

# Wizualizacja danych

## - wykład 7

dr Piotr Jastrzębski

# **Analiza danych - podstawowe pojęcia**

# Analiza danych - podstawowe pojęcia

Współczesne znaczenia słowa “statystyka”:

zbiór danych liczbowych pokazujący kształtowanie procesów i zjawisk np. statystyka ludności. wszelkie czynności związane z gromadzeniem i opracowywaniem danych liczbowych np. statystyka pewnego problemu dokonywana przez GUS. charakterystyki liczbowe np. statystyki próby np. średnia arytmetyczna, odchylenie standardowe itp. dyscyplina naukowa - nauka o metodach badania zjawisk masowych.

Zjawisko/procesy masowe - badaniu podlega duża liczba jednostek.

Dzieli się na:

- ▶ gospodarcze (np. produkcja, konsumpcja, usługi reklama),
- ▶ społeczne (np. wypadki drogowe, poglądy polityczne),
- ▶ demograficzne (np. urodzenia, starzenie, migracje).

## Statystyka - dyscyplina naukowa - podział:

- ▶ statystyka opisowa - zajmuje się sprawami związanymi z gromadzeniem, prezentacją, analizą i interpretacją danych liczbowych. Obserwacja obejmuje całą badaną zbiorowość.
- ▶ statystyka matematyczna - uogólnienie wyników badania części zbiorowości (próby) na całą zbiorowość.

Zbiorowość statystyczna, populacja statystyczna: zbiór obiektów podlegających badaniu statystycznemu. Tworzą je jednostki podobne do siebie, logicznie powiązane, lecz nie identyczne. Mają pewne cechy wspólne oraz pewne właściwości pozwalające je różnicować.

▶ przykłady:

- ▶ badanie wzrostu Polaków - mieszkańcy Polski
- ▶ poziom nauczania w szkołach woj. warmińsko-mazurskiego - szkoły woj. warmińsko-mazurskiego.

▶ podział:

- ▶ zbiorowość/populacja generalna - obejmuje całość,
- ▶ zbiorowość/populacja próbna (próba) - obejmuje część populacji.

Jednostka statystyczna: każdy z elementów zbiorowości statystycznej.

- ▶ przykłady:
  - ▶ studenci UWM - student UWM
  - ▶ mieszkańcy Polski - każda osoba mieszkająca w Polsce
  - ▶ maszyny produkowane w fabryce - każda maszyna

## Cechy statystyczne

- ▶ właściwości charakteryzujące jednostki statystyczne w danej zbiorowości statystycznej.
- ▶ dzielimy je na stałe i zmienne.



## Cechy stałe

- ▶ takie właściwości, które są wspólne wszystkim jednostkom danej zbiorowości statystycznej.
- ▶ podział:
  - ▶ rzeczowe - kto lub co jest przedmiotem badania statystycznego,
  - ▶ czasowe - kiedy zostało przeprowadzone badanie lub jakiego okresu czasu dotyczy badanie,
  - ▶ przestrzenne - jakiego terytorium (miejsce lub obszar) dotyczy badanie.
- ▶ przykład: studenci WMil UWM w Olsztynie w roku akad. 2017/2018:
  - ▶ cecha rzeczowa: posiadanie legitymacji studenckiej,
  - ▶ cecha czasowa - studenci studiujący w roku akad. 2017/2018
  - ▶ cecha przestrzenna - miejsce: WMil UWM w Olsztynie.

## Cechy zmienne

- ▶ właściwości różnicujące jednostki statystyczne w danej zbiorowości.
- ▶ przykład: studenci UWM - cechy zmienne: wiek, płeć, rodzaj ukończonej szkoły średniej, kolor oczu, wzrost.

### Ważne:

- ▶ obserwacji podlegają tylko cechy zmienne,
- ▶ cecha stała w jednej zbiorowości może być cechą zmienną w innej zbiorowości.

Przykład: studenci UWM mają legitymację wydaną przez UWM. Studenci wszystkich uczelni w Polsce mają legitymacje wydane przez różne szkoły.

## Podział cech zmiennych:

- ▶ cechy mierzalne (ilościowe) - można je wyrazić liczbą wraz z określoną jednostką miary.
- ▶ cechy niemierzalne (jakościowe) - określane słownie, reprezentują pewne kategorie.

Przykład: zbiorowość studentów. Cechy mierzalne: wiek, waga, wzrost, liczba nieobecności. Cechy niemierzalne: płeć, kolor oczu, kierunek studiów.

