Pandas

**import pandas as pd**

Wczytywanie pliku CSV:

**df = pd.read\_csv(‘ścieżka do pliku’)**

Sprawdzanie ilości kolumn i wierszy

**df.shape**  *-> to zwraca (a, b), gdzie a to wiersze, b kolumny*

Informacje o zawartości otwartego pliku

**x.info()**



**wyświetla wszystkie wiersze/kolumny**

Pokazuje pierwsze x wierszy

**df.head(x)**

Pokazuje ostatnie x wierszy

**df.tail(x)**

Pokazuje pary klucz-wartość jako tabela

people= {

"first" :["Ania", "Marysia", "Ola"], -----🡪 **nazwa = pd.DataFrame(people)**

"last" :["Osten", "Marck", "Sir"],

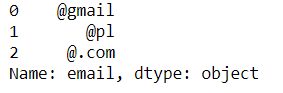
"email": ["@gmail", "@pl", "@.com"]}

co daje :

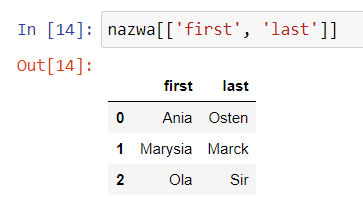


„Wyciąganie” wartości z powyższej tabeli

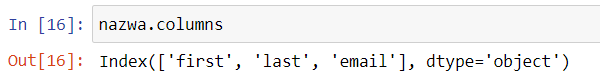
np. **nazwa[‘email’]**



Kilka wartości



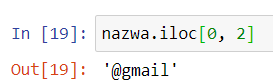
Pokazuje jakie mamy kolumny w pliku



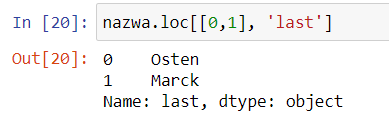
Wyciągamy, który chcemy wiersz oraz kolumnę:

są na to dwa sposoby iloc[] oraz loc[], gdy używamy iloc to odnosimy się do indeksów, natomiast przy loc wyszukujemy po nazwie wiersza/kolumny

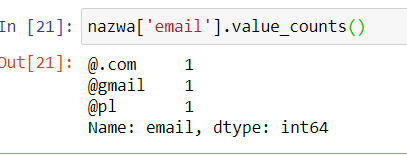
np. Chcemy wziąć pierwszy wiersz oraz trzecią kolumnę za pomocą iloc(indeksowanie zaczyna się od 0)



np. Chcemy zobaczyć pierwszy i drugi wiersz oraz drugą kolumnę za pomocą loc



Zliczanie występujących wartości dla danej kolumny



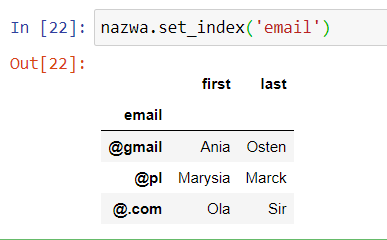
Indeksowanie wierszy

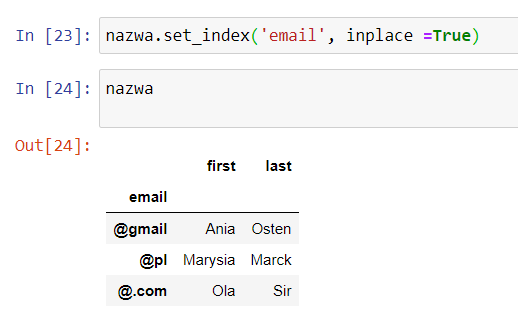
Wcześniej wiersze miały domyślne indeksowanie 0, 1 itd. ale możemy zaindeksować wiersz wybrana wartością, która się w nim znajduje

Mamy naszą tabelę:



i teraz chcemy aby indeksami wierszy był adres email ponieważ powinien on być unikatowy dla każdej osoby



