Obsługa plików z danymi

Michał Gałka

Pliki CSV

Format CSV

- CSV *(ang. Comma Separated Values)*
- Został opisany w dokumencie RFC4180.
- Istnieje wiele implementacji formatu CSV.

Format CSV

- Linie w pliku zakończone są znakiem końca wiersza (CRLF).
  - Ostatnia linia może nie zawierać znaku końca.
  - Jeśli znak CRLF jest elementem pola musi być ujęty w cudzysłów.
- Standardowym separatorem pól jest przecinek.
  - Zdarza się, że separatorem jest średnik, albo tabulator (niezalecane przez standard)
  - Separator musi być jednoznaczny dla całego pliku.

Format CSV

- Zawartość pól może być ujęta w cudzysłów.
- Wartości zawierające znak separatora muszą być ujęte w cudzysłów.
- Jeśli cudzysłów jest wartością pola musi zostać podwojony i ujęty w cudzysłów.

Format CSV

- Pierwsza linia może zawierać nagłówki kolumn (nazwy pól).
- Spacje i inne białe znaki przynależą do pola.

CSV w Pandas

```
import pandas as pd
pd.read_csv('data/iris.csv')
pd.read_csv('https://example.com/iris.csv')
pd.read_csv('data/iris.csv.gz')
pd.read_csv('data/iris.zip')
```

Wybrane parametry funkcji `read_csv()`

- `sep` (domyślnie: `','`)
  - Określa separator dla danych w plikach CSV
  - `None` automatyczne wykrycie separatora (wymusza użycie silnika w Pythonie)
  - separator dłuższy niż 1 znak i różne od wyrażenia `\s+` wymusza użycie silnika w Pythonie.
  -

Wybrane parametry funkcji `read_csv()`

- `header` (domyślnie: `'infer'`)
  - Określa liczbę linii użytych jako nagłówek
  - Zachowanie domyślne powoduje "wywnioskowanie" nazw kolumn z pierwszego wiersza
    - Jeśli nazwy kolumn nie zostały przekazane do funkcji to zachowuje się tak jak `header=0` i nazwy kolumn pobierane są z pierwszego wiersza
    - Jeśli nazwy kolumn zostały przekazane do funkcji to zachowuje się jak `header=None`
  - Jawne ustawienie `header=0` powoduje nadpisanie nazw kolumn przekazanych w parametrze `names`
- `names` (domyślnie: `None`)
  - Lista nazw kolumn do używana w przypadku braku nagłówka.
  - W przypadku braku nagłówka należy jawnie ustawić `header=None`
- `index_col`

Wybrane parametry funkcji `read_csv()`

- `dtype`
  - Nazwa typu, lub słownik mapujący nazwy kolumn na dany typ.

```
{'a': np.float64, 'b': np.int32}
```

  - Jeśli ustawiony jest parametr `converters`, konwertery zostaną użyte zamiast `dtype`.
- `converters`
  - Słownik, którego wartościami są funkcje, a kluczami nazwy kolumn, lub liczby całkowite.

Wybrane parametry funkcji `read_csv()`

- `true_values`, `false_values`
  - listy wartości które mają być traktowane jako True/False.
- `skiprows`
  - lista numerów wierszy do pominięcia przy wczytywaniu (indeksowanych od 0)
  - Obiekt wykonywalny pobierający wiersz jako parametr i zwracający `True`, albo `False`.
    - `True` - wiersz zostaje pominięty
    -

Wybrane parametry funkcji `read_csv()`

- na_values
  - Dodatkowe wartości, które będą rozpoznawane jako `NA/NaN`
  - Jeśli wartością jest słownik, można wyspecyfikować dodatkowe wartości dla konkretnych kolumn.
  - Domyślne wartości interpretowane jako `NaN`

```
'#N/A', '#N/A N/A', '#NA', '-1.#IND', '-1.#QNAN',
'-NaN', '-nan', '1.#IND', '1.#QNAN', 'N/A', 'NA',
'NULL', 'NaN', 'n/a', 'nan', 'null'
```

