

Regression Models

Table of contents

Kata Pengantar	3
Acknowledgements	3
Pendahuluan	4
Colophon	5
I Dasar-Dasar R	6
1 Install R dan RStudio	8
1.1 Install R	8
2 Pengantar R	10
2.1 Install R dan R Studio	10
2.2 Dasar-Dasar R	11
2.3 Tipe Data dalam Variabel	11
2.3.1 Operasi dengan Variabel	12
2.3.2 Menimpa dan Mengubah Nilai Variabel	12
2.4 Struktur Data pada R	12
2.4.1 Vektor	13
2.4.2 Matriks	13
2.4.3 Data Frame	13
2.4.4 List	14
2.4.5 Faktor	14
2.5 Fungsi dan Pemrograman di R	15
2.5.1 Definisi Fungsi	15
2.5.2 Pemanggilan fungsi	16
2.5.3 Argumen fungsi	16
2.5.4 Fungsi dengan Beberapa Nilai Kembalian	16
2.5.5 Fungsi Bawaan di R	17
2.5.6 Fungsi Anonim	17

Kata Pengantar

Buku ini ditujukan bagi mahasiswa, dosen, peneliti, serta praktisi yang ingin memahami dan mengaplikasikan model regresi menggunakan perangkat lunak R dengan pendekatan Quarto untuk dokumentasi analisis yang lebih interaktif dan reproducible.

Pemodelan regresi merupakan salah satu teknik statistik yang fundamental dalam berbagai disiplin ilmu, mulai dari ekonomi, sosial, teknik, hingga kesehatan. Dengan kemampuan R sebagai perangkat lunak open-source yang kuat, pengguna dapat melakukan analisis data secara efisien serta menyajikan hasil secara profesional menggunakan Quarto. Buku ini disusun secara sistematis mulai dari konsep dasar hingga implementasi dalam berbagai kasus nyata, disertai contoh kode dan interpretasi hasil.

Buku ini dilisensikan di bawah [CC BY-NC-ND 3.0](#) License. Jika Anda ingin mendapatkan salinan file PDF buku ini, Anda dapat memesannya di LYNK. Jika Anda menghargai kesempatan untuk membaca buku ini secara gratis dan ingin memberikan dukungan, silakan berdonasi ke LYNK.

Acknowledgements

Buku ini dihosting oleh <https://www.netlify.com> sebagai bagian dari dukungan mereka terhadap perangkat lunak open source dan komunitasnya.

Pendahuluan

Regresi merupakan salah satu teknik statistik yang paling banyak digunakan dalam berbagai bidang penelitian dan aplikasi industri. Metode ini memungkinkan kita untuk memahami hubungan antara variabel-variabel dalam suatu sistem, membuat prediksi, serta mengidentifikasi faktor-faktor yang berpengaruh terhadap variabel respons. Dari dunia ekonomi hingga ilmu kesehatan, pemodelan regresi telah menjadi alat utama dalam analisis data dan pengambilan keputusan berbasis bukti.

Dalam beberapa dekade terakhir, perkembangan teknologi komputasi dan statistik telah mengubah cara kita melakukan analisis data. Salah satu perkembangan yang paling signifikan adalah munculnya perangkat lunak open-source seperti R, yang menawarkan fleksibilitas tinggi, dukungan komunitas yang luas, serta berbagai pustaka statistik yang terus berkembang. R telah menjadi salah satu alat utama dalam dunia akademik dan industri untuk analisis data, termasuk dalam pemodelan regresi.

Buku ini disusun untuk memberikan pemahaman yang mendalam tentang model regresi serta bagaimana menerapkannya menggunakan R. Dengan pendekatan yang sistematis, buku ini mencakup konsep-konsep dasar regresi, mulai dari regresi linear sederhana hingga regresi non-linear dan model regresi tingkat lanjut. Selain itu, buku ini menggunakan **Quarto**, sebuah sistem dokumentasi modern yang memungkinkan analisis data yang lebih interaktif dan reproducible.

