Języki programowania – Python
Eksploracja danych

Marzena Nowakowska

Wydział Zarządzania i Modelowania Komputerowego
Politechnika Świętokrzyska

Budynek C, p. 3.21

spimn@tu.kielce.pl

Materiały pomocnicze dla studentów: https://staff.tu.kielce.pl/spimn/pjp/ → Python – Pandas cheat sheet

Data Wrangling with Pandas. CheatSheet - Przetwarzanie danych za pomocą Pandas. Ściągawka

Syntax – Creating DataFrames	Składnia – tworzenie ramki danych
Subset Observations (Rows)	Podzbiór obserwacji, wybór wierszy
Subset Variables (Columns)	Podzbiór zmiennych, wybór kolumn
Summarize Data	Podsumowanie danych
Group Data	Grupowane danych
Make New Columns	Tworzenie nowych kolumn
Plotting	Tworzenie wykresów

Podstawowe statystyki w *pandas*

Miary statystyczne dla szeregu statystycznego otrzymuje się poparz wywołanie właściwych metod na rzecz obiektu *Series* lub wybranych kolumn *DataFrame (kolumna jest typu Series).* Wywołanie metody na rzecz całej ramki danych generuje miary statystyczne dla wszystkich kolumn numerycznych. Metody ignorują wartości puste (*None*).

Metoda	**Zwracana wartość**
count()	liczba obserwacji
sum()	suma wartości szeregu
mean()	średnia wartość w szeregu
median()	mediana szeregu
std()	odchylenie standardowe
var()	wariancja
min()	minimalna wartość
max()	maksymalna wartość
quantile()	kwantyl szeregu
mode()	modalna (dominanata) wartości
skew()	skośność
kurt()	kurtoza (spłaszczenie)
nunique()	liczba wartości unikatowych
describe()	zbiór statystyk: liczebność szeregu, średnia, odchylenie standardowe, minimum, Q1, Q2 (mediana), Q3, maksimum

Przykłady obliczeń statystyk w *DataFrame*

```
import pandas as pd
df = pd.read_csv("titanic.csv")
# Usunięcie nieanalizowanych kolumn;
df1=df.drop(['sibsp','ticket','cabin','boat', 'body', 'parch'], axis=1)
sttNum=df1.describe()
df1["age"].describe() ;   df1[["age", "fare"]].describe()     # dla wybranych kolumn
df1[["age", "fare"]].(["count","sum", "mean", "min", "max", "median"])  # wybrane miary
df1[df1["fare"]>0]["fare"].describe()      # dla wybranych wierszy ; średnia cen niezerowych
# Wyznaczenie miar dla zmiennych jakościowych; top jest dominantą (modalną,
# występującą najczęściej), freq jest częstością dominanty (ostatniej dla cechy
# welomodalnej )
sttJkosc=df1[[ 'sex', 'embarked', 'home_dest']].describe()
# Wybrane statystyki wg grup
MeanAgeBySex=df1.groupby(["sex"])["age"].mean()
Q25ByClass=df1[df1["fare"]>0].groupby(["pclass"])["fare"].describe(percentiles=[0.30])
SttBySexClass=df1.groupby(["sex", "pclass"])["age"].describe()
SurvByClass=df1.groupby(["pclass", "survived"])["survived"].count()
SurvBySex=df1.groupby(["sex", "survived"])["survived"].count()
```

Wybrane operacje na danych z *DataFrame* w *pandas*

DataFrame.duplicated(*subset=None, keep='first'*) zwraca serię danych z wartościami logicznymi (*True*, *False*) dla każdego wiersza informując, czy wiersz jest duplikatem. Domyślnie dla każdego zestawu zduplikowanych wartości pierwsze wystąpienie jest ustawiane na *False*, a wszystkie pozostałe na *True*.

DataFrame.drop(*labels=None, axis=0*) usunięcie wskazanych w *labels* wierszy lub kolumn, *axis* informuje której współrzędnej dotyczy usunięcie; *axis* = 0 dla wierszy, *axis* = 1 dla kolumn. Metoda zwraca ramkę danych niezawierającą usuniętych kolekcji danych (kolumn lub wierszy).

DataFrame.isnull() zwraca ramkę danych o tym samym rozmiarze z wartościami *True* lub *False* w miejscu elementów badanej ramki, które są odpowiednio: puste (brak wartości) lub niepuste (wartość obecna).

