| **Nama:**  **Adrian Halim**  **NIM: 064002200043** | C:\Users\RPL-SI 02\Pictures\288px-Trisakti_Logo.svg.png | **MODUL 3**  **Nama Dosen:**  **Dedy Sugiarto** |
| --- | --- | --- |
| **Hari/Tanggal:**  **Senin, 17 JULI 2023** | **Praktikum Statistika** | **Nama Asisten Labratorium:**   1. **Elen Fadilla Estri**   **064002000008**   1. **Rukhy Zaifa Aduhalim**   **064002000041** |

**Pengelolaan Data pada Data Frame**

1. **Teori Singkat**

R (juga dikenal sebagai GNU S) adalah bahasa pemrograman dan perangkat lunak untuk analisis statistika dan grafik. R dibuat oleh Ross Ihaka dan Robert Gentleman di Universitas Auckland, Selandia Baru, dan kini dikembangkan oleh R Development Core Team, di mana Chambers merupakan anggotanya. R dinamakan sebagian setelah nama dua pembuatnya (Robert Gentleman dan Ross Ihaka), dan sebagian sebagian dari permainan nama dari S.

Bahasa R kini menjadi standar de facto di antara statistikawan untuk pengembangan perangkat lunak statistika, serta digunakan secara luas untuk pengembangan perangkat lunak statistika dan analisis data. R merupakan bagian dari proyek GNU. Kode sumbernya tersedia secara bebas di bawah Lisensi Publik Umum GNU, dan versi biner prekompilasinya tersedia untuk berbagai sistem operasi. R menggunakan antarmuka baris perintah, meski beberapa antarmuka pengguna grafik juga tersedia.

R menyediakan berbagai teknik statistika (permodelan linier dan nonlinier, uji statistik klasik, analisis deret waktu, klasifikasi, klasterisasi, dan sebagainya) serta grafik. R, sebagaimana S, dirancang sebagai bahasa komputer sebenarnya, dan mengizinkan penggunanya untuk menambah fungsi tambahan dengan mendefinisikan fungsi baru. Kekuatan besar dari R yang lain adalah fasilitas grafiknya, yang menghasilkan grafik dengan kualitas publikasi yang dapat memuat simbol matematika. R memiliki format dokumentasi seperti LaTeX, yang digunakan untuk menyediakan dokumentasi yang lengkap, baik secara daring (dalam berbagai format) maupun secara cetakan.

RStudio merupakan integrated development environment (IDE) khusus bagi bahasa pemrograman R. Software ini menyediakan R console, code editor dengan syntax highlighting, code completion dan direct execution, environment, history, connections, dan fitur-fitur tambahan lainnya seperti file manager, packages manager, help, plot viewer, hingga project versioning menggunakan git. RStudio sebenarnya memiliki dua versi, yaitu open source (gratis) dan commercial edition (berbayar). RStudio juga tidak hanya terbatas dalam bentuk aplikasi dekstop, melainkan terdapat versi RStudio Server, yaitu RStudio yang dapat diakses melalui browser yang terhubung dengan suatu jaringan komputer. Untuk saat ini, versi RStudio yang akan dijelaskan hanyalah RStudio open source berbasis dekstop saja.

1. **Alat dan Bahan**

Hardware : Laptop/PC

Software : R Studio

1. **Elemen Kompetensi**
   1. Latihan pertama – Merge Data

1. Jalankan souce code berikut. Ganti nama variable (seperti data\_nama) menjadi variable dengan nama kalian masing - masing. Data yang digunakan adalah data houseprice.csv. Melakukan Read CSV dengan cara menginput data houseprice, sesuaikan dengan lokasi direktori dimana kalian menyimpan file csvnya.

| data\_nama = read.delim(“clipboard”)  View(data\_nama) |
| --- |

Output:

|  |
| --- |

2. Lakukan subset data dengan cara sub set data frame khusus yang memiliki nilai variabel Bedrooms =2

| nama <- data\_nama[data\_nama$Bedrooms==2,]  View(nama) |
| --- |

Output:

|  |
| --- |

3. Selanjutnya ialah mengubah nama variabel. Berikut adalah tahapan untuk mengubah nilai dalam kolom Bathrooms dalam kondisi jika Bathrooms >2 , maka akan diganti dengan nilai large, jika tidak maka small.