Często ze względów praktycznych cechom niemierzalnym przypisywane są kody liczbowe. Nie należy ich jednak mylić z cechami mierzalnymi. Np. 1 - wykształcenie podstawowe, 2 - wykształcenie zasadnicze, itd. . .

## Podział cech mierzalnych:

- ▶ ciągłe - mogące przybrać każdą wartość z określonego przedziału, np. wzrost, wiek, powierzchnia mieszkania.
- ▶ skokowe - mogące przyjmować konkretne (dyskretne) wartości liczbowe bez wartości pośrednich np. liczba osób w gospodarstwie domowych, liczba osób zatrudnionych w danej firmie.

Cechy skokowe zazwyczaj mają wartości całkowite choć nie zawsze jest to wymagane np. liczba etatów w firmie (z uwzględnieniem części etatów).

# Skale

## Skala pomiarowa

- ▶ to system, pozwalający w pewien sposób usystematyzować wyniki pomiarów statystycznych.
- ▶ podział:
  - ▶ skala nominalna,
  - ▶ skala porządkowa,
  - ▶ skala przedziałowa (interwałowa),
  - ▶ skala ilorazowa (stosunkowa).

## Skala nominalna

- ▶ skala, w której klasyfikujemy jednostkę statystyczną do określonej kategorii.
- ▶ wartość w tej skali nie ma żadnego uporządkowania.
- ▶ przykład:

Religia	Kod
Chrześcijaństwo	1
Islam	2
Buddyzm	3

## Skala porządkowa

- ▶ wartości mają jasno określony porządek, ale nie są dane odległości między nimi,
- ▶ pozwala na uszeregowanie elementów.
- ▶ przykłady:

Wykształcenie	Kod
Podstawowe	1
Średnie	2
Wyższe	3

Dochód	Kod
Niski	1
Średni	2
Wysoki	3

## Skala przedziałowa (interwałowa)

- ▶ wartości cechy wyrażone są poprzez konkretne wartości liczbowe,
- ▶ pozwala na porównywanie jednostek (coś jest większe lub mniejsze),
- ▶ nie możliwe jest badanie ilorazów (określenie ile razy dana wartość jest większa lub mniejsza od drugiej).
- ▶ przykład:

Miasto	Temperatura w $^{\circ}C$	Temperatura w $^{\circ}F$
Warszawa	15	59
Olsztyn	10	50
Gdańsk	5	41
Szczecin	20	68



## Skala ilorazowa (stosunkowa)

- ▶ wartości wyrażone są przez wartości liczbowe,
- ▶ możliwe określenie jest relacji mniejsza lub większa między wartościami,
- ▶ możliwe jest określenie stosunku (ilorazu) między wartościami,
- ▶ występuje zero absolutne.
- ▶ przykład:

Produkt	Cena w zł
Chleb	3
Masło	8
Gruszki	5

## Rodzaje badań statystycznych

- ▶ badanie pełne - obejmują wszystkie jednostki zbiorowości statystycznej.
  - ▶ spis statystyczny,
  - ▶ rejestracja bieżąca,
  - ▶ sprawozdawczość statystyczna.
- ▶ badania częściowe - obserwowana jest część populacji. Przeprowadza się wtedy gdy badanie pełne jest niecelowe lub niemożliwe.
  - ▶ metoda monograficzna,
  - ▶ metoda reprezentacyjna.

# Etapy badania statystycznego

- ▶ projektowanie i organizacja badania: ustalenie celu, podmiotu, przedmiotu, zakresu, źródła i czasu trwania badania;
- ▶ obserwacja statystyczna;
- ▶ opracowanie materiału statystycznego: kontrola materiału statystycznego, grupowanie uzyskanych danych, prezentacja wyników danych;
- ▶ analiza statystyczna.

# Analiza danych zastanych

Analiza danych zastanych – proces przetwarzania danych w celu uzyskania na ich podstawie użytecznych informacji i wniosków. W zależności od rodzaju danych i stawianych problemów, może to oznaczać użycie metod statystycznych, eksploracyjnych i innych.