Keunggulan utama dari Quarto adalah kemampuannya dalam menggabungkan kode R dengan teks naratif dalam satu dokumen yang dapat di-render menjadi berbagai format, seperti HTML, PDF, atau dokumen Word. Hal ini memungkinkan pembaca untuk tidak hanya memahami teori di balik model regresi, tetapi juga melihat langsung bagaimana implementasinya dalam bentuk kode yang dapat dijalankan dan dimodifikasi sesuai kebutuhan.

Selain teori dan implementasi dasar, buku ini juga membahas berbagai studi kasus dari berbagai disiplin ilmu. Setiap studi kasus disajikan dengan pendekatan yang praktis, mulai dari eksplorasi data, pemilihan model yang tepat, interpretasi hasil, hingga validasi model. Pendekatan ini bertujuan agar pembaca tidak hanya memahami konsep secara teoretis, tetapi juga mampu menerapkannya dalam situasi nyata.

Salah satu tantangan utama dalam pemodelan regresi adalah pemilihan model yang sesuai dengan karakteristik data yang dianalisis. Overfitting, multikolinearitas, heteroskedastisitas, serta asumsi-asumsi dalam regresi adalah beberapa isu yang sering muncul dalam analisis data.

Buku ini membahas bagaimana cara mengidentifikasi serta mengatasi tantangan-tantangan tersebut menggunakan berbagai teknik statistik dan visualisasi dalam R.

Seiring dengan meningkatnya jumlah data yang tersedia saat ini, metode regresi juga terus berkembang. Model regresi klasik kini dilengkapi dengan pendekatan modern seperti **regularisasi (Ridge dan Lasso Regression)**, **regresi non-parametrik**, hingga **regresi berbasis machine learning**. Buku ini memperkenalkan berbagai metode tersebut dan bagaimana mereka dapat digunakan untuk meningkatkan akurasi serta interpretabilitas model.

Target pembaca buku ini mencakup mahasiswa, dosen, peneliti, serta praktisi di berbagai bidang yang tertarik untuk memahami dan menerapkan model regresi dengan R. Buku ini ditulis dengan bahasa yang sederhana dan disertai dengan banyak contoh sehingga dapat diikuti oleh pembaca dari berbagai latar belakang, baik yang baru mengenal R maupun yang sudah memiliki pengalaman dalam analisis data.

Kami berharap buku ini dapat menjadi referensi yang bermanfaat bagi pembaca dalam mengembangkan keterampilan analisis data mereka. Selain itu, dengan penggunaan Quarto sebagai alat utama dalam dokumentasi dan presentasi hasil analisis, kami berharap pembaca dapat mengadopsi praktik yang lebih transparan dan reproducible dalam penelitian mereka.

Akhir kata, kami mengucapkan terima kasih kepada semua pihak yang telah berkontribusi dalam penyusunan buku ini. Semoga buku ini dapat memberikan manfaat yang luas serta menjadi salah satu rujukan utama dalam memahami dan mengaplikasikan model regresi dengan R.

Colophon

Versi daring dari buku ini tersedia di <https://r4rm.netlify.app> dan sumber kode buku ini dapat diakses di <https://github.com/dedenistiawan/>. Buku ini dibuat menggunakan **Quarto**, yang memudahkan penulisan buku yang menggabungkan teks dan kode yang dapat dieksekusi.

Part I

Dasar-Dasar R

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

– Tyrion Lannister-Game of Thrones

R adalah bahasa pemrograman dan perangkat lunak yang dirancang khusus untuk keperluan analisis data, pengolahan statistik, dan visualisasi data. Bahasa ini menawarkan fleksibilitas untuk menulis kode guna menjalankan berbagai metode analisis statistik, mulai dari analisis statistik deskriptif hingga prediktif. Sebagai perangkat lunak, R menyediakan lingkungan interaktif untuk eksplorasi data, pengolahan, serta pembuatan laporan yang informatif. Kemampuan ini diperkuat dengan paket visualisasi yang sangat kaya seperti **ggplot2**, **lattice**, dan **plotly**, yang memungkinkan pengguna membuat grafik berkualitas tinggi. R tidak hanya membantu menganalisis data tetapi juga memvisualisasikannya dengan cara yang interaktif dan estetis, sehingga dapat menggali wawasan yang lebih dalam. R pertama kali dikembangkan pada awal 1990-an oleh **Ross Ihaka** dan **Robert Gentleman** di University of Auckland, Selandia Baru. Proyek ini terinspirasi oleh bahasa pemrograman S yang dikembangkan oleh John Chambers di Bell Laboratories. Pada tahun 1995, R dirilis sebagai perangkat lunak open source, sehingga memungkinkan pengembang di seluruh dunia untuk berkontribusi dalam pengembangannya.