| nama$Bathrooms <- ifelse(nama$Bathrooms > 2, c("large"), c("small"))  View(nama) |
| --- |

Output:

|  |
| --- |

4. Lalu langkah selanjutnya adalah menambah variabel. Tahap 1 🡪 Membuat variable baru dari dataku1 dengan nilai sesuai kondisi ifelse yang ditentukan. Sesuaikan nama variable baru dengan nama praktikan.

| nama$newvariable <- ifelse(nama$Offers > 2, c("large"), c("small"))  View(nama) |
| --- |

Output

|  |
| --- |

Tahap 2 🡪 Mengubah isi baris data dari kolom baru yang telah dibuat

| nama$newvariable <- nama$Price/nama$SqFt  View(nama) |
| --- |

Output:

|  |
| --- |

5. Delete Variabel. Selain bisa menambah, kita juga bisa menghapus variable. Dalam percobaan ini kita akan menghapus variable yang baru saja kita buat.

| nama$newvariable<-NULL |
| --- |

Output:

|  |
| --- |

6. Merge Data Frame. Tahap 1 🡪 Merge kolom Artinya memisahkan data sesuai kolom yang diinginkan lalu kemudian digabungkan.

| data\_nama.kolom1dan2 <- data\_nama[,1:2]  View(data\_nama.kolom1dan2) |
| --- |

Output:

|  |
| --- |

Tahap 2 🡪 Merge kolom Artinya memisahkan data sesuai kolom yang diinginkan lalu kemudian digabungkan.

| data\_elen.kolom3dan4 <- data\_elen[,3:4]  View(data\_elen.kolom3dan4 |
| --- |

Output:

|  |
| --- |

Tahap 3 🡪 Merge kolom Artinya memisahkan data sesuai kolom yang diinginkan lalu kemudian digabungkan.

| data\_nama.kolom1sd4<-cbind(data\_nama.kolom1dan2, data\_nama.kolom3dan4)  View(data\_nama.kolom1sd4) |
| --- |

Output:

|  |
| --- |

7. Merge Baris artinya memisahkan data sesuai baris yang diinginkan dengan menggunakan range baris. Lalu kemudian digabungkan.

| data\_nama.baris1sd3 <- data\_nama[1:3,]  data\_nama.baris4sd6 <- data\_nama[4:6,]  data\_nama.baris1sd6 <- rbind(data\_nama.baris1sd3, data\_nama.baris4sd6)  View(data\_nama.baris1sd6) |
| --- |

Output:

|  |
| --- |

8. Sort data frame. Apa yang terjadi dengan data setelah di sort?

Jawaban: ?

| data\_nama.sort<-data\_nama[order(data\_nama$Price),]  View(data\_nama.sort) |
| --- |

Output:

|  |
| --- |

* 1. Latihan Kedua – Tugas

Gunakan dataset pada tugas 1 yang telah ditambah lagi datanya dengan 10 mhs TIF/SI. Sehingga total baris data pada file tersebut berjumlah 30. Ulangi kembali perintah-perintah di atas dan sesuaikan dengan data anda. Dan lampirkan Screen Capture untuk tiap poin yang ada.

1. Read CSV

| data\_Adrian = read.delim("clipboard")  View(data\_Adrian) |
| --- |

Output:

|  |
| --- |

2. Subset Data 🡪 Gunakan Kolom Tinggi Badan untuk bagian ini

| #subset data kolom tinggi  data\_Adrian <- data\_Adrian[data\_Adrian$Tinggi.Badan==157,]  View(data\_Adrian) |
| --- |

Output:

|  |
| --- |

3. Mengubah nilai suatu variable 🡪 Ubah isi kolom tinggi badan untuk tinggi > 160 menjadi “Tinggi” dan jika bukan berarti “Pendek”

| #ubah nilai suatu variable untuk kolom tinggi>160  data\_Adrian$Tinggi.Badan <- ifelse(data\_Adrian$Tinggi.Badan > 160, c("tinggi"), c("pendek"))  View(data\_Adrian) |
| --- |