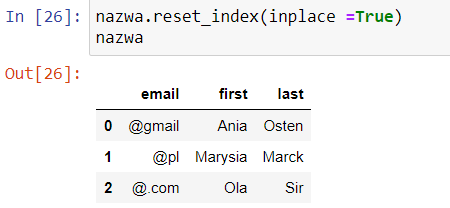
****Jednak jak będziemy wyświetlali później nasze dane ‘nazwa’ to nie będziemy widzieć takiego ustawienia tylko normalne jak w pierwszej tabeli w tym paragrafie. Jak chcemy żeby było widoczne jak powyżej cały czas to d o instrukcji trzeba dodać **inplace = True**

Sprawdzanie jakie mamy indeksy



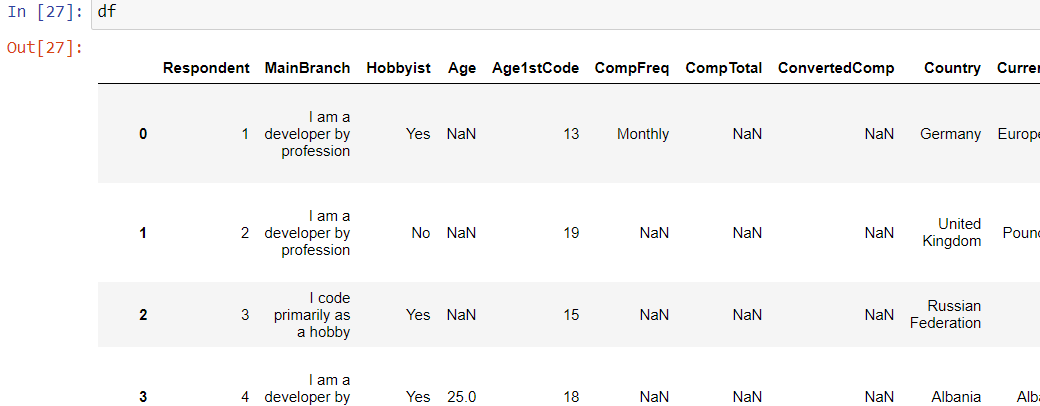
Usuwanie przyjętego indeksowania wierszy

Czyli jak ustaliliśmy, że nazwa email będzie naszym indeksem dla wierszy ale teraz już nie chcemy takiego indeksowania



Ustawianie indeksowania od razu przy wczytywaniu pliku

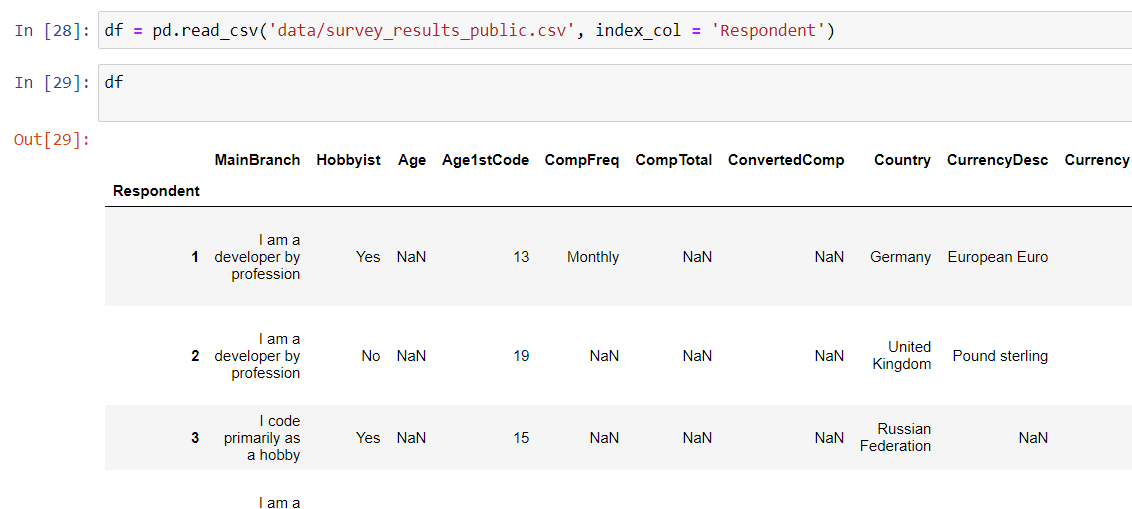
Mamy takie dane:



I chcemy żeby indeksem była kolumna Respondent

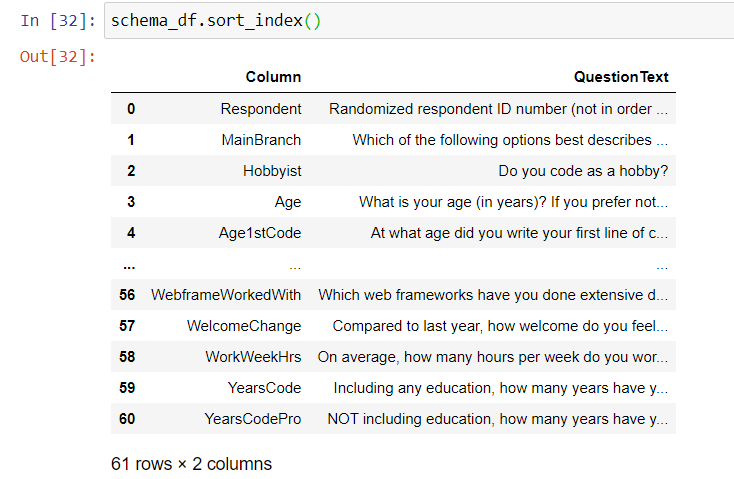
a więc przy wczytywaniu dodajemy:

**index\_col = ‘Respondent’**

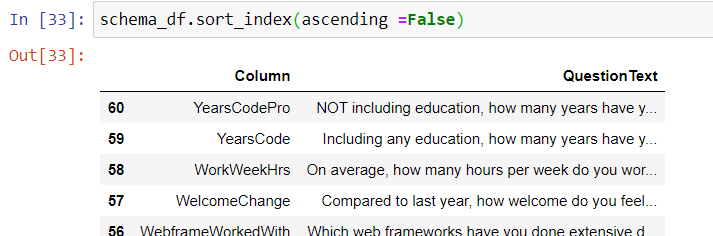


Sortowanie indeksu

\*\* alfabetycznie lub rosnąco(liczbowo)



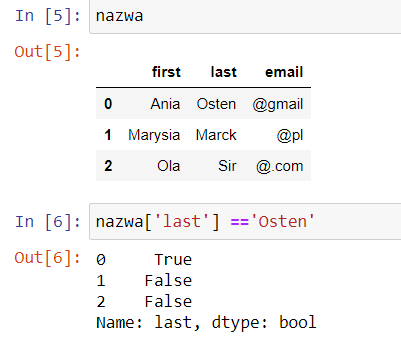
\*\* malejąco



Te zmiany będą widoczne tylko raz, natomiast jeśli chcemy żeby takie sortowanie już zostało na stałe to musimy dodać w instrukcji sortującej **inplace= True**

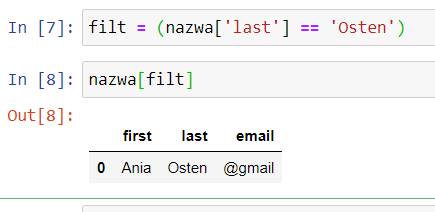
# 4. Filtrowanie

Szukanie danej wartości w kolumnie

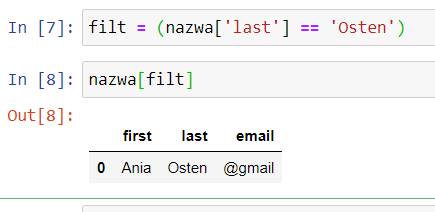


Przypisuje albo wartość True albo False w zależności czy ‘Osten’ pasuje do danego wiersza

Wyniki naszego wyszukiwania możemy przypisać do zmiennej



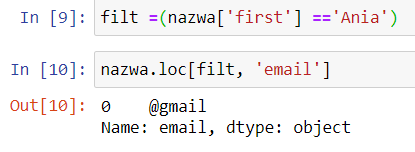
Teraz jeśli podamy filt tak jak podawaliśmy kolumny czyli nazwa[filt] otrzymamy nie wartości prawda fałsz tylko te osoby które faktycznie tak mają na nazwisko