Sejak itu, R berkembang pesat dengan ribuan paket yang tersedia di **CRAN** (Comprehensive R Archive Network) serta repositori lainnya. Dukungan komunitas dan kontribusi aktif dari pengembang di seluruh dunia telah menjadikan R sebagai bahasa yang terus berevolusi untuk memenuhi kebutuhan analisis data modern. R memiliki salah satu komunitas pengguna paling aktif di dunia. Komunitas ini terdiri dari akademisi, peneliti, praktisi industri, hingga pengembang perangkat lunak. Forum diskusi seperti **RStudio Community**, Stack Overflow, dan mailing list R membantu pengguna menyelesaikan masalah serta berbagi ide.

Ekosistem R terus berkembang dengan adanya **RStudio** sebagai Integrated Development Environment (IDE) yang populer. Selain itu, repositori paket seperti CRAN dan **Bioconductor**. Bioconductor adalah sebuah platform perangkat lunak untuk analisis data genomik yang kompleks, dikembangkan khusus untuk mendukung penelitian dalam bioinformatika dan biologi komputasi. Dibangun di atas bahasa pemrograman R, Bioconductor menyediakan berbagai paket yang dirancang untuk analisis data omics, seperti genomik, proteomik, dan metabolomik. Dukungan komunitas yang besar, dokumentasi yang melimpah, dan ketersediaan sumber daya membuat R menjadi alat yang mudah diakses oleh pemula sekaligus sangat berguna bagi pengguna tingkat lanjut.

R dapat diintegrasikan dengan berbagai bahasa pemrograman lain, seperti Python, C++, Java, dan SQL. Kompatibilitas ini memungkinkan pengguna untuk menggabungkan kekuatan masing-masing bahasa dalam satu proyek. Misalnya, pengguna dapat memanfaatkan Python untuk preprocessing data skala besar, kemudian menggunakan R untuk analisis statistik mendalam dan visualisasi. Selain itu, R juga mendukung integrasi dengan berbagai perangkat lunak lain seperti Excel, Tableau, dan Power BI, membuatnya menjadi bagian penting dalam alur kerja data modern.

1 Install R dan RStudio

R adalah bahasa pemrograman dan lingkungan komputasi yang sangat populer dalam analisis data dan statistik. Sebelum memulai pembelajaran mengenai model regresi menggunakan R, langkah pertama yang perlu dilakukan adalah menginstal R dan RStudio. R merupakan perangkat lunak sumber terbuka yang dapat diunduh secara gratis dari situs resmi CRAN (Comprehensive R Archive Network), sedangkan RStudio adalah antarmuka pengguna berbasis grafis yang mempermudah penggunaan R.

1.1 Install R

Untuk menginstal R, kunjungi situs resmi [R Project](#). Pilih sistem operasi yang sesuai dengan perangkat yang digunakan, seperti Windows, macOS, atau Linux.

Untuk menginstall R, kamu tidak memerlukan budget untuk membeli lisensi. R merupakan aplikasi open source sehingga dapat kamu unduh secara gratis di laman aslinya. Buka Halaman [R Project](#) untuk download Installer R seperti pada Figure 1.1

The Comprehensive R Archiv

Download and Install R

Precompiled binary distributions of the base system and contributed packages are available for download. If you are using a supported platform, you want one of these versions of R:

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package manager for the latest version. You can also download the source code and build R from source. See the [Source Code](#) link above.

Figure 1.1: Halaman R Project

2 Pengantar R

2.1 Install R dan R Studio

Untuk memulai menggunakan R, langkah pertama adalah menginstal perangkat lunaknya. Berikut langkah-langkah instalasinya:

1. Buka situs resmi **CRAN** (Comprehensive R Archive Network)
2. Pilih sistem operasi yang digunakan (Windows, macOS, atau Linux).
3. Unduh file instalasi dan ikuti instruksi instalasinya. Unduh dan Instal RStudio

Setelah R terinstal, unduh RStudio, sebuah Integrated Development Environment (IDE) untuk R, dari <https://www.rstudio.com>. Pilih versi gratis (RStudio Desktop) dan instal sesuai dengan sistem operasi Anda.

