Dataframe II

12 Mar 2024

COMBINING DATAFRAME

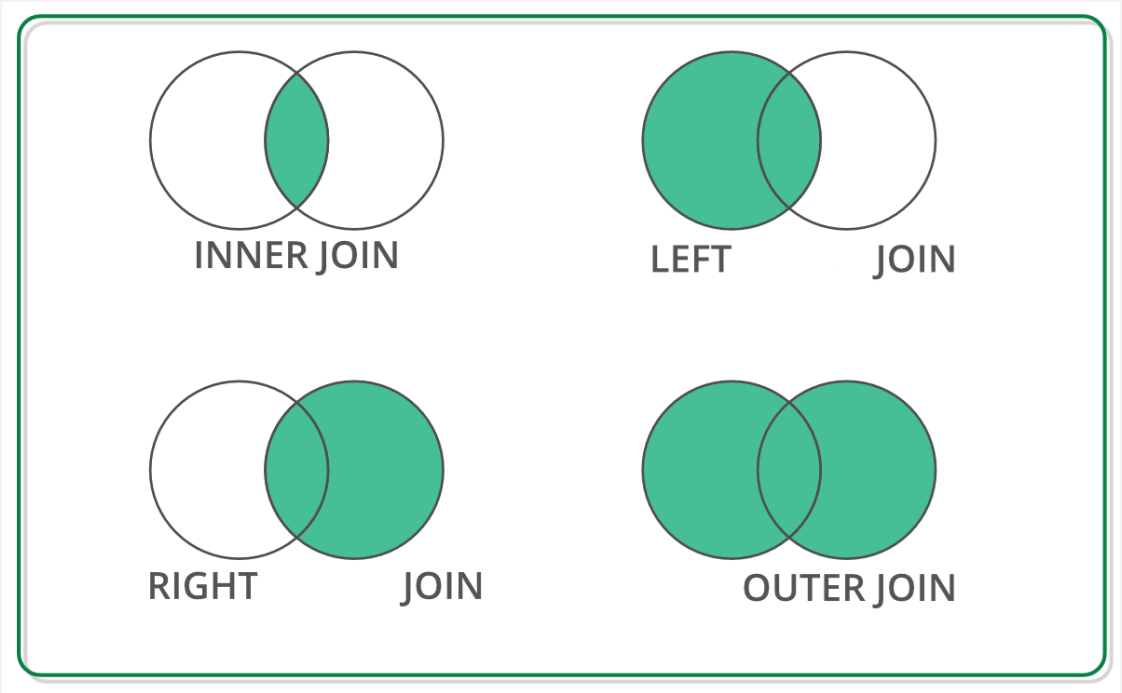
COMBINING DATAFRAME

- Untuk menggabungkan dua DataFrame, Anda dapat menggunakan berbagai metode tergantung pada kasus penggunaan Anda.
- Metode yang umum digunakan adalah:
  - merge()
  - join()
  - concat()

COMBINING DATAFRAME : Merge

- Fungsi merge() di pandas adalah salah satu fitur inti untuk melakukan operasi penggabungan data yang mirip dengan operasi join dalam basis data SQL.
- Ini memungkinkan pengguna untuk menggabungkan dua DataFrame atau Series berdasarkan satu atau lebih kunci bersama dengan cara yang sangat fleksibel, termasuk mendukung inner joins, left joins, right joins, dan full outer joins.

COMBINING DATAFRAME : Merge



COMBINING DATAFRAME : Merge

- Syntax:

```
pd.merge(left, right, how='inner', on=None, left_on=None,
right_on=None, left_index=False, right_index=False, sort=False,
suffixes=('_x', '_y'))
```

  - **left**: DataFrame yang berada di sisi kiri operasi merge.
  - **right**: DataFrame yang berada di sisi kanan operasi merge.
  - **how**: Tipe join yang digunakan; inner, outer, left, atau right.
  - **on**: Nama kolom/kolom untuk join. Harus ada di kedua DataFrame.
  - **left_on**: Kolom dari DataFrame kiri untuk digunakan sebagai kunci.
  - **right_on**: Kolom dari DataFrame kanan untuk digunakan sebagai kunci.
  - **left_index**: Jika True, gunakan index dari DataFrame kiri sebagai kunci join.
  - **right_index**: Jika True, gunakan index dari DataFrame kanan sebagai kunci join.
  - **sort**: Urutkan data berdasarkan kunci join setelah penggabungan jika True.
  - **suffixes :** tambahan akhiran pada kolom yang overlap

COMBINING DATAFRAME : JOIN

- Pandas join() adalah metode yang memungkinkan Anda dengan mudah menggabungkan kolom dari dua DataFrame yang berbeda.
- Ini sangat mirip dengan metode merge(), tetapi join() dirancang untuk menggabungkan DataFrame berdasarkan **index**.
    - meskipun juga memungkinkan penggabungan berdasarkan kolom dengan menggunakan parameter **on**.
- Secara default, join() melakukan **left join**

COMBINING DATAFRAME : JOIN

- Syntax:

```
DataFrame.join(other, on=None, how='left', lsuffix='',
rsuffix='', sort=False)
```

  - **other**: DataFrame lain, Series, atau list DataFrame yang akan digabungkan.
  - **on**: Nama kolom atau list nama kolom pada DataFrame pemanggil untuk digunakan sebagai kunci penggabungan. Jika tidak diatur, akan menggunakan index untuk penggabungan.
  - **how**: Menentukan cara penggabungan: left (default), right, outer, inner.
  - **lsuffix** dan **rsuffix**: Suffix yang ditambahkan ke kolom yang tumpang tindih pada DataFrame kiri dan kanan.
  - **sort**: Mengurutkan hasil gabungan berdasarkan kunci penggabungan jika True.