Korzystanie z danych zastanych jest przykładem badań niereaktywnych - metod badań zachowań społecznych, które nie wpływają na te zachowania. Dane takie to: dokumenty, archiwa, sprawozdania, kroniki, spisy ludności, księgi parafialne, dzienniki, pamiętniki, blogi internetowe, audio-pamiętniki, archiwa historii mówionej i inne. (Wikipedia)

Dane zastane możemy podzielić ze względu na (Makowska red. 2013):

- ▶ Charakter: Ilościowe, Jakościowe
- ▶ Formę: Dane opracowane, Dane surowe
- ▶ Sposób powstania: Pierwotne, Wtórne
- ▶ Dynamikę: Ciągła rejestracja zdarzeń, Rejestracja w interwałach czasowych, Rejestracja jednorazowa
- ▶ Poziom obiektywizmu: Obiektywne, Subiektywne
- ▶ Źródła pochodzenia: Dane publiczne, Dane prywatne

Analiza danych to proces polegający na sprawdzaniu, porządkowaniu, przekształcaniu i modelowaniu danych w celu zdobycia użytecznych informacji, wypracowania wniosków i wspierania procesu decyzyjnego. Analiza danych ma wiele aspektów i podejść, obejmujących różne techniki pod różnymi nazwami, w różnych obszarach biznesowych, naukowych i społecznych. Praktyczne podejście do definiowania danych polega na tym, że dane to liczby, znaki, obrazy lub inne metody zapisu, w formie, którą można ocenić w celu określenia lub podjęcia decyzji o konkretnym działaniu. Wiele osób uważa, że dane same w sobie nie mają znaczenia – dopiero dane przetworzone i zinterpretowane stają się informacją.

# Proces analizy danych

Analiza odnosi się do rozbicia całości posiadanych informacji na jej odrębne komponenty w celu indywidualnego badania. Analiza danych to proces uzyskiwania nieprzetworzonych danych i przekształcania ich w informacje przydatne do podejmowania decyzji przez użytkowników. Dane są zbierane i analizowane, aby odpowiadać na pytania, testować hipotezy lub obalać teorie. Istnieje kilka faz, które można wyszczególnić w procesie analizy danych. Fazy są iteracyjne, ponieważ informacje zwrotne z faz kolejnych mogą spowodować dodatkową pracę w fazach wcześniejszych.

# Zdefiniowanie wymagań

Przed przystąpieniem do analizy danych, należy dokładnie określić wymagania jakościowe dotyczące danych. Dane wejściowe, które mają być przedmiotem analizy, są określone na podstawie wymagań osób kierujących analizą lub klientów (którzy będą używać finalnego produktu analizy). Ogólny typ jednostki, na podstawie której dane będą zbierane, jest określany jako jednostka eksperymentalna (np. osoba lub populacja ludzi). Dane mogą być liczbowe lub kategoriowe (tj. Etykiety tekstowe). Faza definiowania wymagań powinna dać odpowiedź na 2 zasadnicze pytania:

- ▶ co chcemy zmierzyć?
- ▶ w jaki sposób chcemy to zmierzyć?



# Gromadzenie danych

Dane są gromadzone z różnych źródeł. Wymogi, co do rodzaju i jakości danych mogą być przekazywane przez analityków do “opiekunów danych”, takich jak personel technologii informacyjnych w organizacji. Dane ponadto mogą być również gromadzone automatycznie z różnego rodzaju czujników znajdujących się w otoczeniu - takich jak kamery drogowe, satelity, urządzenia rejestrujące obraz, dźwięk oraz parametry fizyczne. Kolejną metodą jest również pozyskiwanie danych w drodze wywiadów, gromadzenie ze źródeł internetowych lub bezpośrednio z dokumentacji.

# Przetwarzanie danych

Zgromadzone dane muszą zostać przetworzone lub zorganizowane w sposób logiczny do analizy. Na przykład, mogą one zostać umieszczone w tabelach w celu dalszej analizy - w arkuszu kalkulacyjnym lub innym oprogramowaniu. Oczyszczanie danych Po fazie przetworzenia i uporządkowania, dane mogą być niekompletne, zawierać duplikaty lub zawierać błędy. Konieczność czyszczenia danych wynika z problemów związanych z wprowadzaniem i przechowywaniem danych. Czyszczenie danych to proces zapobiegania powstawaniu i korygowania wykrytych błędów. Typowe zadania obejmują dopasowywanie rekordów, identyfikowanie nieścisłości, ogólny przegląd jakości istniejących danych, usuwanie duplikatów i segmentację kolumn. Niezwykle istotne jest też zwracanie uwagi na dane których wartości są powyżej lub poniżej ustalonych wcześniej progów (ekstrema).