Output:

|  |
| --- |

4. Menambah Variabel

* Buat variabel kolom baru dengan nama “Jurusan” dengan isi baris datanya adalah “Infor22”
* Buat varibel kolom baru kedua dengan nama “Fakultas” dengan isi baris datanya adalah “FTI”

| #menambah variabel  data\_Adrian$Jurusan <- ("INFOR22")  View(data\_Adrian)  data\_Adrian$Fakultas <- ("FTI")  View(data\_Adrian) |
| --- |

Output:

|  |
| --- |

5. Delete Variabel 🡪 Hapus kolom Fakultas

| data\_Adrian$Fakultas <- NULL  View(data\_Adrian) |
| --- |

Output:

|  |
| --- |

6. Merge Data Frame

* Gabung kolom Nama dan Gender
* Gabung kolom Angkatan dan Tinggi Badan
* Gabungkan 2 variabel kolom gabungan diatas

| #merge data frame  data\_Adrian.kolom1dan2 <- data\_Adrian[,1:2]  View(data\_Adrian.kolom1dan2)  data\_Adrian.kolom3dan4 <- data\_Adrian[,3:4]  View(data\_Adrian.kolom3dan4)  data\_Adrian.kolom1sd4 <- cbind(data\_Adrian.kolom1dan2, data\_Adrian.kolom3dan4)  View(data\_Adrian.kolom1sd4) |
| --- |

Output:

|  |
| --- |

7. Merge Baris

* Buat gabungan baris 1-5
* Buat gabungan baris 25-30
* Gabungkan 2 variabel diatas

| data\_Adrian.baris1sd5 <- data\_Adrian.kolom1sd4[1:5,]  View(data\_Adrian.baris1sd5)  data\_Adrian.baris25sd30 <- data\_Adrian.kolom1sd4[25:30,]  View(data\_Adrian.baris25sd30)  data\_Adrian.baris1sd30<-rbind(data\_Adrian.baris1sd5, data\_Adrian.baris25sd30)  View(data\_Adrian.baris1sd30) |
| --- |

Output:

|  |
| --- |

8. Sort Data Frame 🡪 Lakukan sort berdasarkan waktu perjalanan

| data\_Adrian.sort<-  data\_Adrian[order(data\_Adrian$Waktu.Perjalanan),]  View(data\_Adrian) |
| --- |

Output:

|  |
| --- |

1. **File Praktikum**

Github Repository:

|  |
| --- |

1. **Soal Latihan**

Soal:

1. Apa saja kegunaan pengelolaan data pada data frame?
2. Dalam kasus apakah data perlu dihapus?

Jawaban:  
1. Pengelolaan data pada data frame sangat penting dalam analisis data dan pengolahan data menggunakan bahasa pemrograman seperti Python dan R. Data frame adalah struktur data berbentuk tabular yang terdiri dari baris dan kolom, mirip dengan tabel dalam database atau spreadsheet. Beberapa kegunaan utama dari pengelolaan data pada data frame adalah sebagai berikut:

Penyimpanan Data: Data frame digunakan untuk menyimpan data secara terstruktur, sehingga memudahkan dalam membaca, memanipulasi, dan menganalisis data.

Manipulasi Data: Pengelolaan data frame memungkinkan kita untuk melakukan berbagai operasi manipulasi data seperti pemfilteran, penggabungan, pengurutan, pengelompokan, dan transformasi data.

Analisis Data: Data frame memungkinkan analisis statistik dan eksplorasi data dengan mudah, seperti menghitung statistik deskriptif, membangun model prediksi, dan melakukan visualisasi data.

Pembersihan Data: Data frame memungkinkan identifikasi dan penanganan nilai-nilai yang hilang atau tidak valid dalam data. Proses ini dikenal sebagai pembersihan data (data cleaning).

Preprocessing Data: Sebelum menerapkan model atau algoritma pada data, data frame memungkinkan kita untuk melakukan preprocessing data seperti penskalaan, pengkodean kategori, dan normalisasi, agar data siap digunakan dalam proses pembelajaran mesin.

Ekstraksi Fitur (Feature Extraction): Dalam analisis data dan pembelajaran mesin, kita sering kali perlu mengekstraksi fitur dari data yang kompleks. Data frame memungkinkan ekstraksi dan pemilihan fitur dengan mudah.