DataFrame.sum(*axis=0*) tworzy serię danych zawierającą sumy wartości elementów ramki wg wskazanego wymiaru:

axis = 0 (lub *'index'*) reprezentuje kolumny; suma wartości dla kolejnych kolumn

axis = 1 (lub *'columns'*) reprezentuje wiersze; suma wartości dla kolejnych wierszy

Wizualizacja danych w *pandas*

DataFrame.plot(kind = "line", x = None, y = None, figsize = None, title = None, ...)
Series.plot(kind = "line", figsize = None, title = None, ...)

Najważniejsze parametry:

kind – rodzaj wykresu
x, y – kolumny określające zmienne dla osi odpowiednio poziomej i pionowej
figsize – rozmiar
title – tytuł

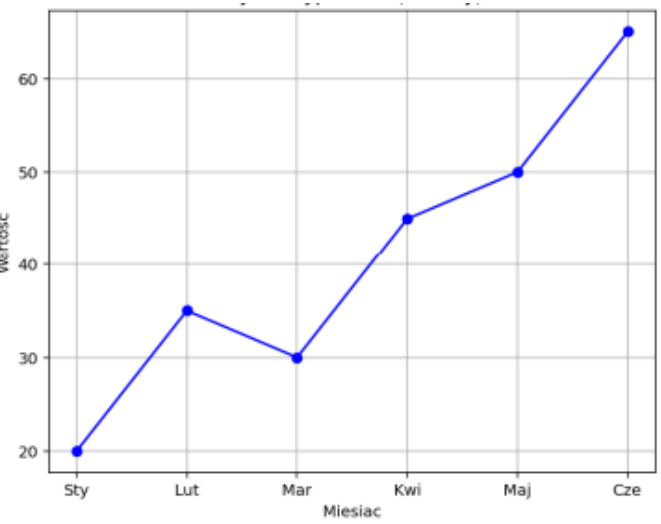
Inne metody wywoływane na rzecz *Series* (nazwy funkcji zaznaczone * w tabeli):

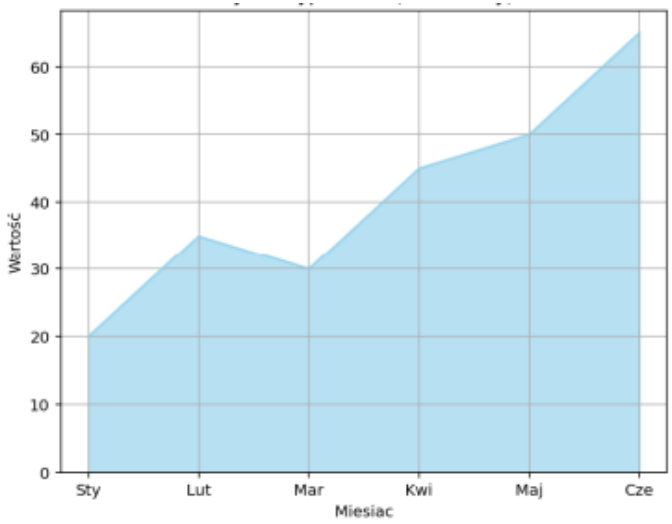
Wartość parametru *kind*	**Series**	**Opis**
line	*	Wykres liniowy (wartość domyślna)
area	*	Wykres obszarowy
pie	*	Wykres kołowy
bar	*	Wykres słupkowy pionowy
barh	*	Wykres słupkowy poziomy
hist	*	Histogram
box	*	Wykres pudełkowy
scatter		Wykres punktowy

