10.52804

##SIMPSON 1/3

simpson<-function(f,a,b){  
 hasil <- ((b-a)/6) \* (f(a) + 4\*f((a+b)/2) + f(b))  
 return (hasil)  
}  
f<-function(x) x^4\*exp(-x)  
simpson(f,0,5)

## [1] 14.1975

##SIMPSON 3/8

simpson38<-function(f,a,b){  
 h <- (b-a)/3  
 hasil <- (((b-a)/8) \* (f(a) + 3\*f(a+h) + 3\*f(b-h) + f(b)))  
 return (hasil)  
}  
f<-function(x) x^4\*exp(-x)  
simpson38(f,0,5)

## [1] 13.62245

##BOOLE

boole<-function(f,a,b){  
 h<-(b-a)/4  
 hasil <- ((b-a)/90) \* ((7\*f(a) + 32\*f(a+h) + 12\*f((a+b)/2) + 32\*f(b-h) + 7\*f(b)))  
 return (hasil)  
}  
f<-function(x) x^4\*exp(-x)  
boole(f,0,5)

## [1] 13.28679

## Selisih

library(knitr)

## Warning: package 'knitr' was built under R version 4.2.3

#hasil  
y = 13.42816 # nilai sebenarnya  
a = 10.52804 # TRAPEZOID  
b = 14.1975 # SIMPSON 1/3  
c = 13.62245 # SIMPSON 3/8  
d = 13.28679 # BOOLE  
  
#selisih  
y\_1 = (y-y)  
a\_1 = (y-a)  
b\_1 = (y-b)  
c\_1 = (y-c)  
d\_1 = (y-d)  
  
# Tabel  
tabel <- data.frame(  
 metode = c("nilai sebenarnya", "TRAPEZOID", "SIMPSON 1/3", "SIMPSON 3/8" , "BOOLE"),  
 hasil = c(y, a, b, c, d),  
 selisih = c(y\_1, a\_1, b\_1, c\_1, d\_1)  
)  
  
kable(tabel)

| metode | hasil | selisih |
| --- | --- | --- |
| nilai sebenarnya | 13.42816 | 0.00000 |
| TRAPEZOID | 10.52804 | 2.90012 |
| SIMPSON 1/3 | 14.19750 | -0.76934 |
| SIMPSON 3/8 | 13.62245 | -0.19429 |
| BOOLE | 13.28679 | 0.14137 |

Dari tabel tersebut dapat dilihat bahwa metode boole memiliki selisih dari nilai sebenarnya yang paling kecil yaitu sebesar 0.14137, atau dengan kata lain metode ini yang paling mendekati nilai sebenarnya, sehingga dapat disimpulkan bahwa metode boole adalah metode yang paling optimal untuk mencari niali optimal dari fungsi $ \_{0}^{5} t4e-t ,dt $

# 2. Buatlah fungsi R yang dinamakan regresi.mle yang argumenya terdiri dari formulas, data dan optim\_method.

Argumen formulas merupakan rumus regresi yang biasanya ditulis dalam fungsilm, argumen data adalah data yang akan diregresikan dalam bentuk data.frame, dan optim\_method merupakan metode optimasi yang tersedia dalam optim. Argumen optim\_method memiliki nilai de- fault Nelder-Mead. Output fungsi ini berupa list yang berisi nilai estimasi koefisien, standard error koefisien dan simpangan baku regresi. Estimasi parameter pada regesi.mle ini dilakukan dengan menggunakan pendekatan MLE (Maximum Likelihood Estimation)

# Membuat fungsi regresi.mle  
regresi.mle <- function(formula, data, optim\_method = "Nelder-Mead") {  
 # Mengonversi formula menjadi model matrix  
 X <- model.matrix(formula, data)  
   
 # Mengekstrak variabel respons  
 Y <- data[, 1] # Mengasumsikan variabel respons ada di kolom pertama  
   
 # Fungsi untuk menghitung negative log-likelihood  
 neg\_log\_likelihood <- function(beta) {  
 mu <- X %\*% beta  
 sigma <- summary(lm(Y ~ X - 1))$sigma  
 ll <- sum(dnorm(Y, mean = mu, sd = sigma, log = TRUE))  
 return(-ll)  
 }  
   