COMBINING DATAFRAME : MERGE VS JOIN

Fitur	join()	merge()
Default Key	Index dari DataFrame pemanggil.	Harus secara eksplisit ditentukan menggunakan on, left_on, atau right_on.
Kunci Penggabungan	Secara default berdasarkan index, tetapi bisa menggunakan on untuk kolom pada DataFrame pemanggil.	Bisa berdasarkan kolom atau index (menggunakan left_index dan right_index).
Tipe Penggabungan	Left join secara default, tapi bisa diubah (left, right, inner, outer) menggunakan how.	Mendukung semua jenis penggabungan (left, right, inner, outer) melalui how.
Fleksibilitas	Lebih terbatas dibandingkan merge(), lebih fokus pada penggabungan index.	Lebih fleksibel dalam menentukan kunci penggabungan dan tipe penggabungan.
Kolom Tumpang Tindih	Membutuhkan lsuffix dan rsuffix untuk menangani kolom yang tumpang tindih.	Sama, membutuhkan suffixes untuk menangani kolom yang tumpang tindih.
Sintaks	Dipanggil sebagai metode dari DataFrame (e.g., df1.join(df2)).	Dipanggil sebagai fungsi Pandas (e.g., pd.merge(df1, df2)).
Use Case	Sederhana dan cepat untuk penggabungan berdasarkan index atau ketika kunci penggabungan ada pada kolom DataFrame pemanggil.	Lebih sesuai untuk penggabungan kompleks, termasuk multiple keys, dan ketika kunci penggabungan perlu ditentukan lebih fleksibel.

COMBINING DATAFRAME : CONCAT

- Fungsi concat() di Pandas adalah cara yang sangat fleksibel untuk menggabungkan Series atau DataFrame sepanjang sumbu tertentu.
- Baik untuk menumpuk (stacking) vertikal maupun menggabungkan horizontal (side-by-side)

COMBINING DATAFRAME : CONCAT

- Syntax:

```
pd.concat(objs, axis=0, join='outer', ignore_index=False,
keys=None, levels=None, names=None, verify_integrity=False,
sort=False, copy=True)
```

  - **objs**: List atau dict dari objek pandas (Series, DataFrame) yang ingin digabungkan.
  - **axis**: {0, 1, ...}, default 0. Axis untuk penggabungan. 0 adalah vertikal (menambahkan baris), 1 adalah horizontal (menambahkan kolom).
  - **join**: {‘inner’, ‘outer’}, default ‘outer’. Cara menangani index pada sumbu lainnya. 'outer' akan menggabungkan semua index (union), sementara 'inner' hanya akan menggabungkan index yang tumpang tindih (intersection).
  - **ignore_index**: Bool, default False. Jika True, index yang dihasilkan tidak akan mempertahankan index asli dari objek yang digabungkan. Berguna jika index tidak penting.
  - **keys**: Sequence, default None. Jika diisi, akan menambahkan level multi-index pada sumbu penggabungan, menggunakan nilai keys sebagai level terluar.

COMBINING DATAFRAME : APPEND

- Metode append() di Pandas adalah cara yang cepat dan mudah untuk menambahkan baris dari satu DataFrame atau Series ke DataFrame lain.
- Secara fungsional, append() melakukan operasi yang mirip dengan pd.concat() tetapi dioptimalkan untuk menambahkan baris secara vertikal (axis=0).
- Metode ini mengembalikan DataFrame baru dengan baris yang ditambahkan ke DataFrame asli
- **Metode ini akan dihapuskan oleh python**

COMBINING DATAFRAME : APPEND

- Syntax:

```
DataFrame.append(other, ignore_index=False,
verify_integrity=False, sort=False)
```

  - **other**: DataFrame, Series/dict-like object, atau list dari objek-objek tersebut yang akan ditambahkan ke DataFrame pemanggil.
  - **ignore_index**: Jika True, index pada DataFrame hasil akan dibuat ulang dengan index yang berurutan dan tidak mempertahankan index dari DataFrame asli. Berguna jika index tidak penting atau jika Anda ingin menghindari index duplikat.

