|  |  |  |
| --- | --- | --- |
| **Nama:** | **Praktikum Statistika** | **MODUL 3** |
| **NIM:** | **Nama Dosen:** |
|  | **Dedy Sugiarto** |
| **Hari/Tanggal:** | **Nama Asisten Labratorium:**   1. **Kharisma Maulida Saara**   **064002200024**   1. **Tarum Widyasti Pertiwi 064002200027** |
| **Rabu, 20 Maret 2023** |

Pengelolaan Data pada Data Frame

# Teori Singkat

Pengelolaan data pada DataFrame merupakan konsep penting dalam analisis data menggunakan Python, terutama dengan library seperti pandas. Berikut adalah beberapa teori terkait pengelolaan data pada DataFrame:

* DataFrame: DataFrame adalah struktur data dua dimensi yang digunakan untuk menyimpan dan mengelola data dalam bentuk tabel yang terorganisir. Setiap kolom dalam DataFrame mewakili sebuah variabel, sedangkan setiap baris mewakili sebuah catatan atau observasi.
* Kolom dan Baris: Kolom dalam DataFrame mewakili variabel, sementara baris mewakili catatan atau observasi. Kolom dan baris dapat diakses menggunakan label atau indeks numerik.
* Pengindeksan: Pengindeksan memungkinkan akses ke bagian tertentu dari DataFrame. Anda dapat mengakses kolom menggunakan nama kolom, sedangkan baris dapat diakses menggunakan indeks numerik atau label baris.
* Manipulasi Data: Anda dapat melakukan berbagai operasi manipulasi data pada DataFrame, seperti menambah atau menghapus kolom dan baris, mengubah nilai, menggabungkan DataFrame, mengurutkan data, serta melakukan operasi agregasi seperti penghitungan rata-rata atau jumlah.
* Pembersihan Data: Pembersihan data adalah proses untuk menghapus atau memperbaiki data yang tidak valid, hilang, atau tidak lengkap. Ini melibatkan deteksi dan penanganan nilai yang hilang, duplikat, atau outlier.
* Pengindeksan dan Pemotongan: Pengindeksan dan pemotongan memungkinkan Anda untuk memilih subset dari DataFrame berdasarkan kriteria tertentu. Anda dapat menggunakan metode seperti .loc[], .iloc[], dan metode pemotongan untuk melakukan ini.
* Pengurutan Data: Pengurutan data memungkinkan Anda untuk mengurutkan DataFrame berdasarkan nilai pada satu atau beberapa kolom. Hal ini mempermudah analisis dan pemrosesan data yang memerlukan data yang diurutkan.
* Operasi Statistik: Anda dapat melakukan berbagai operasi statistik pada DataFrame, seperti penghitungan rata-rata, median, standar deviasi, korelasi, dan lainnya. Library seperti pandas menyediakan fungsi bawaan untuk melakukan operasi ini dengan mudah.
* Menggabungkan DataFrame: Anda dapat menggabungkan dua atau lebih DataFrame berdasarkan kunci tertentu menggunakan metode seperti .merge() atau .concat(). Ini berguna saat Anda perlu menggabungkan data dari berbagai sumber.
* Ekspor dan Impor Data: Anda dapat mengimpor data dari berbagai sumber seperti file CSV, Excel, SQL database, atau web API ke dalam DataFrame. Anda juga dapat mengekspor DataFrame ke berbagai format file untuk berbagi atau menyimpan data.

Pengelolaan data pada DataFrame sangat penting dalam analisis data menggunakan Python, karena memungkinkan Anda untuk menyusun, membersihkan, memanipulasi, dan menganalisis data dengan efisien. Dengan menggunakan library seperti pandas, Anda dapat dengan mudah mengelola data dalam format tabular dan menjalankan berbagai analisis data dengan Python.

1. **Alat dan Bahan**

Hardware : Laptop/PC Software : Jupyter Notebook

# 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.

import pandas as pd

# Membaca data dari clipboard

data\_nama = pd.read\_clipboard()

# Menampilkan data

print(data\_nama)

Output:

|  |
| --- |
|  |

* + 1. Lakukan subset data dengan cara sub set data frame khusus yang memiliki nilai variabel Bedrooms =2

# Filter data\_nama untuk hanya mengambil baris dengan Bedrooms == 2

nama = data\_nama[data\_nama['Bedrooms'] == 2]

# Menampilkan nama

print(nama)

Output:

|  |
| --- |
|  |

* + 1. 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.

