

1. JSON & XML & CSV & XLS

Inspiracją do zadań jest adres

<https://github.com/jdorman/awesome-json-datasets> . Można tu znaleźć o wiele więcej serwisów zwracających dane w formacie json. Drugim źródłem może być <https://github.com/toddmotto/public-apis#anti-malware> . Aby podglądać dane w jsonie, które czasami są nieczytelnie sformatowane, polecam <https://jsonformatter.curiousconcept.com/> .

1.1 [ADRES NIE DZIAŁA] Napisz funkcję bitcoinWaluta(waluta), która zwraca notowanie 1 bitcoina w zadanej walucie. Podpowiedź:

<https://api.bitcoinaverage.com/ticker/global/USD/> i

<https://api.bitcoinaverage.com/ticker/global/all>

1.2 Używając <http://httpbin.org/ip> zwróć swój adres IP.

1.3a Napisz funkcję ciekawostkiMatematyczne(liczba, typ), która zwróci ciekawostkę danego typu o zadanej liczbie. Typy ciekawostek to trivia, math, date, lub year. Podpowiedź: <http://numbersapi.com/> i

<http://numbersapi.com/5/math> .

1.3b Nasze ulubione ciekawostki o liczbach mogą być także zwracane w formacie JSON. I tak np. <http://numbersapi.com/1989/year?json> zwróci nam ciekawostkę o roku 1989 w formacie JSON. Napisz analogiczną funkcję jak w 1.2a, która operuje na formacie JSON.

1.3c Napisz funkcję masoweCiekawostki(liczbaStart, liczbaKoniec, typ), która zwraca ramkę danych Pandas o 2 kolumnach: pierwsza to liczba, druga to ciekawostka z tą liczbą związana. Bazuj na formacie JSON. Spróbuj napisać dwie wersje tej funkcji: jedna pobiera każdą ciekawostkę z osobna, a druga pobiera je w jednym zapytaniu. Bonus: Niech zwracana ramka danych ma 3 kolumny, gdzie trzecia oznacza typ ciekawostki (na

stałe pobieramy wszystkie 4 typy ciekawostek dla wszystkich liczb). Dodaj parametr, który decyduje, czy dane są posortowane po liczbie, czy typie ciekawostki.

1.4 Napisz funkcję `pogoda(miasto)`, która pobierze pogodę dla zadanego miasta używając Yahoo Weather API. Przykładowe zapytanie dla Warszawy:

[https://query.yahooapis.com/v1/public/yql?q=select%20wind%20from%20weather.forecast%20where%20woeid%20in%20\(select%20woeid%20from%20geo.places\(1\)%20where%20text=%27warsaw%27\)&format=json](https://query.yahooapis.com/v1/public/yql?q=select%20wind%20from%20weather.forecast%20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text=%27warsaw%27)&format=json)

Analogiczne zapytanie dla XML:

[https://query.yahooapis.com/v1/public/yql?q=select%20wind%20from%20weather.forecast%20where%20woeid%20in%20\(select%20woeid%20from%20geo.places\(1\)%20where%20text=%27warsaw%27\)&format=xml](https://query.yahooapis.com/v1/public/yql?q=select%20wind%20from%20weather.forecast%20where%20woeid%20in%20(select%20woeid%20from%20geo.places(1)%20where%20text=%27warsaw%27)&format=xml)

Wprawiać się w zapytaniach można na stronie <https://developer.yahoo.com/weather/>.

W przypadku XML dostań się do danych poprzez:

- a) Indeksami
- b) Poprzez nazwy dzieci (ciąg wywołań `find()`)
- c) Poprzez różne XPath (np. dokładna ścieżka, wszystkie dzieci elementu “item”, wszystkie elementy zawierające atrybut “temp”, wszystkie dzieci (także te niebezpośrednie) o nazwie “condition”)

1.5 Na stronie <http://www.vizgr.org/historical-events/> możemy otrzymać różne historyczne zdarzenia. Napisz funkcję `historyczneZdarzenia(dateBegin, dateEnd)`, która pobiera dane dla zadanego przedziału czasowego i zwraca ramkę danych Pandas. Przykładowe zapytanie:

http://www.vizgr.org/historical-events/search.php?format=json&begin_date=20110101&end_date=20151231&lang=en .