Wybrane parametry funkcji `read_csv()`

- `parse_dates`
  - Wartość logiczna. `True` - parser powinien spróbować przetworzyć indeks na datę.
  - Lista liczb całkowitych, lub nazw kolumn.
    - `[1, 2, 3]` - parser powinien przetworzyć kolumny 1, 2 oraz 3 na datę.
  - Lista list.
    - `[[1, 3]]` - parser powinien połączyć kolumny 1 i 3 i przetworzyć to połączenie na datę.
  - Słownik
    - `{'foo': [1, 3]}` - parser powinien przetworzyć kolumny 1 i 3 na datę i nadać im nazwę "foo".
  - Jeśli wartość nie daje się poprawnie skonwertować cała kolumna pozostaje niezmieniona, a dane będą miały typ `object`.
- `date_parser` (domyślnie: `None`)
  - Funkcja jaka powinna zostać użyta do parse'owania daty.
  - Domyślnie używana jest `dateutil.parser.parser`.

Wybrane parametry funkcji `read_csv()`

- `thousands` (domyślnie: `None`)
  - Znak separujący tysiące.
- `decimal` (domyślnie: '.')
  - Znak separujący część dziesiętną.
- encoding (domyślnie: `None`)
  - Standard kodowania znaków używany przy odczycie/zapisie (np. 'utf-8').
  - Lista standardów kodowania w Pythonie: https://docs.python.org/3/library/codecs.html#standard-encodings

Obsługa wartości NA/NaN

- `pandas.DataFrame.dropna()`
  - Usuwa kolumny/wiersze zawierające wartości `NaN`
- Wybrane parametry:
  - `axis`:
    - 0 lub 'index' - usuwa wiersze zawierające wartości `NaN`.
    - 1 lub 'columns' - usuwa kolumny zawierające wartości `NaN`.
  - `how` (domyślnie: `'any'`):
    - `any` - Jeśli zawiera przynajmniej jedną wartość `NaN`.
    - `all` - Jeśli nie zawiera innych wartości niż `NaN`.
    - `inplace` (`True`/`False`) - Zmienia bieżący obiekt, lub tworzy kopię.

Obsługa wartości NA/NaN

- `pandas.DataFrame.fillna()`
  - `value` - Wartość użyta do wypełnienia. Akceptowalne typy to `scalar`, słownik, `Series`, `DataFrame`. Wartość nie może być listą.
  - `axis`
    - 0 lub 'index' - zmienia wiersze zawierające wartości `NaN`.
    - 1 lub 'columns' - zmienia kolumny zawierające wartości `NaN`.
  - `inplace` (`True`/`False`) - Zmienia bieżący obiekt, lub tworzy kopię.
  - `method`
    - `pad`/`ffill` - Używa ostatniej poprawnej wartości do wypełnienia kolejnej luki.
    - `backfill`/`bfill` - Używa następnej wartości do wypełnienia luki.

Podział wartości w kolumnie

- `df.col.str.split(pat=None, n=-1, expand=False)`
  - `pat` - łańcuch znaków, lub wyrażenie regularne określające separator, `None` oznacza podział względem białych znaków.
  - `n` - Ilość podziałów po których należy zaprzestać operacji. `0`, `-1` lub `None` oznacza zachowanie wszystkich podziałów
  - `expand` - Oznacza, czy podział powinien stworzyć nowe kolumny

Obsługa plików Excel

```
import pandas as pd
pd.read_excel('data/iris.xls')
```

Odczyt danych z pliku Excel

- Funkcja `read_excel()` posiada wiele parametrów identycznych jak funkcja `read_csv`.
- Parametr `sheet_name` pozwala określić z których arkuszy w pliku należy pobrać dane.

Parametr `sheet_name`

- Pozwala określić nazwę lub numer arkusza do odczytu.
- `None` odczytuje wszystkie arkusze.
- Domyślnie przyjmuje wartość `0`.
- Typ wartości zwracanej przez funkcję zależy od parametru, który określał zakres arkuszy do odczytu:
  - `str`/`int` - zwraca `DataFrame`.
  - `list`/`None` - zwraca słownik z kluczami będącymi nazwami kolumn oraz wartościami `DataFrame`

Parametr `usecols`

- Wartość parametru `usecols` określa, które kolumny powinny trafić do wynikowego `DataFrame`.
  - `None` - Wszystkie kolumny.
  - `int` - Nr ostatniej kolumny do włączenia do wyniku.
  - lista `int` - lista numerów kolumn.
  - `str` - rozdzielona przecinkami lista kolumn i zakresów kolumn w excelu.

Grupowanie i agregacja danych

Złączenia ramek