AGGREGATE, GROUP, PIVOT

AGGREGATE

- Aggregasi dalam konteks DataFrame Pandas adalah proses menggabungkan beberapa nilai menjadi satu nilai ringkasan.
- Ini sering digunakan dalam analisis data untuk merangkum informasi, seperti menghitung rata-rata, median, jumlah, atau standar deviasi dari sekelompok data.
- Fungsi agregasi sangat berguna saat bekerja dengan data yang dikelompokkan menggunakan metode groupby(), tetapi juga dapat digunakan langsung pada DataFrame atau Series untuk mendapatkan ringkasan data secara keseluruhan.

AGGREGATE

- Pandas menyediakan beberapa metode built-in untuk agregasi, contoh:
  - sum(): Menghitung total dari nilai.
  - mean(): Menghitung rata-rata.
  - median(): Menemukan nilai tengah.
  - min(): Nilai minimum.
  - max(): Nilai maksimum.
  - count(): Menghitung jumlah elemen.
  - std(): Standar deviasi.
  - var(): Varians.

AGGREGATE

- Fungsi agg() (singkatan dari aggregate) memungkinkan untuk menerapkan satu atau lebih operasi agregasi pada DataFrame atau hasil dari groupby(). Dapat menerima:
    - String nama fungsi ('sum', 'mean', dll.).
    - Fungsi (np.sum, np.mean, dll.).
    - List dari fungsi.
    - Dict dari kolom nama ke fungsi atau list fungsi.

GROUP

- Grouping dalam konteks DataFrame di Pandas adalah proses pengelompokan data berdasarkan satu atau beberapa kriteria dan kemudian menerapkan fungsi agregasi untuk merangkum atau menghitung statistik dari setiap grup.
- Ini adalah salah satu operasi yang paling bermanfaat dalam analisis data karena memungkinkan untuk melakukan analisis yang lebih mendalam terhadap subset data.
- Operasi grouping di Pandas dilakukan dengan metode groupby(), yang mengikuti prinsip split-apply-combine:
    - Split: Data dibagi menjadi grup berdasarkan kriteria tertentu.
    - Apply: Fungsi agregasi (seperti sum, mean, max, min, count, dll.) diterapkan secara terpisah ke setiap grup.
    - Combine: Hasil dari langkah "apply" digabungkan kembali menjadi sebuah struktur data baru.

PIVOT TABLE

- Pivot table adalah alat yang sangat berguna dalam analisis data untuk merangkum, mengatur, dan menganalisis data yang tersebar di satu atau lebih tabel.
- Fungsi pivot_table() di Pandas memungkinkan pengguna untuk dengan mudah melakukan operasi pivot pada data yang tersimpan dalam DataFrame.
- Operasi pivot ini mengubah data yang ada dalam format panjang menjadi format yang lebih lebar, yang memfasilitasi analisis yang lebih mendalam melalui pengelompokan dan agregasi data.

PIVOT TABLE

- Syntax:

```
pd.pivot_table(data, values=None, index=None, columns=None,
aggfunc='mean', fill_value=None, margins=False, dropna=True,
margins_name='All')
```

  - **data**: DataFrame yang akan di-pivot.
  - **values**: Kolom yang akan di-agregasi, yang menjadi nilai dalam tabel pivot yang dihasilkan.
  - **index**: Kolom dalam data yang nilai uniknya akan menjadi baris tabel pivot.
  - **columns**: Kolom dalam data yang nilai uniknya akan menjadi kolom tabel pivot.
  - **aggfunc**: Fungsi atau daftar fungsi yang digunakan untuk agregasi data. Defaultnya adalah 'mean'. Fungsi lain yang sering digunakan termasuk 'sum', 'count', 'min', 'max', dll.
  - **fill_value**: Nilai yang digunakan untuk menggantikan nilai NaN yang dihasilkan dari operasi pivot.
  - **margins**: Jika True, menambahkan total baris/kolom di akhir tabel pivot.
  - **dropna**: Jika True, kolom yang hanya berisi nilai NaN akan dihapus.
  - **margins_name**: Nama untuk baris/kolom total jika margins=True.

LAMBDA FUNCTION

LAMBDA FUNCTION

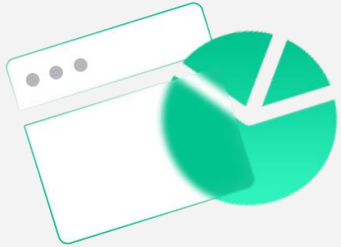
- Lambda function adalah cara cepat dan ringkas untuk menerapkan operasi atau fungsi tanpa perlu mendefinisikannya secara formal dengan def.
- Fungsi ini sangat berguna untuk operasi yang sederhana dan dapat dilakukan dalam satu baris.
- Dalam Pandas, lambda function sering digunakan bersama dengan metode seperti apply(), applymap(), dan map() untuk menerapkan fungsi ke DataFrame atau Series.

LAMBDA FUNCTION

- Syntax:

```
lambda arguments: expression
```

  - **arguments**: variabel input ke fungsi.
  - **expression**: ekspresi atau operasi yang diterapkan pada arguments dan nilai yang dihasilkan oleh ekspresi ini akan menjadi hasil fungsi.



Terima Kasih