 # Nilai awal untuk optimisasi  
 initial\_values <- rep(0, ncol(X))  
   
 # Estimasi Maximum Likelihood menggunakan optim  
 result <- optim(par = initial\_values, fn = neg\_log\_likelihood, method = optim\_method)  
   
 # Mengekstrak koefisien dan standard error  
 coef\_estimates <- result$par  
 se <- sqrt(diag(vcov(lm(Y ~ X - 1))))  
   
 # Membuat output berupa list  
 output <- list(  
 coef\_estimates = coef\_estimates,  
 standard\_error = se,  
 residual\_sd = summary(lm(Y ~ X - 1))$sigma  
 )  
   
 return(output)  
}

# Set seed for reproducibility  
set.seed(123)  
  
# Banyaknya observasi  
n <- 5  
  
# Menghasilkan nilai acak untuk x1, x2, dan x3  
x1 <- runif(n)  
x2 <- runif(n)  
x3 <- runif(n)  
  
# Membuat dataframe  
data <- data.frame(y = rnorm(n), x1, x2, x3)  
  
# Menampilkan dataframe  
head(data)

## y x1 x2 x3  
## 1 1.2805549 0.2875775 0.0455565 0.9568333  
## 2 -1.7272706 0.7883051 0.5281055 0.4533342  
## 3 1.6901844 0.4089769 0.8924190 0.6775706  
## 4 0.5038124 0.8830174 0.5514350 0.5726334  
## 5 2.5283366 0.9404673 0.4566147 0.1029247

formula\_regresi <- y ~ x1 + x2 + x3  
hasil <- regresi.mle(formula\_regresi, data = data)  
hasil

## $coef\_estimates  
## [1] 7.7499177 -5.5517950 -0.6391364 -5.2558160  
##   
## $standard\_error  
## X(Intercept) Xx1 Xx2 Xx3   
## 11.338484 9.173772 4.811599 8.878855   
##   
## $residual\_sd  
## [1] 2.747434

Output yang dihasilkan dari fungsi regresi.mle adalah sebuah list yang berisi estimasi parameter regresi, standard error, dan residual standar deviasi. dapat dilihat standar error yang paling rendah adalah x2, yang menandakan nilai x2 adalah estimasi yang paling baik dari yang lainnya. sementara nilai residual tidak terlalu rendah yaitu 2.747434, sehingga diperlukan perbaikan atau pengembangan fungsi kembali untuk menurunkan angka residual

# 3. Bangkitkan peubah Y, X1, X2 dan X3

berdasarkan model regresi linear berganda berikut ini:

dengan mengasumsikan bahwa ϵ N (0, 1). Banyaknya amatan yang dibangkitkan adalah 1000

# Mengatur seed agar hasil tetap sama setiap dijalankan  
set.seed(123)  
  
# Jumlah observasi  
n <- 1000  
  
# Menghasilkan variabel X1, X2, X3 dengan distribusi normal  
X1 <- rnorm(n)  
X2 <- rnorm(n)  
X3 <- rnorm(n)  
  
# Menghasilkan gangguan acak dengan distribusi normal (0, 1)  
epsilon <- rnorm(n, mean = 0, sd = 1)  
  
# Menghitung variabel respons Y berdasarkan model regresi linear berganda  
Y <- 10 + 3 \* X1 + 5 \* X2 + 7 \* X3 + epsilon  
  
# Membuat data.frame dari variabel yang dihasilkan  
df <- data.frame(Y = Y, X1 = X1, X2 = X2, X3 = X3)  
  
# Menampilkan beberapa baris pertama dari data.frame  
head(df)

## Y X1 X2 X3  
## 1 -0.3919541 -0.56047565 -0.99579872 -0.5116037  
## 2 5.4405003 -0.23017749 -1.03995504 0.2369379  
## 3 9.3469342 1.55870831 -0.01798024 -0.5415892  
## 4 17.3879585 0.07050839 -0.13217513 1.2192276  
## 5 1.4585907 0.12928774 -2.54934277 0.1741359  
## 6 16.0037690 1.71506499 1.04057346 -0.6152683

pada kode tersebut diterapkan set.seed untuk menjaga konsistensi hasil yang sama setiap menjalankan code. jumlah observasi adalah 1000 dengan variabel ,, berdistribusi normal, dan parameter model regresi sesuai dengan soal. kemudian variabel tersebut di buat kedalam suatu data frame yang nnati berisikan 1000 observasi yang dihasilkan berdasarkan model yang di buat.