

SPRAWOZDANIE

Zajęcia: Eksploracja i wizualizacja danych

Prowadzący: prof. dr hab. inż. Vasyl Martsenyuk

Laboratorium nr 1 Data: 26.09.2021 Temat: Ustalenia platformy Jupyter. Użycia biblioteki pandas w celu eksploracji i wizualizacji danych Wariant: 1	Piotr Rybka Informatyka II stopień, niestacjonarne, III semestr, gr. 1
--	---

Repozytorium z kodem programu:

https://github.com/prybka82-student/eksploracja_i_wizualizacja_danych

Polecenie

Zadanie dotyczy pobrania danych z pliku, tworzenia ramki danych, wykonania poszczególnych zadań na podstawie odpowiedniego zbioru danych:

<http://ghdx.healthdata.org/record/ihme-data/development-assistance-health-database-1990-2020>

Zadanie 1

Załadować bibliotekę `pandas`.

```
1 import pandas as pd
```

Zadanie 2

Utworzyć ramkę danych ze słownika.

```
1 dane1 = {  
2     "liczby": [532, 2532, 523, 2543, 235, 231, 34, 23552, 5324],  
3     "kategorie": ["a", "b", "c", "d", "e", "f", "g", "h", "i"]  
4 }  
5  
6 df1 = pd.DataFrame(dane1)  
7 df1
```

Zadanie 3

Zapisać ramkę danych w pliku `csv`.

```
1 sciezka = r"C:\Users\piotr\Downloads\dane_ze_slownika.csv"  
2  
3 df1.to_csv(sciezka, encoding="utf-8")
```

Zadanie 4

Utworzyć ramkę danych z listy list.

```
1 dane2 = [  
2     [532, 2532, 523, 2543, 235, 231, 34, 23552, 5324],  
3     ["a", "b", "c", "d", "e", "f", "g", "h", "i"]  
4 ]  
5  
6 df2 = pd.DataFrame(dane2)
```

Zadanie 5

Wczytać dane z pliku csv.

```
1 sciezka = r"C:\Users\piotr\Downloads\IHME_DAH_DATABASE_1990_2020_CSV_1\  
    IHME_DAH_DATABASE_1990_2020_Y2021M09D22.CSV"  
2  
3 df = pd.read_csv(sciezka, low_memory=False, )
```

Zadanie 6

Wyświetlić pierwsze 10 wierszy ramki danych.

```
1 df.head(10)
```

Zadanie 7

Wyświetlić ostatnie 10 wierszy ramki danych.

```
1 df.tail(10)
```

Zadanie 8

Wyświetlić informacje o ramce danych.

```
1 df.info()
```

Zadanie 9

Przetransponować dane w ramce danych.

```
1 df_transposed = df.T
2
3 df_transposed.head()
```

Zadanie 10

Wyświetlić liczbę wierszy i kolumn zawartych w ramce danych.

```
1 rows, cols = df.shape
2 print(f"Kolumn: {cols}")
3 print(f"Wierszy: {rows}")
```

Zadanie 11

Wyodrębnić wiersze i kolumny przy użyciu nazw i indeksów.

```
1 df["year"]
2
3 df.year
4
5 df[["year", "source", "recipient_country"]]
6
7 df.loc[:, "year":"channel"] # wszystkie wiersze, kolumny od "year" do "
   channel" włącznie
8
9 df.iloc[0:10, 2:4] # wiersze od pierwszego do dziesiątego, kolumny od
   trzeciej do czwartej
```

Zadanie 12

Wyświetlić podstawowe informacje statystyczne o kolumnach liczbowych (liczba wartości niepowtarzalnych, średnia, odchylenie standardowe, minimum, maksimum, wartości kwartyli).

```
1 liczbowe = df.select_dtypes(include='number')
2 liczbowe.describe()
```

Zadanie 13

Wyświetlić podstawowe informacje statystyczne o kolumnach kategorizowanych (liczba wartości niepowtarzalnych, wartość najczęstsza, liczba wystąpień wartości najczęstszej).

```
1 kategoryzowane = df.select_dtypes(exclude='number')
2 kategoryzowane.describe()
```