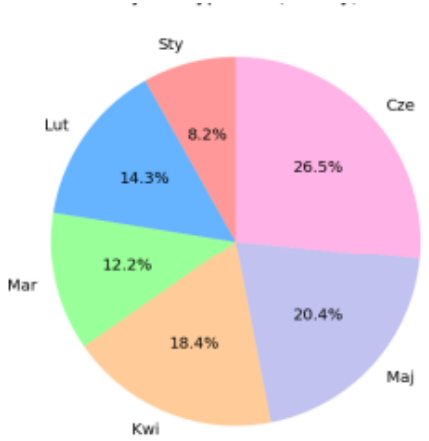
Wybrane wykresy

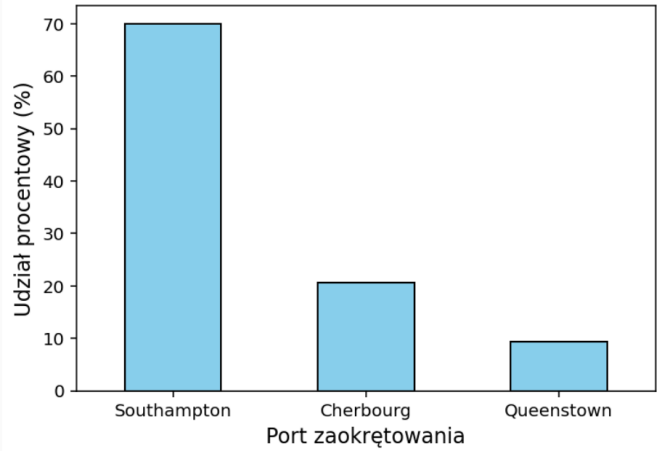
```
# dane do wykresów
df1=df.drop(['sibsp','ticket','cabin','boat', 'body', 'parch'], axis=1)
Surv=df1.groupby("survived")["survived"].count()
SurvByClass=df1.groupby(["pclass", "survived"])["survived"].count()
SurvBySex=df1.groupby(["sex", "survived"])["survived"].count()
#wykresy
df1.plot(kind = 'scatter', y = 'age', x = 'fare')          # wykres rozrzutu
df1.plot(kind = 'hist', y = 'fare')                        # histogram
df1.boxplot(column=['age'], by='sex')                      # wykresy pudełko i wąsy
Sex=df1.groupby("sex")["survived"].count()
Sex.plot.barh(); Surv.plot.barh()                          #wykresy słupkowe poziome

import seaborn as sb  # może być potrzebna instalacja: pip install seaborn
sb.violinplot(x=df1["age"])                                # wykres wiolinowy
sb.violinplot(data=df1, x="age", y='sex')
sb.violinplot(data=df1, x="pclass", y='age', hue='survived')
df1.hist()                                                 #histogram
df1["age"].hist()
df1.hist(column="age", by='sex', bins=10)
df1.groupby("pclass")["pclass"].count().plot.pie(figsize=(10,10), colors=["red", "green",
"blue"])                                                   # wykres  kołowy
```

