

Pandas: Kolommen toevoegen

Dr. Cornelis Stal

April 28, 2022

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

1 Kolommen toevoegen

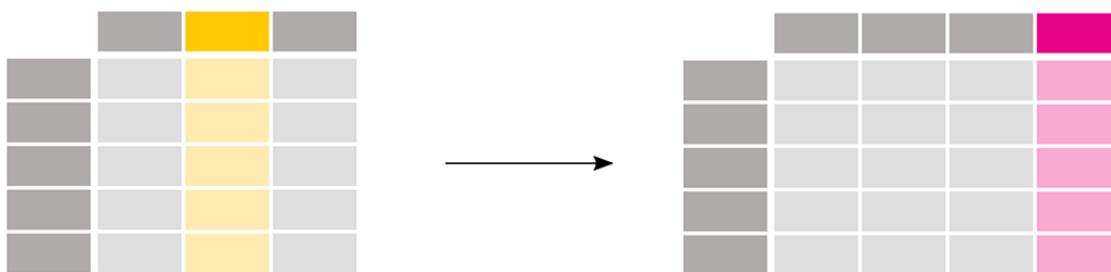
Deze tutorial is een vertaling van de *Pandas Tutorial* op https://pandas.pydata.org/pandas-docs/stable/getting_started/.

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/\n        documents/Bouwen%20%26%20wonen/2.1%20Vastgoedprijzen/NL_immo_jaar.xlsx'\n\nvastgoed = pd.read_excel(statbel, sheet_name='Per gemeente', skiprows=2,\n                        usecols=['refnis', 'lokaliteit', 'jaar', 'aantal transacties',\n                                'mediaan prijs(€)', 'eerste kwartiel prijs(€)', 'derde kwartiel prijs(€)'])\nvastgoed.head()
```

1.1 Nieuwe kolommen aanmaken op basis van bestaande data



Laten we de bedragen in € eens uitdrukken in M€ (miljoen):

```
[ ]: vastgoed["M_m"] = vastgoed["mediaan prijs(€)"] / 1000000\nvastgoed["Q1_m"] = vastgoed["eerste kwartiel prijs(€)"] / 1000000\nvastgoed["Q3_m"] = vastgoed["derde kwartiel prijs(€)"] / 1000000
```

```
vastgoed.head()
```

Om een nieuwe kolom aan te maken gebruiken we vierkante haakjes [], waarin de naam van de nieuwe kolom geschreven wordt. Inclusief de verwijzing naar de `DataFrame` waarbinnen deze nieuwe kolom moet worden opgeslagen, zal dit geheel aan de linkerzijde van de toekenning geplaatst worden.

Opmerking: de berekening van nieuwe waarden gebeurt element per element ofwel *element-wise*. Dit betekent dat iedere waarde in de gegeven kolommen gedeeld worden door een miljoen. Het is bijgevolg niet vereist om te werken met lussen en iteraties om zelf regel per regel te verwerken.



Veronderstellen we nu dat de totale waarde van het vastgoed voor een bepaalde gemeente berekend kan worden door het product van de mediaan en het aantal transacties. Het resultaat slaan we op in een nieuwe kolom:

```
[ ]: vastgoed['waarde'] = (vastgoed['M_m'] * vastgoed['aantal transacties'])
      vastgoed.head()
```

Deze berekening vindt opnieuw element per element (*element-wise*) plaats. De vermenigvuldiging * wordt bijgevolg toegepast op alle rijen.

Andere wiskundige operatoren (+, -, *, /) en logische operatoren (<, >, =,...) functioneren eveneens element per element. Deze laatste groep operatoren was reeds beproken bij de tutorial over [het selecteren van data uit `DataFrames` \('subsets'\)](#), waarbij we rijen gefilterd hebben uit een tabel met behulp van conditionele expressies.

Voor meer geavanceerde logica kunnen we arbitraire Python code meegeven als attribuut aan de `apply()`-methode.

De namen van de verschillende kolommen zijn verre van ideaal (speciale tekens, spaties, lange namen, ...). We zullen de kolomhoofdingen daarom eens hernoemen:

```
[ ]: vastgoed_renamed = vastgoed.rename(
      columns={'refnis': 'NIS', 'lokaliteit': 'NAAM', 'jaar': 'JAAR',
              'aantal transacties': 'AANTAL', 'mediaan prijs(€)': 'MEDIAAN',
              'eerste kwartiel prijs(€)': 'Q1', 'derde kwartiel prijs(€)': 'Q3'},
      errors="raise"
)
      vastgoed.head()
```

We gebruiken de `rename()`-functie om de namen van kolommen te veranderen. Het hernoemen

van rijen is echter ook mogelijk met deze functie. Als attribuut voor deze functie geven we een **dictionary** mee, waarbij de oorspronkelijke kolomnamen als sleutel (**key**) worden vermeld. De nieuwe namen zijn de bijbehorende waarden (**value**).

Het hernoemen van kolomhoofdingen is niet beperkt tot concrete tekstvelden. We kunnen ook zogenaamde ‘mapping’-functies gebruiken. Dit laat toe om bijvoorbeeld alle hoofdingen te converteren naar hoofdletters:

```
[ ]: vastgoed_renamed = vastgoed_renamed.rename(columns=str.upper)  
vastgoed_renamed.head()
```

Gebruikshandleiding: meer details over het hernoemen van rijen en kolommen kan gevonden worden in de [corresponderende sectie](#) in de handleiding.

1.2 Te onthouden:

- Nieuwe kolommen worden aangemaakt door de output aan de **DataFrame** toe te kennen met de nieuwe naam tussen vierkante haakjes [];
- Operaties op rijen worden element per element *element-wise* waardoor geen lussen nodig zijn;
- Gebruik de **rename()**-functie in combinatie met een **dictionary**-object of in combinatie met een functie om rijlabels of kolomnamen te hernoemen.

Gebruikshandleiding: de handleiding bevat een aparte sectie over het [toevoegen en verwijderen van kolommen](#). We zullen deze operaties in de komende tutorials nog aan bod laten komen.