Wyświetlanie tylko jednej rzeczy z wyfiltrowanego wiersza

Użyjemy do tego funkcji loc, której pierwszym parametrem był szukany wiersz i tutaj należy podać kryterium naszego filtru lub zmienną do niego przypisaną a drugi parametr to ten który szukamy dla wyfiltrowanej osoby

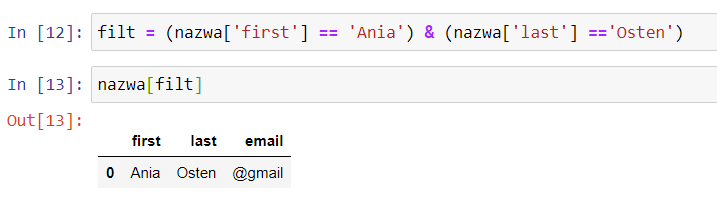
np. chcemy znaleźć osoby których imię to Ania ale chcemy wyszukać wśród tych osób tylko ich adresy email a nie wszystkie informacje



Filtrowanie warunkowe z and i or

Działa to bardzo intuicyjnie tak jak w normalnych programach, dodajemy do naszego filtru kolejną instrukcję

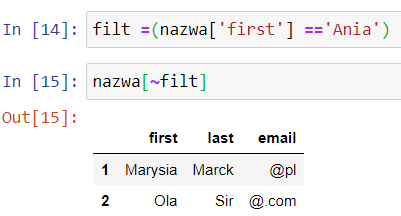
np. chcemy uzyskać osobę o imieniu Ania która ma na nazwisko Osten



And to symbol & , or działa tak samo tylko ze znakiem |

Zaprzeczenie filtru

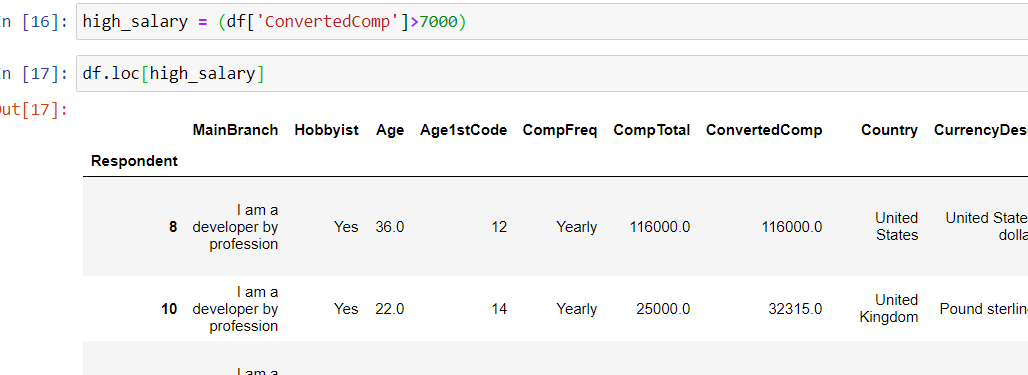
Mamy nasz filtr wyszukujący osobę o imieniu Ania a teraz chcemy żeby wyszukiwał wszystkich tylko nie osoby o imieniu Ania



Filtrowanie z >, < , =

Szukamy osób zarabiających więcej niż 7000

Dane dotyczące pensji znajdują się w kolumnie ConvertedComp



Filtrowanie z osobną listą

Chcemy znaleźć kilka państw w naszej tabeli

Tworzymy listę z państwami, które chcemy znaleźć

I dodajemy do filtru metodę isin()



Wyszukiwanie frazy w ciągu znaków

W naszych danych osoby programujące w danych językach mają je zapisane w jednym ciągu np. Python\Java\Shell itd. Teraz jeśli chcemy znaleźć osoby programujące w Pythonie musimy posłużyć się metodą znajdującą wśród znaków

