Pandas: Data herstructureren

Dr. Cornelis Stal

April 28, 2022

[ ]:
```
import pandas as pd
```

1 Tabellen herstructureren

Deze tutorial is een vertaling van de *Pandas Tutorial* op https://pandas.pydata.org/pandas-docs/stable/getting_started/.

Data: voor deze tutorial zullen we gebruikmaken van de jaarlijkse vastgoedcijfers die bijgehouden en beschikbaar gemaakt worden door Statbel via deze link. We richten ons hierbij meer bepaald op de cijfers van verkoop van onroerende goederen (N) per jaar 2010-2021 voor de individuele gemeenten.

Hier hier om de data te downloaden.

[ ]:
```
statbel = 'https://statbel.fgov.be/sites/default/files/files/\
    documents/Bouwen%20%26%20wonen/2.1%20Vastgoedprijzen/NL_immo_jaar.xlsx'

vastgoed = pd.read_excel(statbel, sheet_name='Per gemeente', skiprows=2,
    parse_dates=["jaar"], usecols=['refnis', 'lokaliteit', 'jaar',
    'aantal transacties', 'mediaan prijs(€)', 'eerste kwartiel prijs(€)',
    'derde kwartiel prijs(€)'])
vastgoed['jaar'] = pd.DatetimeIndex(vastgoed['jaar'], yearfirst=True).year
vastgoed.head()
```

1.1 Rijen sorteren

Laten we de data eens sorteren op toenemende mediaanprijs:

[ ]:
```
vastgoed.sort_values(by="mediaan prijs(€)").head()
```

Met de `Series.sort_values()`-methodes worden rijen gesorteerd volgens een opgegeven kolom. De index meevolgen met deze sortering en komen dus nog altijd overeen met de oorspronkelijke indexering.

Aan het `by`-argument kunnen we in plaats van één enkele string met een kolomhoofding ook een lijst met kolomhoofdingen meegeven. In onderstaand voorbeeld sorteren we eerst op de naam van de gemeente, en vervolgens op de mediaanprijs:

[ ]:
```
vastgoed.sort_values(
    by=['lokaliteit', 'mediaan prijs(€)'],
    ascending=False
).head()
```

Gebruikshandleiding: meer informatie over het sorteren van tabelen wordt gegeven in de sectie over het sorteren van data in de handleiding.

1.2 Tabel omzetten van lang formaat naar breed formaat

Laten we eens een beperkte subset nemen uit de Vlaamse vastgoeddata. We beperken ons tot de statistieken die gegeven zijn voor de gemeenten Gent, Merelbeke en Melle, en steken deze data in een nieuwe `DataFrame` genaamd `sample`:

[ ]:
```
sample = vastgoed.loc[vastgoed["lokaliteit"].isin(
    ['GENT', 'MERELBEKE', 'MELLE'])]
sample = sample.rename(
    columns={'refnis': 'NISCODE', 'mediaan prijs(€)' : 'MEDIAAN',
    'jaar': 'JAAR', 'lokaliteit': 'NAAM', 'aantal transacties': 'TRANSACTIES',
    'eerste kwartiel prijs(€)': 'Q1', 'derde kwartiel prijs(€)': 'Q3'}
)
sample.head()
```

We kunnen deze data ook voorstellen met een verzameling kolommen voor iedere afzonderlijke gemeente:

[ ]:
```
sample_pivot = sample.pivot(
    index="JAAR",
    columns="NAAM",
    values=["MEDIAAN", 'Q1', 'Q3', 'TRANSACTIES']
)
sample_pivot.head()
```

De `pivot()`-functie is bedoeld om data om te vormen, ofwel om te 'reshaping'. Het is hierbij slechts vereist om te beschikken over een enkele waarde voor iedere combinatie van index en kolom.