# Właściwa analiza danych

Istnieje kilka metod, które można wykorzystać do tego celu, na przykład data mining, business intelligence, wizualizacja danych lub badania eksploracyjne. Ta ostatnia metoda jest sposobem analizowania zbiorów informacji w celu określenia ich odrębnych cech. W ten sposób dane mogą zostać wykorzystane do przetestowania pierwotnej hipotezy. Statystyki opisowe to kolejna metoda analizy zebranych informacji. Dane są badane, aby znaleźć najważniejsze ich cechy. W statystykach opisowych analitycy używają kilku podstawowych narzędzi - można użyć średniej lub średniej z zestawu liczb. Pomaga to określić ogólny trend aczkolwiek nie zapewnia to dużej dokładności przy ocenie ogólnego obrazu zebranych danych. W tej fazie ma miejsce również modelowanie i tworzenie formuł matematycznych - stosowane są w celu identyfikacji zależności między zmiennymi, takich jak korelacja lub przyczynowość.

## Raportowanie i dystrybucja wyników

Ta faza polega na ustalaniu w jakiej formie przekazywać wyniki. Analityk może rozważyć różne techniki wizualizacji danych, aby w sposób wyraźnym i skuteczny przekazać wnioski z analizy odbiorcom. Wizualizacja danych wykorzystuje formy graficzne jak wykresy i tabele. Tabele są przydatne dla użytkownika, który może wyszukiwać konkretne rekordy, podczas gdy wykresy (np. wykresy słupkowe lub liniowe) dają spojrzenie ilościowych na zbiór analizowanych danych.

# Skąd brać dane?

Darmowa repozytoria danych:

- ▶ Bank danych lokalnych GUS - link
- ▶ Otwarte dane - link
- ▶ Bank Światowy - link

Przydatne strony:

- ▶ <https://astrafox.pl/10-darmowych-baz-danych-na-wyciagniecie-reki/>
- ▶ <https://www.nature.com/sdata/policies/repositories>
- ▶ <https://medium.freecodecamp.org/https-medium-freecodecamp-org-best-free-open-data-sources-anyone-can-use-a65b514b0f2d>

**“Tidy data”**

# Koncepcja

Koncepcja czyszczenia danych (ang. tidy data):

- ▶ WICKHAM, Hadley . Tidy Data. Journal of Statistical Software, [S.l.], v. 59, Issue 10, p. 1 - 23, sep. 2014. ISSN 1548-7660. Available at: <https://www.jstatsoft.org/v059/i10>. Date accessed: 25 oct. 2018. doi:<http://dx.doi.org/10.18637/jss.v059.i10>.

# Zasady “czystych danych”

Idealne dane są zaprezentowane w tabeli:

Imię	Wiek	Wzrost	Kolor oczu
Adam	26	167	Brązowe
Sylwia	34	164	Piwnie
Tomasz	42	183	Niebieskie

Na co powinniśmy zwrócić uwagę?

- ▶ jedna obserwacja (jednostka statystyczna) = jeden wiersz w tabeli/macierzy/ramce danych
- ▶ wartości danej cechy znajdują się w kolumnach
- ▶ jeden typ/rodzaj obserwacji w jednej tabeli/macierzy/ramce danych



## Przykłady nieuporządkowanych danych

Imię	Wiek	Wzrost	Brązowe	Niebieskie	Piwne
Adam	26	167	1	0	0
Sylwia	34	164	0	0	1
Tomasz	42	183	0	1	0

**Nagłówki kolumn muszą odpowiadać cechom, a nie wartościom zmiennych.**

# Kod do analizy

https:

[//gist.github.com/pjastr/309281eedf2ca5d0425b26e8d12eaa6f](https://gist.github.com/pjastr/309281eedf2ca5d0425b26e8d12eaa6f)

# **Biblioteka Pandas**

# Biblioteca Pandas

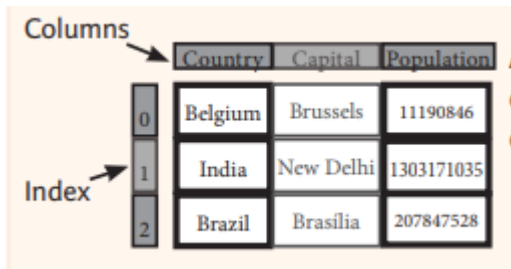
Import:

```
import pandas as pd
```

Seria - Series

a	3
b	-5
c	7
d	4

## Ramka danych - DataFrame



The diagram illustrates a DataFrame structure. It features a grid of data with three columns and three rows. The columns are labeled 'Country', 'Capital', and 'Population'. The rows are indexed from 0 to 2. Arrows point from the labels 'Columns' and 'Index' to their respective parts of the table.