RStudio adalah IDE yang dirancang khusus untuk mempermudah penggunaan R. Dibandingkan dengan antarmuka dasar R, RStudio menawarkan pengalaman pengguna yang lebih nyaman melalui fitur-fitur berikut:

- **Script Editor:** Memungkinkan penulisan dan pengelolaan skrip dengan mudah.
- **Console:** Tempat menjalankan perintah R secara langsung.
- **Environment Pane:** Menampilkan variabel, data, dan objek yang sedang aktif.
- **Plots Pane:** Menampilkan hasil visualisasi seperti grafik dan plot.
- **Packages Pane:** Memudahkan instalasi dan pengelolaan pustaka.

CRAN (Comprehensive R Archive Network) adalah repositori resmi yang menyimpan ribuan paket tambahan untuk memperluas fungsi R. Paket-paket ini mencakup berbagai bidang aplikasi, seperti statistik, visualisasi, dan machine learning. Untuk mengunduh dan menginstal paket di R, langkah-langkahnya adalah sebagai berikut:

1. Buka RStudio dan pastikan Anda terhubung ke internet.
2. Gunakan perintah `install.packages()` untuk menginstal library atau paket.
3. Setelah instalasi selesai, muat paket ke dalam sesi kerja menggunakan perintah `library()`. Contoh:

```
library(ggplot2)
#> Warning: package 'ggplot2' was built under R version 4.4.2
```

Paket yang diunduh akan otomatis tersimpan di komputer Anda dan dapat digunakan kapan saja tanpa perlu mengunduh ulang. Dengan CRAN, pengguna dapat dengan mudah menyesuaikan R untuk berbagai kebutuhan analisis data.

2.2 Dasar-Dasar R

Objek dan Variabel Dalam R, **objek** adalah entitas yang menyimpan data atau hasil perhitungan. Objek bisa berupa angka, teks, tabel, atau bahkan fungsi. Setiap objek di R memiliki nama yang digunakan untuk merujuknya dalam kode, dan objek ini dapat disimpan ke dalam variabel untuk memudahkan manipulasi data.

Variabel adalah nama yang diberikan untuk menyimpan nilai atau objek. Variabel di R tidak memerlukan deklarasi tipe data sebelumnya, yang berarti kita dapat langsung menyimpan nilai apapun ke dalam variabel tersebut.

Variabel dapat dibuat dengan cara memberi nama pada objek dan menggunakan operator penugasan `<-` untuk menyimpan nilai ke dalam variabel.

```
x <- 5      # Menyimpan angka 5 dalam variabel x
nama <- "John" # Menyimpan string "John" dalam variabel nama
```

Di atas, `x` dan `nama` adalah variabel yang menyimpan objek berupa angka dan teks.

Mengakses Variabel Setelah variabel dibuat, kita dapat mengakses nilai yang tersimpan dalam variabel tersebut dengan cukup menyebutkan nama variabel:

```
print(x)    # Output: 5
#> [1] 5
print(nama) # Output: "John"
#> [1] "John"
```

2.3 Tipe Data dalam Variabel

Variabel di R dapat menyimpan berbagai tipe data, antara lain:

- **Numerik:** Menyimpan angka dengan atau tanpa desimal.
- **Karakter (String):** Menyimpan teks.

- **Logika:** Menyimpan nilai TRUE atau FALSE.
- **Faktor:** Digunakan untuk variabel kategorikal.
- **Daftar (List), Vektor, Matriks, dan Data Frame:** Struktur data kompleks yang menyimpan koleksi data.