Zadanie 14

Usunięcie brakujących wartości z ramki danych.

```
1 df.dropna(inplace=True) # inplace=True - zmodyfikowane zostana oryginalne dane
```

Zadanie 15

Wybrać te wiersze ramki danych, które spełniają podany warunek dotyczący wybranej kolumny.

```
1 df[df["recipient_country"] == "Poland"]
```

Zadanie 16

Wybrać wiersze ramki danych, które spełniają kilka warunków jednocześnie.

```
1 df[(df["recipient_isocode"] == "POL") & (df["year"] >= 2005)]
```

Zadanie 17

Wybrać wiersze, które zawierają w kolumnie kategorizowanej określone słowo.

```
1 df[df["source"]=="Japan"]
```

Zadanie 18

Wybrać wiersze, które nie zawierają w kolumnie kategoryzowanej podanego wyrazu.

```
1 selection = df[df["gbd_region"]!="Asia, East"]
```

Zadanie 19

Utworzyć kolumnę na podstawie istniejącej kolumny w ramce danych.

```
1 recipient_countries = df.recipient_country
```

Zadanie 20

Usunąć kolumnę z ramki danych.

```
1 df_copy = df
2 df_copy.drop(["source", "channel"], axis=1, inplace=True) # usuwa kolumny "
    source" i "channel"
```

Zadanie 21

Zmienić nazwę kolumny w ramce danych.

```
1 df_copy.rename(columns={"year": "rok", "recipient_isocode": "
    kod_iso_odbiorcy", "recipient_country": "kraj_odbiorcy"}, inplace=True)
```

Zadanie 22

Zapisać ramkę danych jako plik w formacie csv.

```
1 sciezka = r"C:\Users\piotr\Downloads\df_copy.csv"
2 df.to_csv(sciezka, encoding="utf-8")
```

Zadanie 23

Wyświetlić średnią, maksymalną i minimalną wartość danej kolumny.

```
1 col = df["elim_ch"]
2 mean = col.mean()
3 max_ = col.max()
4 min_ = col.min()
5
6 print(f"srednia: {mean}\nmaksimum: {max_}\nminimum: {min_}")
```

Zadanie 24

Podać liczbę wierszy wybranej kolumny.

```
1 df.rok.count()
```

Zadanie 25

Wyodrębnić wartości unikatowe w kolumnie ramki danych.

```
1 df_copy["kraj_odbiorcy"].unique()
```

Zadanie 26

Wyświetlić liczbę rekordów spełniających podany warunek.

```
1 df_copy[df_copy["kraj_odbiorcy"]=="Turkey"].rok.count()
```

Zadanie 27

Posortować wiersze ramki danych według wartości w wybranej kolumnie (malejąco i rosnąco).

```
1 df_copy.sort_values(['kraj_odbiorcy'], ascending=True).head()
2
3 df_copy.sort_values(['kraj_odbiorcy'], ascending=False).head()
```

Zadanie 28

Wyświetlić wiersze zawierające 10 największych i najmniejszych wartości z wybranej kolumny.

```
1 df.nlargest(10, 'elim_ch')[['rok', 'elim_ch']]
2
3 df.nsmallest(10, 'elim_ch')[['rok', 'elim_ch']]
```

Zadanie 29

Wyświetlić wiersze zawierające 10 największych wartości wybranej kolumny pod warunkiem, że wartości w innej kolumnie mają określoną wartość.

```
1 df[(df['kraj_odbiorcy'].isin(['Poland', 'Germany'])) & (df['rok'] == 1990)
    ].nlargest(10, 'elim_ch')
```

Zadanie 30

Pogrupować wiersze według wartości kolumny kategoryzowanej, a następnie uśrednić wartości wszystkich kolumn liczbowych.

```
1 df.groupby('kraj_odbiorcy').agg('mean')
```

Zadanie 31

Pogrupować wiersze wg wartości kolumny kategoryzowanej, a następnie uśrednić wartości dla wybranych kolumn, a dla innych obliczyć częstość wystąpień i medianę.

```
1 kraje = df.groupby('kraj_odbiorcy').agg({'gbd_location_id': ['mean'], '
    elim_ch': ['mean'], 'prelim_est': ['mean'], 'rok': ['count', 'median'],
    'wb_location_id': ['count', 'median'], 'gbd_superregion_id': ['count', '
    median']})
```