	Country	Capital	Population
0	Belgium	Brussels	11190846
1	India	New Delhi	1303171035
2	Brazil	Brasilia	207847528

```
import pandas as pd
import numpy as np

s = pd.Series([3, -5, 7, 4])
print(s)
```

```
## 0    3
## 1   -5
## 2    7
## 3    4
## dtype: int64
```

```
print(s.values)
```

```
## [ 3 -5  7  4]
```

```
print(type(s.values))
```

```
## <class 'numpy.ndarray'>
```

```
t = np.sort(s.values)
```

```
print(t)
```

```
## [-5  3  4  7]
```



```
print(s.index)
```

```
## RangeIndex(start=0, stop=4, step=1)
```

```
print(type(s.index))
```

```
## <class 'pandas.core.indexes.range.RangeIndex'>
```

```
s = pd.Series([3, -5, 7, 4], index=['a', 'b', 'c', 'd'])  
print(s)
```

```
## a    3
```

```
## b   -5
```

```
## c    7
```

```
## d    4
```

```
## dtype: int64
```

```
print(s['b'])
```

```
## -5
```

```
s['b'] = 8  
print(s)
```

```
## a    3
```

```
## b    8
```

```
## c    7
```

```
## d    4
```

```
## dtype: int64
```

```
print(s[s>5])
```

```
## b      8
```

```
## c      7
```

```
## dtype: int64
```

```
print(s*2)
```

```
## a      6  
## b     16  
## c     14  
## d      8  
## dtype: int64
```

```
print(np.sin(s))
```

```
## a      0.141120  
## b      0.989358  
## c      0.656987  
## d     -0.756802  
## dtype: float64
```

```
d = {'key1': 350, 'key2': 700, 'key3': 70}
s = pd.Series(d)
print(s)
```

```
## key1      350
## key2      700
## key3       70
## dtype: int64
```

```
d = {'key1': 350, 'key2': 700, 'key3': 70}
k = ['key0', 'key2', 'key3', 'key1']
s = pd.Series(d, index=k)
print(s)
```

```
## key0      NaN
## key2      700.0
## key3       70.0
## key1      350.0
## dtype: float64
```

```
pd.isnull(s)
```

```
## key0      True  
## key2      False  
## key3      False  
## key1      False  
## dtype: bool
```

```
pd.notnull(s)
```

```
## key0      False  
## key2      True  
## key3      True  
## key1      True  
## dtype: bool
```



```
s.isnull()
```

```
## key0      True  
## key2      False  
## key3      False  
## key1      False  
## dtype: bool
```

```
s.notnull()
```

```
## key0      False  
## key2      True  
## key3      True  
## key1      True  
## dtype: bool
```

```
s.name = "Wartość"  
s.index.name = "Klucz"  
print(s)
```

```
## Klucz  
## key0      NaN  
## key2      700.0  
## key3      70.0  
## key1      350.0  
## Name: Wartość, dtype: float64
```

```
data = {'Country': ['Belgium', 'India', 'Brazil'],
        'Capital': ['Brussels', 'New Delhi', 'Brasília'],
        'Population': [11190846, 1303171035, 207847528]}
frame = pd.DataFrame(data)
print(frame)
```

```
##      Country      Capital  Population
## 0  Belgium  Brussels    11190846
## 1   India  New Delhi   1303171035
## 2  Brazil  Brasília    207847528
```

```
df = pd.DataFrame(data, columns=['Country', 'Capital',  
                                'Population'])  
print(df)
```

```
##      Country      Capital  Population  
## 0  Belgium  Brussels    11190846  
## 1    India  New Delhi    1303171035  
## 2   Brazil  Brasília    207847528
```

```
print(df.iloc[[0],[0]])
```

```
##      Country  
## 0  Belgium
```

```
print(df.loc[[0], ['Country']])
```

```
##      Country  
## 0  Belgium
```

```
print(df.loc[2])
```

```
## Country          Brazil
## Capital          Brasília
## Population       207847528
## Name: 2, dtype: object
```

```
print(df.loc[:, 'Capital'])
```

```
## 0    Brussels
## 1    New Delhi
## 2    Brasília
## Name: Capital, dtype: object
```

```
print(df.loc[1, 'Capital'])
```

```
## New Delhi
```

```
print(df[df['Population']>1200000000])
```

```
##      Country      Capital  Population  
## 1      India  New Delhi  1303171035
```

```
print(df.drop('Country', axis=1))
```

```
##      Capital  Population
## 0  Brussels   11190846
## 1  New Delhi  1303171035
## 2  Brasília   207847528
```



```
print(df.shape)
```

```
## (3, 3)
```

```
print(df.index)
```

```
## RangeIndex(start=0, stop=3, step=1)
```

```
print(df.columns)
```

```
## Index(['Country', 'Capital', 'Population'], dtype='object')
```

```
print(df.info())
```

```
## <class 'pandas.core.frame.DataFrame'>  
## RangeIndex: 3 entries, 0 to 2  
## Data columns (total 3 columns):  
## Country      3 non-null object  
## Capital      3 non-null object  
## Population    3 non-null int64  
## dtypes: int64(1), object(2)  
## memory usage: 200.0+ bytes  
## None
```

```
print(df.count())
```

```
## Country      3  
## Capital      3  
## Population    3  
## dtype: int64
```