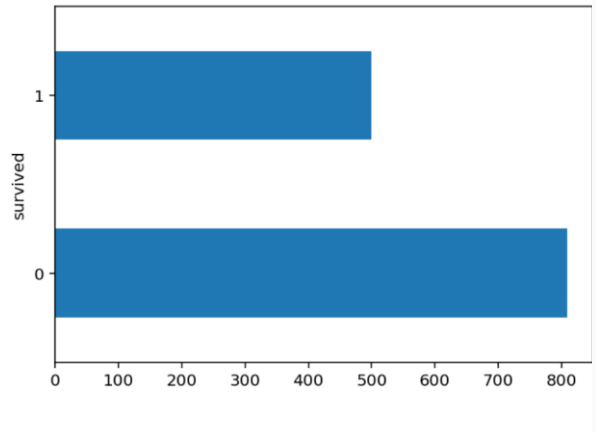
Rodzaje wykresów 1 - ilustracje

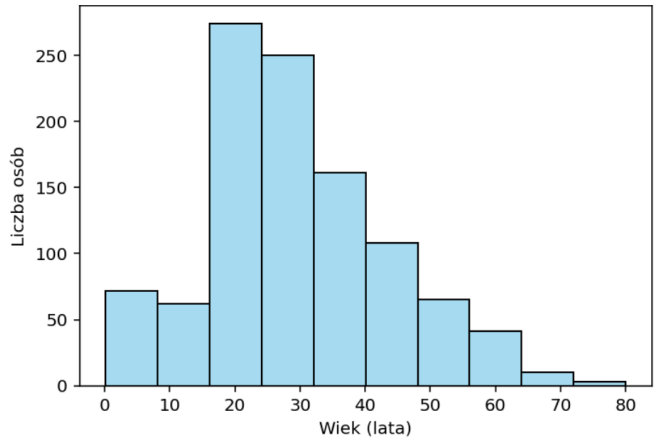






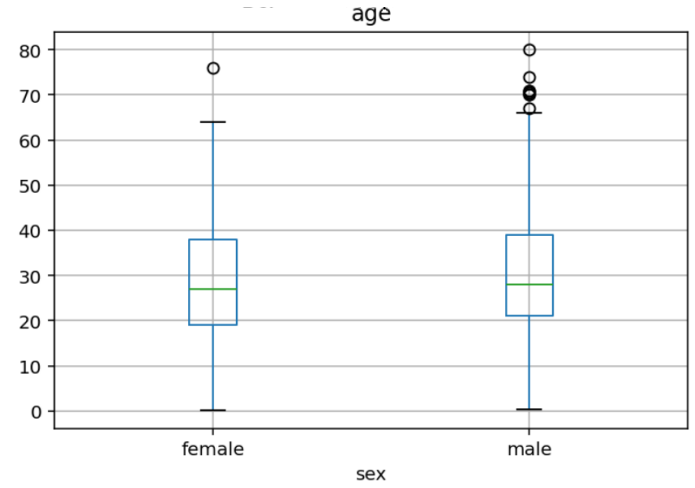


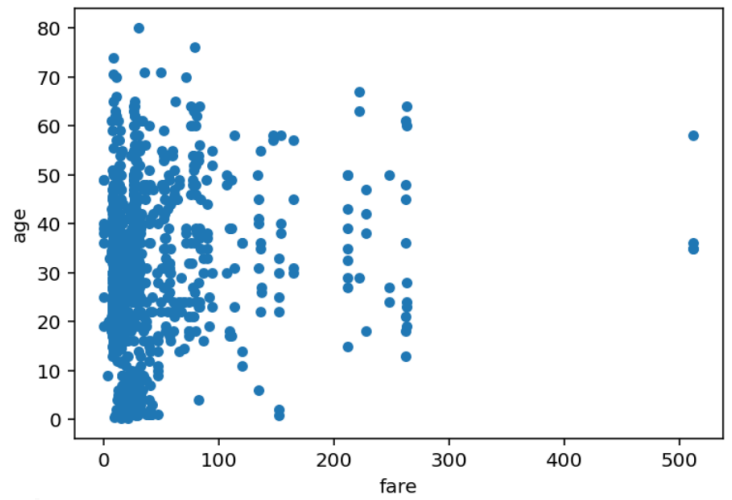


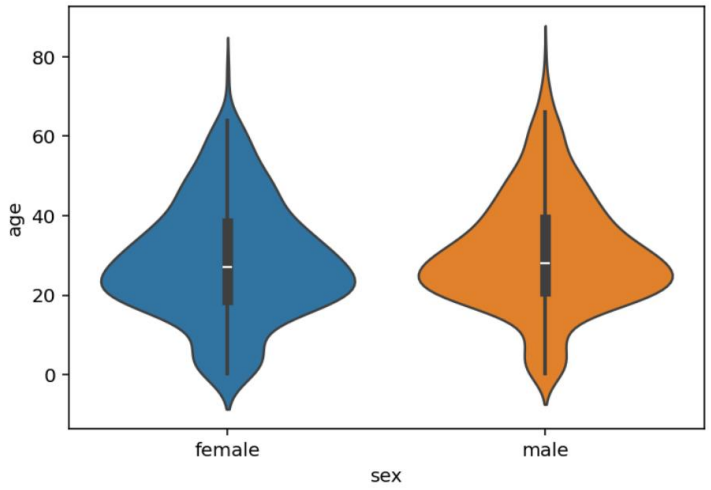


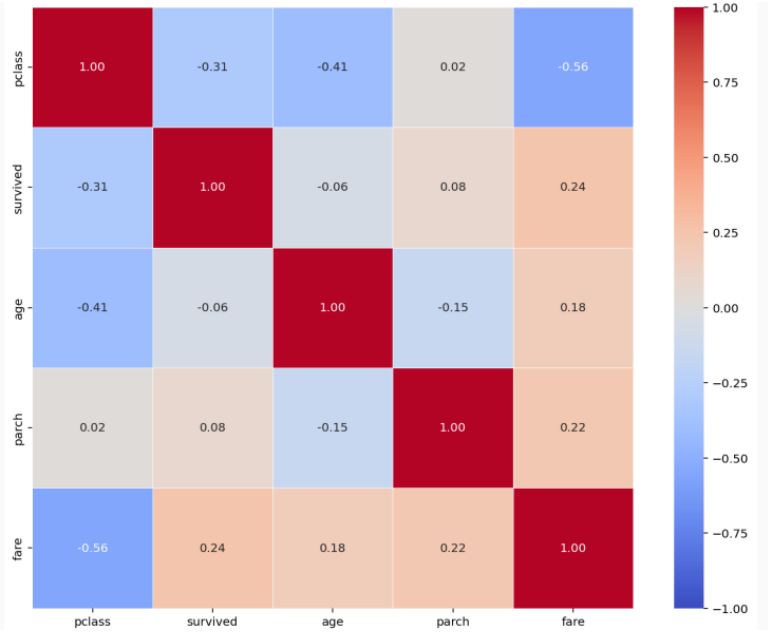
liniowy; obszarowy; kołowy; słupkowy pionowy; słupkowy poziomy, histogram