Czy da się wykonać to zadanie poprzez format JSON? Dlaczego?

Analogicznie, dla formatu XML:

http://www.vizgr.org/historical-events/search.php?format=xml&begin_date=20110101&end_date=20151231&lang=en

Bonus: sporządź graficzną reprezentację, w jakich okresach było najwięcej zdarzeń. Np. pogrupuj dane po miesiącach.

Bonus2: Wydobądź wszystkie linki z opisów (wyrażeniem regularnym)

1.6 Pod adresem

<http://api.tvmaze.com/singlesearch/shows?q=mr-robot&embed=episodes> można uzyskać listę odcinków serialu Mr. Robot. Przetwórz dane tak, aby otrzymać ramkę danych, gdzie jeden wiersz to jeden odcinek (a kolumny to informacje o odcinku: nazwa, numer sezonu, numer odcinka, adres url, a także datę emisji odcinka (wraz z godziną)).

Bonus: <http://www.tvmaze.com/api> - możemy znaleźć inne informacje o serialach, możemy np. zwracać dzień, w którym dany serial jest emitowany.

1.7 Ściągnij z mojej strony zbiór danych siećKsięgarni.xml . Wczytaj go, a następnie:

- a) Wylistuj, pod jakimi adresami znajdują się księgarnie w tej sieci księgarni
- b) Jakie są dostępne wszystkie książki w tej sieci księgarni
- c) Jakie są dostępne wszystkie książki w tej sieci księgarni, które są z gatunku Fantasy

- d) Wylistuj w ładnej formie wszystkie książki w tej sieci księgarni, które są z gatunku Fantasy, wraz z adresami księgarni, gdzie można je nabyć
- e) Wylistuj w ładnej formie, obok siebie, adres księgarni i nazwisko jej właściciela
- f) Znajdź adres księgarni, gdzie właścicielem jest Matylda Jankowska,
- g) Jakie są dostępne wszystkie książki w księgarni przy ulicy 3 Maja, których autorem jest Jane Austen
- h) Jakie są książki z atrybutem liczba
- i) Książki, których liczba jest większa lub równa 3

1.8 Ściągnij z mojej strony zbiór danych, oznaczający oceny moich studentów z Algorytmów i Struktur Danych 2 z roku akademickiego 2015/2016. Każde zadanie składa się z dwóch części: część na zajęciach, za którą można dostać 4p (przeważnie), a także część domową, którą wysyła się do prowadzącego mailem z domu. Za nią można otrzymać 1p (przeważnie).

Zadania:

- a) Pamiętaj, aby nie było białych znaków przed lub po imieniu i nazwisku.
- b) Adresy e-mail powstają na naszym wydziale w następujący sposób: bierze się pełne nazwisko i na koniec dokleja (konkatenuje) pierwszą literę imienia. I tak Jan Kowalski ma e-mail kowalskij@student.mini.pw.edu.pl . Utwórz kolumnę "e-mail", która ma e-maile studentów. Pamiętaj, że w adresach nie może być polskich znaków.
- c) Utwórz string, który można wstawić bezpośrednio do klienta pocztowego, w którym są adresy e-mail studentów (czyli utwórz napis, gdzie wszystkie adresy są oddzielone przecinkiem).