## Uzupełnianie braków

```
s = pd.Series([3, -5, 7, 4], index=['a', 'b', 'c', 'd'])
s2 = pd.Series([7, -2, 3], index=['a', 'c', 'd'])
print(s+s2)
```

```
## a      10.0
## b       NaN
## c       5.0
## d       7.0
## dtype: float64
```

```
print(s.add(s2, fill_value=0))
```

```
## a    10.0  
## b    -5.0  
## c     5.0  
## d     7.0  
## dtype: float64
```

```
print(s.mul(s2, fill_value=2))
```

```
## a    21.0  
## b   -10.0  
## c   -14.0  
## d    12.0  
## dtype: float64
```

# Obsługa plików csv

Funkcja `pandas.read_csv`

Dokumentacja: [link](#)

Zapis `pandas.DataFrame.to_csv`

Dokumentacja: [link](#)

Grey	Green	Orange	Blue
Grey	Light Green	Light Orange	Light Blue
Grey	Light Green	Light Orange	Light Blue



Grey	Grey	Grey
Grey	Green	Green
Grey	Green	Green
Grey	Orange	Orange
Grey	Orange	Orange
Grey	Blue	Blue
Grey	Blue	Blue

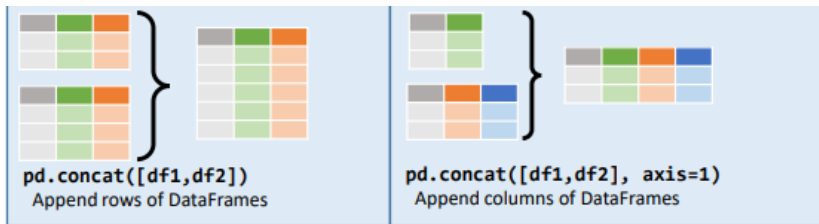
**pd.melt(df)**

Gather columns into rows.



```
df.pivot(columns='var', values='val')
```

Spread rows into columns.



[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/merging.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html)



## “Tidy data”

Imię	Wiek	Wzrost	Kolor oczu
Adam	26	167	Brązowe
Sylwia	34	164	Piwnie
Tomasz	42	183	Niebieskie

- ▶ jedna obserwacja (jednostka statystyczna) = jeden wiersz w tabeli/macierzy/ramce danych
- ▶ wartości danej cechy znajdują się w kolumnach
- ▶ jeden typ/rodzaj obserwacji w jednej tabeli/macierzy/ramce danych

## Obsługa brakujących danych

```
string_data = pd.Series(['aardvark', 'artichoke', np.nan,  
print(string_data)
```

```
## 0    aardvark  
## 1    artichoke  
## 2         NaN  
## 3    avocado  
## dtype: object
```

```
print(string_data.isnull())
```

```
## 0    False  
## 1    False  
## 2     True  
## 3    False  
## dtype: bool
```

```
print(string_data.dropna())
```

```
## 0      aardvark
```

```
## 1      artichoke
```

```
## 3      avocado
```

```
## dtype: object
```

```
from numpy import nan as NA
data = pd.DataFrame([[1., 6.5, 3.], [1., NA, NA],
                    [NA, NA, NA], [NA, 6.5, 3.]])
cleaned = data.dropna()
print(cleaned)
```

```
##      0      1      2
## 0  1.0  6.5  3.0
```

```
print(data.dropna(how='all'))
```

```
##      0      1      2
## 0  1.0  6.5  3.0
## 1  1.0  NaN  NaN
## 3  NaN  6.5  3.0
```

```
data[4] = NA
```

```
print(data.dropna(how='all', axis=1))
```

```
##      0      1      2
## 0  1.0  6.5  3.0
## 1  1.0  NaN  NaN
## 2  NaN  NaN  NaN
## 3  NaN  6.5  3.0
```