# Konversi kolom 'Bathrooms' ke tipe data numerik

nama['Bathrooms'] = pd.to\_numeric(nama['Bathrooms'])

import numpy as np

# Mengubah nilai dalam kolom 'Bathrooms' berdasarkan kondisi yang diberikan

nama['Bathrooms'] = nama['Bathrooms'].apply(lambda x: 'large' if x > 2 else 'small')

# Menampilkan DataFrame setelah modifikasi

print(nama)

Output:

|  |
| --- |
|  |

* + 1. 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.

import numpy as np

# Menambahkan kolom baru 'newvariable' berdasarkan kondisi yang diberikan

nama['newvariable'] = np.where(nama['Offers'] > 2, 'large', 'small')

# Menampilkan DataFrame 'nama' setelah penambahan kolom baru

print(nama)

Output

|  |
| --- |
|  |

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

# Menambahkan kolom baru 'newvariable' dengan hasil pembagian Price oleh SqFt

nama['newvariable'] = nama['Price'] / nama['SqFt']

# Menampilkan DataFrame 'nama' setelah penambahan kolom baru

print(nama)

Output:

|  |
| --- |
|  |

* + 1. Delete Variabel. Selain bisa menambah, kita juga bisa menghapus variable. Dalam percobaan ini kita akan menghapus variable yang baru saja kita buat.

# Menghapus kolom baru 'newvariable' dari DataFrame

nama = nama.drop(columns=['newvariable'])

# Menampilkan DataFrame 'nama' setelah menghapus kolom baru

print(nama)

Output:

|  |
| --- |
|  |

* + 1. Merge Data Frame. Tahap 1 ◻ Merge kolom Artinya memisahkan data sesuai kolom yang diinginkan lalu kemudian digabungkan.

# Memilih kolom 1 dan 2 dari DataFrame data\_nama

kolom1dan2 = data\_nama.iloc[:, 0:2]

# Menampilkan DataFrame kolom1dan2

print(kolom1dan2)

Output:

|  |
| --- |
|  |

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

# Memilih kolom 3 dan 4 dari DataFrame data\_nama

kolom3dan4 = data\_nama.iloc[:, 2:4]

# Menampilkan DataFrame kolom3dan4

print(kolom3dan4)

# Menampilkan DataFrame kolom1dan2

print(kolom3dan4)

Output:

|  |
| --- |
|  |

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

# Menggabungkan dua DataFrame secara horizontal

kolom1sd4 = pd.concat([kolom1dan2, kolom3dan4], axis=1)

# Menampilkan DataFrame kolom1sd4

print(kolom1sd4)

Output:

|  |
| --- |
|  |

* + 1. Merge Baris artinya memisahkan data sesuai baris yang diinginkan dengan menggunakan range baris. Lalu kemudian digabungkan.

# Menggabungkan baris dari dua DataFrame

baris1sd3 = data\_nama.iloc[0:3, :]

baris4sd6 = data\_nama.iloc[3:6, :]

baris1sd6 = baris1sd3.append(baris4sd6)

# Menampilkan DataFrame baris1sd6

print(baris1sd6)

Output:

|  |
| --- |
|  |

* + 1. Sort data frame. Apa yang terjadi dengan data setelah di sort? Jawaban: ?

# Mengurutkan DataFrame berdasarkan kolom 'Price'

data\_nama\_sort = data\_nama.sort\_values(by='Price')

# Menampilkan DataFrame yang sudah diurutkan

print(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

import pandas as pd

# Membaca data dari clipboard

data\_tarum1 = pd.read\_clipboard()

# Menampilkan data

print(data\_tarum1)

Output:

|  |
| --- |
|  |

* + 1. Subset Data ◻ Gunakan Kolom Tinggi Badan untuk bagian ini

tarum1 = data\_tarum1[data\_tarum1['Tinggi Badan'] == 160]

print(tarum1)