Pemantauan Data (Data Monitoring): Dalam beberapa aplikasi, data frame digunakan untuk memantau data secara real-time atau berkala, seperti pemantauan kinerja bisnis atau aplikasi.

2. Dalam kasus data frame, ada beberapa situasi di mana data perlu dihapus:

Data Duplikat: Ketika data frame mengandung duplikat dari baris atau entitas yang sama, data duplikat harus dihapus agar analisis tidak terpengaruh oleh duplikasi ini.

Missing Values: Jika ada nilai yang hilang (missing values) dalam data frame, terdapat beberapa pilihan untuk menanganinya. Salah satu pilihan adalah menghapus baris atau kolom yang mengandung nilai yang hilang jika data yang hilang tersebut cukup banyak dan tidak mewakili informasi penting. Namun, jika nilai yang hilang tersebut penting, maka penanganan yang lebih lanjut diperlukan, seperti mengisi nilai yang hilang dengan nilai rata-rata atau interpolasi.

Data yang Tidak Relevan: Terkadang, data frame bisa berisi kolom atau atribut yang tidak relevan atau tidak dibutuhkan dalam analisis. Data yang tidak relevan tersebut dapat dihapus untuk menyederhanakan analisis dan mengurangi kompleksitas model.

Penghapusan Sebagian Data: Dalam beberapa situasi, mungkin ada kebutuhan khusus untuk menghapus sebagian data tertentu berdasarkan kriteria tertentu, seperti menghapus data yang dikategorikan sebagai outlier atau data yang bersifat rahasia atau sudah kedaluwarsa.

Penting untuk melakukan analisis dan pertimbangan yang hati-hati sebelum menghapus data, karena penghapusan data yang tidak tepat dapat menyebabkan kehilangan informasi penting dan mengganggu integritas analisis yang dilakukan.

1. **Kesimpulan**

1.Dalam penegerjaan praktikum hari ini kita dapat memahami

Dalam pengerjaan praktikum Statistika, Dalam pengerjaan data pada data frame, terutama dalam konteks pemrograman menggunakan bahasa Python atau R, ada beberapa tahapan atau langkah penting yang biasanya dilakukan. Berikut adalah penjelasan mengenai langkah-langkah tersebut:

Impor Library: Pertama-tama, Anda perlu mengimpor library yang diperlukan untuk bekerja dengan data frame. Dalam Python, Anda biasanya akan menggunakan library seperti Pandas, NumPy, dan matplotlib/seaborn untuk analisis dan visualisasi data. Di R, Anda akan menggunakan library seperti dplyr, tidyr, dan ggplot2.

Membaca Data: Data dapat diimpor dari berbagai sumber, seperti file CSV, Excel, atau database. Anda akan menggunakan fungsi yang disediakan oleh library yang telah diimpor untuk membaca data dari sumber tersebut dan menyimpannya dalam bentuk data frame.

Eksplorasi Data: Setelah data berhasil diimpor menjadi data frame, langkah selanjutnya adalah menjalankan eksplorasi data. Anda dapat menggunakan berbagai metode dan fungsi yang disediakan oleh library yang telah diimpor untuk mendapatkan informasi tentang struktur data, melihat beberapa baris pertama dan terakhir dari data, serta mungkin mendapatkan statistik deskriptif sederhana.

Pembersihan Data: Pada tahap ini, Anda akan menangani nilai-nilai yang hilang (missing values) dan duplikat dalam data frame. Anda dapat menggunakan fungsi-fungsi seperti dropna() atau fillna() di Pandas (Python) atau fungsi filter() di dplyr (R) untuk menangani nilai-nilai yang hilang. Untuk mendeteksi dan menghapus duplikat, Anda dapat menggunakan metode drop\_duplicates() di Pandas (Python) atau fungsi distinct() di dplyr (R).

Pengolahan Data: Jika diperlukan, Anda dapat melakukan pengolahan data pada data frame untuk menyiapkan data sesuai dengan kebutuhan analisis selanjutnya. Pengolahan data dapat melibatkan transformasi kolom, penggabungan data frame, atau pemfilteran berdasarkan kriteria tertentu.