## Uzupełnienie braków

```
print(data)
```

```
##      0      1      2      4  
## 0  1.0  6.5  3.0  NaN  
## 1  1.0  NaN  NaN  NaN  
## 2  NaN  NaN  NaN  NaN  
## 3  NaN  6.5  3.0  NaN
```

```
print(data.fillna(0))
```

```
##      0      1      2      4  
## 0  1.0  6.5  3.0  0.0  
## 1  1.0  0.0  0.0  0.0  
## 2  0.0  0.0  0.0  0.0  
## 3  0.0  6.5  3.0  0.0
```

```
print(data.fillna({1: 0.5, 2: 0}))
```

```
##      0      1      2      4  
## 0  1.0  6.5  3.0 NaN  
## 1  1.0  0.5  0.0 NaN  
## 2  NaN  0.5  0.0 NaN  
## 3  NaN  6.5  3.0 NaN
```

## Usuwanie duplikatów

```
data = pd.DataFrame({'k1': ['one', 'two'] * 3 + ['two'],  
                    'k2': [1, 1, 2, 3, 3, 4, 4]})  
print(data)
```

```
##      k1  k2  
## 0  one   1  
## 1  two   1  
## 2  one   2  
## 3  two   3  
## 4  one   3  
## 5  two   4  
## 6  two   4
```



```
print(data.duplicated())
```

```
## 0    False
```

```
## 1    False
```

```
## 2    False
```

```
## 3    False
```

```
## 4    False
```

```
## 5    False
```

```
## 6     True
```

```
## dtype: bool
```

```
print(data.drop_duplicates())
```

```
##      k1  k2  
## 0  one   1  
## 1  two   1  
## 2  one   2  
## 3  two   3  
## 4  one   3  
## 5  two   4
```

## Zastępowanie wartościami

```
data = pd.Series([1., -999., 2., -999., -1000., 3.])  
print(data)
```

```
## 0      1.0  
## 1    -999.0  
## 2      2.0  
## 3    -999.0  
## 4   -1000.0  
## 5      3.0  
## dtype: float64
```

```
print(data.replace(-999, np.nan))
```

```
## 0      1.0  
## 1      NaN  
## 2      2.0  
## 3      NaN  
## 4    -1000.0  
## 5       3.0  
## dtype: float64
```

```
print(data.replace([-999, -1000], np.nan))
```

```
## 0    1.0
```

```
## 1    NaN
```

```
## 2    2.0
```

```
## 3    NaN
```

```
## 4    NaN
```

```
## 5    3.0
```

```
## dtype: float64
```

```
print(data.replace([-999, -1000], [np.nan, 0]))
```

```
## 0    1.0
```

```
## 1    NaN
```

```
## 2    2.0
```

```
## 3    NaN
```

```
## 4    0.0
```

```
## 5    3.0
```

```
## dtype: float64
```

```
print(data.replace({-999: np.nan, -1000: 0}))
```

```
## 0    1.0
```

```
## 1    NaN
```

```
## 2    2.0
```

```
## 3    NaN
```

```
## 4    0.0
```

```
## 5    3.0
```

```
## dtype: float64
```