Output :

|  |
| --- |
|  |

* + 1. Mengubah nilai suatu variable ◻ Ubah isi kolom tinggi badan untuk tinggi > 160 menjadi “Tinggi” dan jika bukan berarti “Pendek”

tarum1['Tinggi Badan'] = pd.to\_numeric(tarum1['Tinggi Badan'])

import numpy as np

tarum1['Tinggi Badan'] = tarum1['Tinggi Badan'].apply(lambda x: 'Tinggi Badan' if x > 160 else 'Pendek')

print(tarum1)

Output:

|  |
| --- |
|  |

* + 1. Menambah Variabel
       - Buat variabel kolom baru dengan nama “Jurusan” dengan isi baris datanya adalah “Infor20”
       - Buat varibel kolom baru kedua dengan nama “Fakultas” dengan isi baris datanya adalah “FTI”

# Menambah kolom "Jurusan" dengan isi baris datanya adalah "Infor20"

tarum1['Jurusan'] = 'Infor20'

# Menambah kolom "Fakultas" dengan isi baris datanya adalah "FTI"

tarum1['Fakultas'] = 'FTI'

# Menampilkan DataFrame setelah penambahan kolom baru

print(tarum1)

Output:

|  |
| --- |
|  |

* + 1. Delete Variabel ◻ Hapus kolom Fakultas

# Menghapus kolom baru 'newvariable' dari DataFrame

tarum1 = tarum1.drop(columns=['Jurusan', 'Fakultas'])

# Menampilkan DataFrame 'nama' setelah menghapus kolom baru

print(tarum1)

Output:

|  |
| --- |
|  |

* + 1. Merge Data Frame
       - Gabung kolom Nama dan Gender
       - Gabung kolom Angkatan dan Tinggi Badan
       - Gabungkan 2 variabel kolom gabungan diatas

# Memilih kolom 1 dan 2 dari DataFrame data\_nama

kolom1dan2 = data\_tarum1.iloc[:, 0:2]

# Menampilkan DataFrame kolom1dan2

print(kolom1dan2)

Output :

|  |
| --- |
|  |

* + 1. Merge Baris
       - Buat gabungan baris 1-5
       - Buat gabungan baris 25-30
       - Gabungkan 2 variabel diatas

# Menggabungkan baris dari dua DataFrame

baris1sd5 = data\_tarum1.iloc[0:5, :]

baris25sd30 = data\_tarum1.iloc[24:30, :]

barisgabungan = baris1sd5.append(baris25sd30)

# Menampilkan DataFrame

print(barisgabungan)

Output:

|  |
| --- |
|  |

* + 1. Sort Data Frame ◻ Lakukan sort berdasarkan waktu perjalanan

# Mengurutkan DataFrame

data\_tarum1\_sort = data\_tarum1.sort\_values(by='Waktu Perjalanan')

# Menampilkan DataFrame yang sudah diurutkan

print(data\_tarum1\_sort)

Output:

|  |
| --- |
|  |

# File Praktikum

Github Repository:

# Soal Latihan

Soal:

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

Jawaban:

1. untuk melakukan penyaringan dan pemilihan data spesifik, memudahkan identifikasi dan analisis tren atau pola tertentu dalam kumpulan data. Kedua, pengelolaan data memfasilitasi proses pembersihan data, seperti menghilangkan nilai yang hilang atau duplikat, yang sangat penting untuk memastikan keakuratan analisis. Ketiga, memungkinkan transformasi data, seperti penggabungan kolom atau baris, agregasi data, atau konversi tipe data, yang mendukung persiapan data untuk analisis lanjutan atau pembuatan model prediktif

2.Menghapus atau memperbaiki data yang tidak valid, hilang, atau tidak lengkap. Ini melibatkan deteksi dan penanganan nilai yang hilang, duplikat, atau outlier.

# Kesimpulan

* 1. Dalam pengerjaan praktikum Statistika, …
  2. Kita juga dapat mengetahui…

# Cek List ()

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Elemen Kompetensi** | **Penyelesaian** | |
| **Selesai** | **Tidak Selesai** |
| **1.** | Latihan Pertama | **Checkmark with solid fill** |  |
| **2.** | Latihan Kedua | **Checkmark with solid fill** |  |

1. **Formulir Umpan Balik**

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Elemen Kompetensi** | **Waktu Pengerjaan** | **Kriteria** |
| **1.** | Latihan Pertama | 30 Menit | Menarik |
| **2.** | Latihan Kedua | 30 Menit | Menarik |

Keterangan:

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