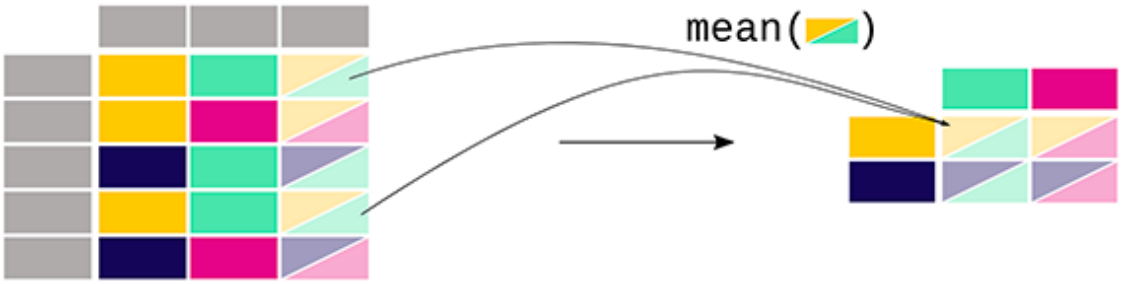
Aangezien `pandas` het plotten van meerdere kolommen standaard ondersteunt, is de conversie van lange tabellen naar brede tabellen zeer interessnt om data voor te stellen in bijvoorbeeld tijdreeksen. We verwijzen hiervoor terug naar de corresponderende tutorial over het plotten van data:

[ ]:
```
sample_pivot['MEDIAAN'].plot()
```

Opmerking: wanneer de `index`-parameter niet gedefinieerd is zal de bestaande index (rij labels) gebruikt worden.

Gebruikshandleiding: voor meer informatie over de `pivot()`-functie verwijzen we door naar de sectie over het maken van draaitabellen in de handleiding.

1.3 Draaitabellen ('pivot tables')



Draaitabellen zijn niet enkel interessant om de vorm van een tabel te veranderen. We kunnen ze ook gebruiken om voor een bepaalde verzameling waarden statistieken te berekenen, bijvoorbeeld de gemiddelde mediaanprijs voor de verschillende jaren en de bijbehorende standaardafwijking:

[ ]:
```
sample.pivot_table(
    index='NAAM', values=['MEDIAAN', 'Q1', 'Q3'],
    aggfunc=['mean', 'std']
)
```

Bij het gebruik van de `pivot()`-functie zullen data enkel herschikt worden. Wanneer meerdere waarden gecombineerd moeten worden met het oog op het berekenen van statistieken (zoals in ons voorbeeld voor het berekenen van de gemiddelde mediaanprijs en de standaardafwijking), wordt gebruik gemaakt van de `pivot_table()`-functie. De gewenste aggregaatfuncties (zoals het gemiddelde of standaardafwijking) geven we mee als attribuut.

Draaitabellen worden vaak aangemaakt met behulp van spreadsheet-software, zoals MS Excel. Wanneer we naast de statistieken per cel ook geïnteresseerd zijn in samenvattingen per kolom (zoals kolomtotalen), stellen we de `margin`-parameter in op `True`:

[ ]:
```
sample.pivot_table(
    values=['MEDIAAN', 'Q1', 'Q3'],
    index='NAAM',
    aggfunc='mean',
    margins=True,
)
```

Gebruikshandleiding: voor meer informatie over de `pivot_table()`-functie verwijzen we door naar de sectie 'pivot tables' in de handleiding.

Opmerking: de `pivot_table()`-functie is gelijkaardig aan de `groupby()`-functie. Vergelijkbare resultaten zullen we verkrijgen als we de `groupby()`-functie gebruiken als we groeperen over de kolom `NAAM`:

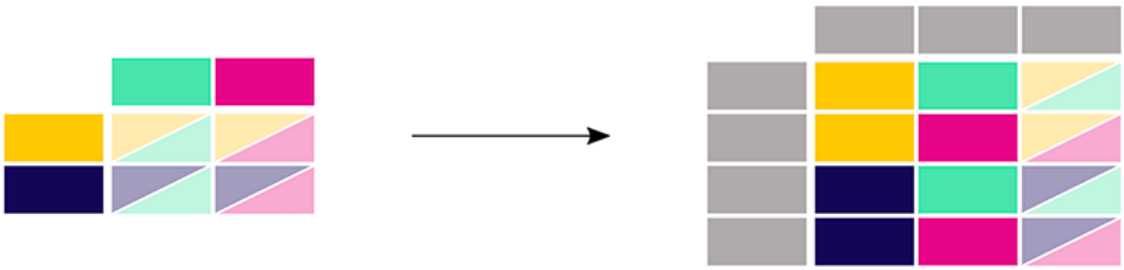
```
sample.groupby('NAAM')['MEDIAAN', 'Q1', 'Q3'].mean()
```