- d) Wczytaj dane tak, aby część laboratoryjna była osobną kolumną, a część domowa osobną.
- e) Nieobecności usprawiedliwione zamienia się na punkty w następujący sposób: oblicza się średnią danego dnia, a także dla danego studenta (średnia w wierszu i kolumnie). Następnie oblicza się średnią z tych dwóch średnich. Pamiętaj, że nowo wyliczone wartości nie powinny być brane pod uwagę przy obliczaniu kolejnych zwolnień.
- f) Wystaw oceny końcowe (utwórz nową kolumnę) na podstawie sumy punktów (przedziały są procentowe):
 - [0%,50%) -> 2
 - [51%,60%) -> 3
 - [61%,70%) -> 3.5
 - [71%,80%) -> 4
 - [81%,90%) -> 4.5
 - [91%,Inf] -> 5

Maksymalną liczbę punktów do zdobycia na laboratoriach należy odczytać z nazw kolumn.

Bonus: narysuj wykresy:

- a) Boxplot rozkładu punktów dla danego zadania,
- b) Wykres liniowy punktów dla danego studenta na kolejnych laboratoriach (interesuje nas suma części laboratoryjnej i domowej)
- c) Boxplot sumy punktów za całe zajęcia
- d) Powtórz obliczenia, ale tym razem operuj na procentach (jeśli ktoś zdobył 2p. na 4p. możliwe, to otrzymał 50% punktów)

1.9 Na stronie

https://danepubliczne.gov.pl/dataset/wypadki_w_szkolach_i_placowkach_o_swiatowych/resource/3c77d0c7-fab7-40da-88d3-4890623304f9 można pobrać plik .xls dotyczący wypadków w szkołach.

Zadania:

- a) Wczytaj pierwszy arkusz przy użyciu wbudowanej funkcji w Pandas. Sprawdź, czy sumy "Razem" na pewno się zgadzają.
- b) Posortuj ramkę po obrażeniach: ciężkich, innych, śmiertelnych. W których szkołach najłatwiej zginąć? W których szkołach najłatwiej o ciężki wypadek? Czy pokrywa się to z doniesieniami z mediów? Jakiej informacji brakuje, aby ocenić te dane w sposób wiarygodny?
- c) Napisz funkcję, która dla zadanej nazwy szkoły, np. "Technikum" zwróci informację (napis), jakich obrażeń było najwięcej w danej szkole.
- d) Wczytaj drugi arkusz ("wg województw"). Wczytaj go w ten sposób, aby utworzyć indeks hierarchiczny, bazujący na nazwie województwa i typie szkoły. Usuń pierwszą kolumnę (02, 04, etc.). Wczytaj poprawnie kolumnę z kodami szkół (00001, 00003, etc.). Czemu nie mieliśmy z tym problemu poprzednio?
- e) Przekształć ramkę danych tak, aby mieć dane pogrupowane po typie szkoły, a dopiero potem po województwach. Operacja będzie polegać na zamianie indeksów i ponownym ich posortowaniu.
- f) Napisz funkcję gdzieNajłatwiej(typSzkoły, typWypadku), która zwróci informację, gdzie (w jakim województwie) jest najłatwiej o dany typ wypadku w danym typie szkoły.
- g) Sprawdź, czy dane z drugiego arkusza pokrywają się z tymi z pierwszego. Innymi szkołami, czy liczba wypadków każdego typu dla każdej szkoły jest taka sama.
- h) * Wczytaj arkusz "część ciała". Niech wczytana ramka danych ma multiindeksy zarówno na wierszach, jak i kolumnach. Następnie zamień multiindeksy, jak poprzednio (najpierw mamy typ szkoły, a potem województwo, tak samo najpierw mamy część ciała, a dopiero potem informację, jak poważny był to wypadek).

Bonus: Na podstawie tych danych narysuj:

- a) Łączną liczbę wypadków dla każdego województwa (zakładka wg województw), wykres słupkowy

- b) Liczba wypadków ze względu na miejsce, w którym doszło do wypadku (zakładka "miejsce"), wykres słupkowy
- c) Liczba uderzeń nieumyślnych w technikach w rozbiciu na województwa, wykres kołowy