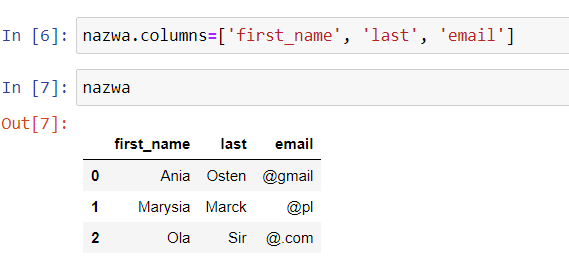


Instrukcja na= False jest umieszczona żeby nie było błędu przy NaN

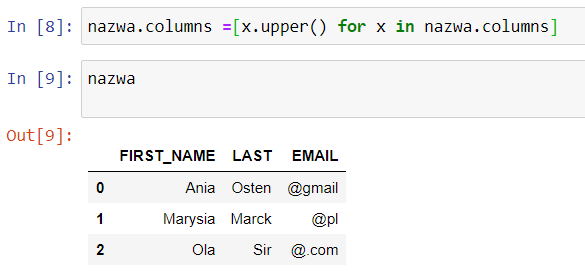
# 5. Aktualizowanie wierszy i kolumn

Zmienianie nazw kolumn

Chcemy zmienić nazwę kolumny ‘first’ na ‘first\_name’ natomiast reszta kolumn pozostaje bez zmian



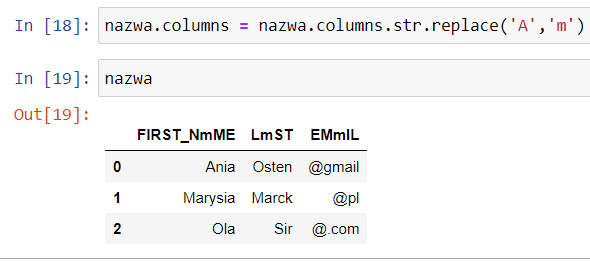
Zmienianie nazw kolumn na napisane dużymi literami



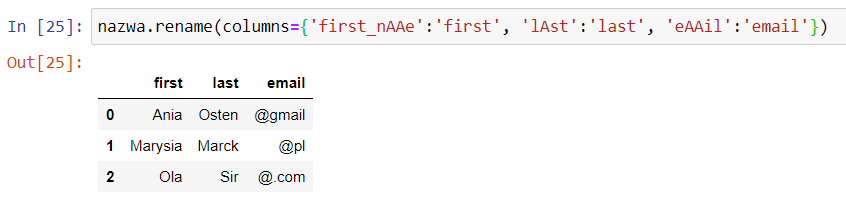
zamiana na małe litery tak samo tylko z funkcją **.lower()**

Zamienianie dowolnej litery/znaku czy ciągu liter/znaków na inny w nazwach kolumn

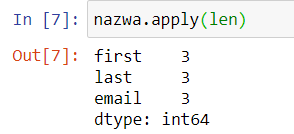
np. zamieniamy każdą literę A na m



Zmienianie wybranych kolumn (nazw) na inne

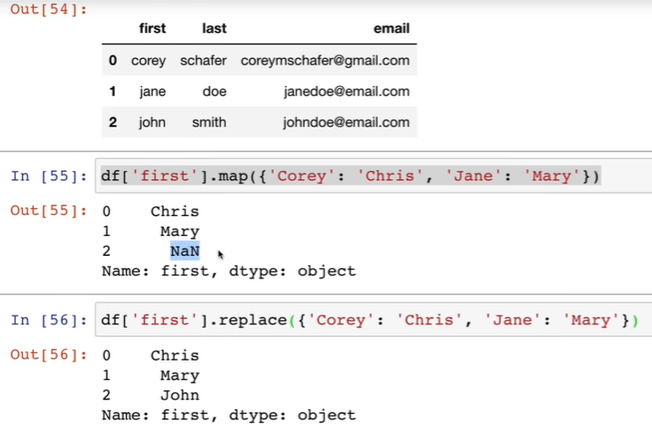


Sprawdzanie ile każda kolumna ma wierszy

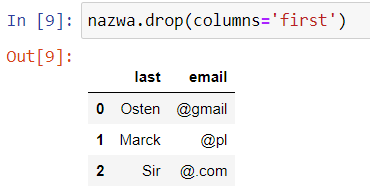


Zamiana wartości

Pokazane dwa sposoby



# 6. Dodawanie kolumn i wierszy

Usuwanie kolumny

Dodawanie wiersza