Zadanie 32

Pobrać nazwy kolumn indeksu złożonego.

```
1 kraje.index
```

Zadanie 33

Posortować kolumnę indeksu złożonego.

```
1 kraje['elim_ch']['mean'].sort_values(ascending=False)
```

Zadanie 34

Stworzyć tabelę przestawną (ang. *pivot table*) na podstawie ramki danych.

```
1 pivot = df.pivot_table(values='elim_ch', index='kraj_odbiorcy', columns='rok',  
    aggfunc='count', margins=False, dropna=True, fill_value=None)
```

Zadanie 35

Wyświetlić indeksy i kolumny tabeli przestawnej.

```
1 pivot.index
```

Zadanie 36

Utworzyć indeks złożony tabeli przestawnej i wyświetlić jego zawartość.

```
1 pivot2 = df.pivot_table(values='elim_ch', index=['gbd_region', 'kraj_odbiorcy'], columns='rok',  
    aggfunc='count', margins=False, dropna=True, fill_value=None)
```

Zadanie 37

Zaimportować moduł pyplot z biblioteki matplotlib oraz wskazać, że wykresy należy rysować bezpośrednio w zeszycie, a nie w osobnej zakładce.

```
1 import matplotlib.pyplot as plt
2
3 %matplotlib inline
```

Zadanie 38

Wyświetlić wykres na podstawie tabeli przestawnej.

```
1 pivot3 = df.pivot_table(values='elim_ch', index='rok', columns='
    gbd_superregion', aggfunc='mean', margins=False, dropna=True, fill_value
    =None)
2
3 fig = pivot3.plot(kind='line')
4 plt.ylabel('mean value of elim_ch')
5 plt.title('Mean values of elim_ch in superregions per year')
6 plt.rcParams["figure.figsize"] = (20,10)
```

Zadanie 39

Narysować histogram na podstawie wartości w wybranej kolumnie.

```
1 df_bar = df.pivot_table(values='elim_ch', index='rok', columns='
    gbd_superregion', aggfunc='sum', margins=False, fill_value=None, dropna=
    True)
2 df_bar.plot(kind='bar')
3 plt.ylabel('elim_ch')
4 plt.title('elim_ch by year and superregion')
```

Zadanie 40

Przedstawić sposoby łączenia ramek danych za pomocą metod `merge` i `concat`.

```
1 part1 = df[['rok', 'kraj_odbiorcy', 'elim_ch', 'gbd_region']]
2 part2 = df[['rok', 'kraj_odbiorcy', 'prelim_est', 'gbd_superregion']]
3
4 pd.merge(part1, part2, on = ['rok', 'kraj_odbiorcy'], how='inner').head()
5
6 pd.merge(part1, part2, on = [part1.index, part2.index], how='inner').head()
7
8 pd.concat([part1, part2], axis=0)
9
10 pd.concat([part1, part2], axis=0).shape
11
12 pd.concat([part1, part2], axis=1)
13
14 pd.concat([part1, part2], axis=1).shape
```

Zadanie 41

Pokazać dodawanie nowych kolumn za pomocą operacji matematycznych.

```
1 df['sum'] = df['elim_ch'] + df['prelim_est']
```

Zadanie 42

Przedstawić przykładowe dodawanie nowych kolumn przy użyciu wyrażenia `lambda`.

```
1 df['years_ago'] = df['rok'].apply(lambda y: 2021 - int(y))
```

Zadanie 43

Ukazać możliwości pracy z dużymi plikami przy użyciu argumentu `chunksize`.

```
1 sciezka = r"C:\Users\piotr\Downloads\IHME_DAH_DATABASE_1990_2020_CSV_1\  
   IHME_DAH_DATABASE_1990_2020_Y2021M09D22.CSV"  
2  
3 chunks = pd.read_csv(sciezka, low_memory=False, chunksize=100_000)  
4  
5 for i, chunk in enumerate(chunks):  
6     print(f"\n\nChunk number {i+1}:")  
7     print(chunk.iloc[0:3,0:4])
```