Contoh variabel dengan berbagai tipe data:

```
angka <- 10          # Numerik
nama <- "Andi"      # Karakter
status <- TRUE        # Logika
kategori <- factor(c("A", "B", "A", "C")) # Faktor
```

2.3.1 Operasi dengan Variabel

Operasi dengan Variabel Variabel dapat digunakan dalam operasi matematika dan logika. Misalnya, jika kita memiliki dua variabel **a** dan **b**, kita bisa melakukan operasi penjumlahan:

```
a <- 3
b <- 4
hasil <- a + b # Menyimpan hasil penjumlahan a dan b ke dalam variabel hasil
print(hasil)    # Output: 7
#> [1] 7
```

2.3.2 Menimpa dan Mengubah Nilai Variabel

```
x <- 5      # x sekarang bernilai 5
x <- 10     # x sekarang bernilai 10 (nilai sebelumnya akan tertimpa)
print(x)    # Output: 10
#> [1] 10
```

2.4 Struktur Data pada R

R menyediakan berbagai jenis struktur data untuk mengelola dan memanipulasi data dalam berbagai bentuk. Pemahaman tentang struktur data di R sangat penting karena memungkinkan kita untuk menyimpan dan bekerja dengan data yang lebih kompleks. Beberapa struktur data dasar yang tersedia di R meliputi vektor, matriks, data frame, dan list.

2.4.1 Vektor

Vektor adalah struktur data paling dasar di R. Vektor adalah sekumpulan elemen yang memiliki tipe data yang sama, seperti angka, karakter, atau logika. Vektor dapat dibuat menggunakan fungsi `c()`, yang berarti combine atau concatenate. Contoh pembuatan vektor:

```
vektor_angka <- c(1, 2, 3, 4, 5)      # Vektor angka
vektor_karakter <- c("A", "B", "C")    # Vektor karakter
vektor_logika <- c(TRUE, FALSE, TRUE)  # Vektor logika
```

Mengakses elemen vektor:

```
vektor_angka[1]  # Mengakses elemen pertama (1)
#> [1] 1
vektor_karakter[2] # Mengakses elemen kedua ("B")
#> [1] "B"
```

Vektor bisa berupa vektor numerik, vektor karakter, atau vektor logika, tergantung tipe data elemen yang ada di dalamnya.

2.4.2 Matriks

Matriks adalah struktur data dua dimensi, yang berarti memiliki baris dan kolom. Semua elemen dalam matriks harus memiliki tipe data yang sama, seperti vektor. Contoh pembuatan matriks:

```
matriks <- matrix(1:6, nrow = 2, ncol = 3)  # Matriks 2x3 dengan elemen dari 1 hingga 6
print(matriks)
#>      [,1] [,2] [,3]
#> [1,]     1     3     5
#> [2,]     2     4     6
```

2.4.3 Data Frame

Data frame adalah struktur data yang lebih kompleks dan sering digunakan dalam analisis data. Data frame terdiri dari kolom-kolom yang dapat memiliki tipe data yang berbeda (numerik, karakter, logika, dll.), mirip dengan tabel dalam basis data atau spreadsheet. Contoh pembuatan data frame:

```

df <- data.frame(
  Nama = c("John", "Alice", "Bob"),
  Umur = c(25, 30, 22),
  Status = c("Single", "Married", "Single")
)
print(df)
#>   Nama Umur Status
#> 1 John  25 Single
#> 2 Alice 30 Married
#> 3 Bob   22 Single

```

2.4.4 List

List adalah struktur data yang paling fleksibel di R. List dapat menyimpan berbagai jenis objek yang berbeda dalam satu struktur data, termasuk vektor, matriks, data frame, atau bahkan fungsi. Contoh pembuatan list:

```

daftar <- list(
  Nama = "John",
  Umur = 25,
  Nilai = c(90, 85, 88)
)
print(daftar)
#> $Nama
#> [1] "John"
#>
#> $Umur
#> [1] 25
#>
#> $Nilai
#> [1] 90 85 88

```

2.4.5 Faktor

Faktor adalah tipe data di R yang digunakan untuk menyimpan data kategorikal. Faktor menyimpan data dalam bentuk level yang dapat digunakan untuk analisis kategorikal, seperti dalam model regresi atau analisis statistik lainnya. Contoh pembuatan faktor:

```

status <- factor(c("Single", "Married", "Single", "Divorced"))
print(status)
#> [1] Single Married Single Divorced
#> Levels: Divorced Married Single

```

Berikut adalah perbandingan antara berbagai struktur data di R:

Struktur Data	Deskripsi	Contoh Penggunaan
Vektor	Sekumpulan elemen dengan tipe data yang sama	Menyimpan data numerik atau karakter
Matriks	Struktur dua dimensi (baris dan kolom)	Menyimpan data dalam bentuk tabel numerik
Data Frame	Tabel dua dimensi dengan tipe data berbeda	Menyimpan data observasi dan variabel
List	Koleksi elemen dengan tipe data yang berbeda	Menyimpan objek dengan tipe data campuran
Faktor	Data kategorikal dengan level yang terbatas	Mengelompokkan data kategorikal

2.5 Fungsi dan Pemrograman di R

R menyediakan berbagai cara untuk mendefinisikan dan menggunakan fungsi dalam pemrograman. Fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu, menerima input, dan memberikan output. Pemrograman berbasis fungsi memungkinkan pengguna untuk menulis kode yang lebih modular dan terstruktur, yang memudahkan pemeliharaan dan pengembangan program.

2.5.1 Definisi Fungsi

Fungsi di R dibuat menggunakan kata kunci `function()`. Fungsi ini dapat menerima satu atau lebih argumen dan menghasilkan nilai output. Contoh pembuatan fungsi:

```

# Fungsi untuk menghitung kuadrat suatu angka
kuadrat <- function(x) {
  return(x^2)  # Mengembalikan nilai kuadrat dari x
}

```

2.5.2 Pemanggilan fungsi

```
hasil <- kuadrat(5) # Memanggil fungsi 'kuadrat' dengan argumen 5
print(hasil)          # Output: 25
#> [1] 25
```

2.5.3 Argumen fungsi

Fungsi di R dapat menerima berbagai jenis argumen, yang bisa berupa variabel, nilai tetap, atau objek lain. Argumen juga bisa diberikan nilai default, yang berarti fungsi tetap dapat dipanggil meskipun beberapa argumen tidak diberikan nilai. Contoh argumen dengan nilai default:

```
# Fungsi untuk menghitung hasil pembagian
bagi <- function(a, b = 2) {
  return(a / b) # Jika b tidak diberikan, akan menggunakan nilai default 2
}

print(bagi(10)) # Output: 5 (10 dibagi 2)
#> [1] 5
print(bagi(10, 5)) # Output: 2 (10 dibagi 5)
#> [1] 2
```

2.5.4 Fungsi dengan Beberapa Nilai Kembalian

Fungsi di R dapat mengembalikan lebih dari satu nilai. Salah satu cara untuk melakukan ini adalah dengan menggunakan list. Fungsi dapat mengembalikan objek yang lebih kompleks, seperti list, untuk menyimpan beberapa hasil sekaligus. Contoh fungsi dengan beberapa nilai kembalian:

```
hasil_operasi <- function(a, b) {
  hasil_penjumlahan <- a + b
  hasil_perkalian <- a * b
  return(list(penjumlahan = hasil_penjumlahan, perkalian = hasil_perkalian))
}

output <- hasil_operasi(4, 5)
print(output$penjumlahan) # Output: 9
#> [1] 9
```

```
print(output$perkalian)    # Output: 20
#> [1] 20
```

2.5.5 Fungsi Bawaan di R

R menyediakan banyak fungsi bawaan untuk melakukan berbagai tugas, seperti manipulasi data, analisis statistik, dan visualisasi. Fungsi-fungsi ini sangat berguna dan sering digunakan dalam berbagai analisis. Contoh fungsi bawaan:

```
# Fungsi untuk menghitung rata-rata
rata_rata <- mean(c(1, 2, 3, 4, 5))  # Output: 3
print(rata_rata)
#> [1] 3
```

2.5.6 Fungsi Anonim

Fungsi anonim adalah fungsi yang tidak memiliki nama. Fungsi jenis ini sering digunakan dalam operasi sementara atau dalam konteks tertentu, seperti dalam operasi dengan apply atau pemrograman berbasis vektor. Contoh fungsi anonim:

```
# Menggunakan fungsi anonim dalam apply
angka <- c(1, 2, 3, 4, 5)
hasil <- sapply(angka, function(x) x^2)  # Menerapkan fungsi untuk menghitung kuadrat
print(hasil)  # Output: 1 4 9 16 25
#> [1] 1 4 9 16 25
```