## Dyskretyzacja i podział na koszyki

```
ages = [20, 22, 25, 27, 21, 23, 37, 31, 61, 45, 41, 32]
bins = [18, 25, 35, 60, 100]
cats = pd.cut(ages, bins)
print(cats)
```

```
## [(18, 25], (18, 25], (18, 25], (25, 35], (18, 25], ...,
## Length: 12
## Categories (4, interval[int64]): [(18, 25] < (25, 35] <
print(cats.codes)
```

```
## [0 0 0 1 0 0 2 1 3 2 2 1]
```



```
print(cats.categories)
```

```
## IntervalIndex([(18, 25], (25, 35], (35, 60], (60, 100]),  
##                closed='right',  
##                dtype='interval[int64]')
```

```
print(pd.value_counts(cats))
```

```
## (18, 25]      5  
## (35, 60]      3  
## (25, 35]      3  
## (60, 100]     1  
## dtype: int64
```

```
cats2 = pd.cut(ages, [18, 26, 36, 61, 100], right=False)
print(cats2)
```

```
## [[18, 26), [18, 26), [18, 26), [26, 36), [18, 26), ...,
## Length: 12
## Categories (4, interval[int64]): [[18, 26) < [26, 36) <
```

```
group_names = ['Youth', 'YoungAdult',
               'MiddleAged', 'Senior']
```

```
print(pd.cut(ages, bins, labels=group_names))
```

```
## [Youth, Youth, Youth, YoungAdult, Youth, ..., YoungAdult
## Length: 12
## Categories (4, object): [Youth < YoungAdult < MiddleAged
```

```
data = np.random.rand(20)
print(pd.cut(data, 4, precision=2))
```

```
## [(0.11, 0.32], (0.11, 0.32], (0.52, 0.72], (0.72, 0.92],
```

```
## Length: 20
```

```
## Categories (4, interval[float64]): [(0.11, 0.32] < (0.32,
```

```
data = np.random.randn(1000)
cats = pd.qcut(data, 4)
print(cats)
```

```
## [(0.663, 3.443], (-0.696, 0.0127], (-0.696, 0.0127], (-3.489, -0.696]]
## Length: 1000
## Categories (4, interval[float64]): [(-3.489, -0.696] < (0.663, 3.443]]
##
```

```
print(pd.value_counts(cats))
```

```
## (0.663, 3.443]      250
## (0.0127, 0.663]    250
## (-0.696, 0.0127]  250
## (-3.489, -0.696]  250
## dtype: int64
```

## Wykrywanie i filtrowanie elementów odstających

```
data = pd.DataFrame(np.random.randn(1000, 4))  
print(data.describe())
```

##	0	1	2	3
## count	1000.000000	1000.000000	1000.000000	1000.000000
## mean	0.002371	-0.032119	0.054141	0.008000
## std	0.996547	0.991181	1.014221	0.969930
## min	-3.402442	-3.127598	-3.095463	-3.216030
## 25%	-0.696731	-0.683101	-0.609631	-0.656510
## 50%	0.004353	-0.019731	0.017842	0.002470
## 75%	0.680981	0.662993	0.770881	0.663240
## max	2.885256	3.341639	3.082286	3.346880

```
col = data[2]
print(col[np.abs(col) > 3])
```

```
## 30      -3.095463
## 103     -3.037219
## 378      3.082286
## 569      3.056987
## 639      3.061522
## 963     -3.092825
## Name: 2, dtype: float64
```

```
print(data[(np.abs(data) > 3).any(1)])
```

```
##           0           1           2           3
## 21  -0.005696  3.341639  1.014961  1.777057
## 30  -0.992754 -0.631915 -3.095463 -0.642610
## 103 -2.250201  0.897881 -3.037219  1.839976
## 122 -2.316555  0.855220  0.176410  3.346889
## 144 -0.826770  3.024839  0.713647 -0.164990
## 181 -3.208942  1.189631 -0.248073 -0.706450
## 378 -0.497194  0.462764  3.082286  1.325432
## 460 -0.000615 -3.127598  0.990078  1.204609
## 569 -2.148141  0.630542  3.056987 -0.105035
## 639 -1.108911 -1.456028  3.061522 -0.044348
## 692  0.838494  0.250162 -1.298824 -3.216035
## 963 -1.365783  0.518457 -3.092825  0.205347
## 972 -3.402442 -1.005381 -2.077324  1.040049
```

# Bibliografia

- ▶ [https://mfiles.pl/pl/index.php/Analiza\\_danych](https://mfiles.pl/pl/index.php/Analiza_danych), dostęp online 1.04.2019.
- ▶ Walesiak M., Gatnar E., Statystyczna analiza danych z wykorzystaniem programu R, PWN, Warszawa, 2009.
- ▶ Wasilewska E., Statystyka opisowa od podstaw, Podręcznik z zadaniami, Wydawnictwo SGGW, Warszawa, 2009.
- ▶ [https://s3.amazonaws.com/assets.datacamp.com/blog\\_assets/PandasPythonForDataScience.pdf](https://s3.amazonaws.com/assets.datacamp.com/blog_assets/PandasPythonForDataScience.pdf), dostęp online 5.4.2019.
- ▶ <https://www.marsja.se/pandas-read-csv-tutorial-to-csv/>, dostęp online 20.04.2019.
- ▶ <https://www.geeksforgeeks.org/python-pandas-melt/>
- ▶ [https://pandas.pydata.org/Pandas\\_Cheat\\_Sheet.pdf](https://pandas.pydata.org/Pandas_Cheat_Sheet.pdf)