Analisis Data: Setelah data frame telah dibersihkan dan diproses, langkah selanjutnya adalah melakukan analisis data sesuai dengan tujuan Anda. Ini bisa termasuk perhitungan statistik, pembuatan grafik visual, pengelompokan data, atau bahkan pembuatan model prediksi menggunakan metode pembelajaran mesin.

Visualisasi Data: Visualisasi data sangat penting untuk memahami pola dan tren dalam data. Anda dapat menggunakan library matplotlib/seaborn di Python atau ggplot2 di R untuk membuat visualisasi data yang informatif dan menarik.

Evaluasi dan Interpretasi: Setelah analisis selesai, tahap terakhir adalah mengevaluasi hasil analisis dan menginterpretasikan temuan-temuan yang diperoleh dari data. Ini akan membantu Anda dalam mengambil keputusan atau menyusun rekomendasi berdasarkan informasi yang telah ditemukan dari data.

Penyimpanan Data (Opsional): Jika diperlukan, Anda dapat menyimpan data yang telah diproses dan diolah ke dalam format yang sesuai untuk penggunaan atau berbagi lebih lanjut.

Langkah-langkah ini tidak selalu harus diikuti dalam urutan yang ketat, dan terkadang beberapa tahap dapat dilakukan secara bersamaan. Pengerjaan data pada data frame memerlukan pemahaman yang baik tentang data yang sedang dikerjakan, serta kemampuan dalam menggunakan fungsi-fungsi dan metode yang sesuai dari library yang digunakan. Dengan demikian, Anda dapat melakukan analisis yang lebih mendalam dan mendapatkan wawasan berharga dari data yang Anda miliki.

Kita juga dapat mengetahui Kesimpulan di atas adalah bahwa pengelolaan data pada data frame sangat penting dalam analisis data dan pengolahan data menggunakan bahasa pemrograman seperti Python dan R. Data frame adalah struktur data berbentuk tabular yang terdiri dari baris dan kolom, mirip dengan tabel dalam database atau spreadsheet. Beberapa kegunaan utama pengelolaan data pada data frame termasuk penyimpanan data, manipulasi data, analisis data, pembersihan data, preprocessing data, ekstraksi fitur, dan pemantauan data. Dalam pengelolaan data frame, ada beberapa situasi di mana data perlu dihapus. Pertama, data duplikat harus dihapus untuk menghindari pengaruh duplikasi pada analisis. Kedua, ketika ada nilai yang hilang (missing values) dalam data frame, ada beberapa opsi untuk menanganinya, termasuk menghapus baris atau kolom yang mengandung nilai yang hilang jika data yang hilang tersebut cukup banyak dan tidak mewakili informasi penting. Namun, jika nilai yang hilang tersebut penting, maka penanganan yang lebih lanjut diperlukan. Ketiga, data yang tidak relevan atau tidak dibutuhkan dalam analisis dapat dihapus untuk menyederhanakan analisis dan mengurangi kompleksitas model. Terakhir, ada kasus khusus di mana sebagian data perlu dihapus berdasarkan kriteria tertentu. Penting untuk berhati-hati dalam melakukan penghapusan data, karena penghapusan data yang tidak tepat dapat menyebabkan kehilangan informasi penting dan mengganggu integritas analisis yang dilakukan. Pengelolaan data yang baik pada data frame akan memastikan data yang digunakan dalam analisis bersih, relevan, dan dapat diandalkan untuk mendukung pengambilan keputusan yang akurat dan tepat.

1. **Cek List (✔)**

| **No** | **Elemen Kompetensi** | **Penyelesaian** | |
| --- | --- | --- | --- |
| **Selesai** | **Tidak Selesai** |
| **1.** | Latihan Pertama | **✅** |  |
| **2.** | Latihan Kedua | **✅** |  |

1. **Formulir Umpan Balik**

| **No** | **Elemen Kompetensi** | **Waktu Pengerjaan** | **Kriteria** |
| --- | --- | --- | --- |
| **1.** | Latihan Pertama | 60 Menit | Menarik |
| **2.** | Latihan Kedua | 60 Menit | Menarik |

Keterangan:

1. Menarik
2. Baik
3. Cukup
4. Kurang