Gebruikshandleiding: de combinatie van de `groupby()`-functie en de `unstack()`-functie maken enkele interessante zaken mogelijk. Voor meer informatie verwijzen we naar de sectie 'combining stats and groupby' in de handleiding.

1.4 Breed naar lang tabelformaat

We vertrekken opnieuw van een breed-formaat tabel zoals aangemaakt in de vorige sectie, en zullen teruggaan naar een lang-formaat tabel:

[ ]:
```
sample_pivot = sample.pivot(index="JAAR", columns="NAAM",
    values=["MEDIAAN", 'Q1', 'Q3'])
sample_pivot.head()
```



Laten we nu terug alle waarden in een enkele kolom verwerken, met een bijbehorende kolom voor de corresponderende variabele:

[ ]:
```
sample_lang = sample_pivot.melt()
sample_lang.columns = sample_lang.columns.fillna('VARIABLE')
sample_lang = sample_lang.rename(
    columns={'value': 'WAARDE', 'variabele': 'VARIABELE'})
sample_lang.head()
```

De `pandas.melt()`-methode van een `DataFrame` converteert de datatabel van een breed-formaat tabel naar een lang-formaat tabel. De oude kolom kolomhoofdingen worden nu waarden van nieuw aangemaakte kolommen. Deze methode zal alle kolommen samenvoegen die niet vermeld worden bij de attribuut `id_vars`. Daarnaast worden er telkens twee nieuwe kolommen aangemaakt: een kolom met de oorspronkelijke kolomhoofding en een kolom met de celwaarden zelf. Deze laatste kolom krijgt standaard de naam `value`, maar die kunnen we eenvoudig hernoemen.

In de volgende code worden enkele parameters voor de `pandas.melt()`-methode verder uitgediept:

[ ]:
```
sample_lang = sample.melt(id_vars=["NAAM", "NISCODE"],
    value_vars=["MEDIAAN", "Q3", "Q1", "TRANSACTIES"],
    value_name="WAARDE", var_name="VARIABELE"
)
```

```
sample_lang.head()
```

De bovenstaande code geeft een zeer gelijkaardig resultaat. De volgende argumenten worden meegegeven: - `id_vars`: definieert de kolommen die als identifiers worden gebruikt; - `value_vars`: definieert welke kolommen moeten worden samengevoegd; - `value_name`: hier wordt een zelf te kiezen naam meegegeven door de kolom met de waarden. Standaard wordt de naam "`value`" gebruikt; - `var_name`: hier wordt een zelf te kiezen naam meegegeven door de kolom met de variabelen. Standaard wordt de naam " `variable`" gebruikt.

Samengevat zijn de argumenten `value_name` en `var_name` zelf te definieren namen voor de twee aan te maken kolommen. De kolommen die samengevoegd moeten worden, zijn gedefinieerd door de argumenten `id_vars` en `value_vars`.

> **Gebruikshandleiding:** voor meer informatie over de conversie van brede naar lange tabellen met behulp van de pandas.melt()-functie verwijzen we naar de sectie getiteld 'reshaping by melt' in de handleiding.

1.5 Te onthouden:

- Voor het sorteren van een of meerdere kolommen maken we gebruik van de `sort_values()`-functie;
- De `pivot()`-functie gebruiken we om de vorm van de data te veranderen. Met de `pivot_table`-functie kunnen we complexere draaitabellen maken met beschrijvende statistieken;
- Het tegenovergestelde resultaat van de `pivot()`-functie (lang naar breed formaat) is de `melt()`-functie (breed naar lang formaat).

> **Gebruikshandleiding:** een volledig overzicht over het herstructureren van data is terug te vinden in de bijbehorende sectie getiteld 'reshaping and pivoting' in de handleiding.