Rodzaje wykresów 2 - ilustracje









pudełkowy; punktowy (rozrzutu); wiolinowy (skrzypcowy); mapa cieplna (korelacji cech numerycznych na Titaniku)

Wybrane wykresy - legenda

```
# dane do wykresów
df1.plot(kind = 'scatter', y = 'age', x = 'fare')      # wykres rozrzutu
df1.plot(kind = 'hist', y = 'fare')                    # histogram
df1.boxplot(column=['age'], by='sex')
Sex=df1.groupby("sex")["survived"].count()
Sex.plot.barh(); Surv.plot.barh()

import seaborn as sb  # może być potrzebna instalacja: pip install seaborn
sb.violinplot(x=df1["age"])
sb.violinplot(data=df1, x="age", y='sex')
sb.violinplot(data=df1, x="pclass", y='age', hue='survived')
df1.hist()
df1["age"].hist()
df1.hist(column="age", by='sex', bins=10)

df1.groupby("pclass")["pclass"].count().plot.pie(figsize=(10,10), colors=["red",
"green", "blue"])
```

Student wczyta dane z pliku **titanic.csv** do ramki danych **df** i przestudiuje działanie programu.

Wizualizacja danych w seaborn

seaborn to biblioteka do wizualizacji danych w Pythonie, zbudowana na bazie *matplotlib* i ściśle zintegrowana z *pandas*

Umożliwia tworzenie wykresów (niebieską czcionką – wybrane metody)

- Dane ilościowe
  - zależności: scatterplot(), lineplot()
  - rozkłady: histplot()
  - macierze: heatmap()
  - regresja: regplot(), residplot()
- Dane jakościowe: boxplot(), violinplot(), stripplot(), swarmplot(), barplot()

Heatmap (mapa cieplna) to wizualizacja danych, w której wartości liczbowe są przedstawione za pomocą kolorów obrazując intensywność zjawiska.

Stripplot (wykres paskowy) przedstawia surowe, nieagregowane dane numeryczne, w tym w podziale na kategorie, jako punktowy rozrzut wzdłuż jednej osi.

Swarmplot (wykres roju) przedstawia surowe, nieagregowane dane numeryczne, w tym podziale na kategorie, jako nienakładające się punkty rozrzutu wzdłuż jednej osi.

https://www.kaggle.com/code/suruthi41/swarmplot-seaborn

Biblioteka seaborn - przykład

```python
import seaborn as sns
import matplotlib.pyplot as plt
# Matplotlib.pyplot to moduł biblioteki Matplotlib umożliwiajacy
# tworzenie wykresów i kontrolowania ich wyglądu w Pythonie.

import pandas as pd
import numpy as np

# Wczytanie stylu seaborn
#sns.set(style="whitegrid")

# -------------------------
# 1. HEATMAP
# Przykładowa macierz korelacji
data = pd.DataFrame({
    "A": np.random.randn(50),
    "B": np.random.randn(50),
    "C": np.random.randn(50)
})

plt.figure(figsize=(5, 4)) # rozmiar w calach; 1 cal = 2,54 cm
sns.heatmap(data.corr(), annot=True, cmap="coolwarm")
plt.title("Mapa cieplna - korelacja")
```

```python
# -------------------------
# Dane ilociowo-jakosciowe
df_cat = pd.DataFrame({
    "kategoria": np.repeat(["A", "B", "C"], 50),
    "wartosc": np.random.randn(150)
})
# 2. STRIPPLOT
plt.figure(figsize=(5, 4))
sns.stripplot(data=df_cat, x="kategoria", y="wartosc")
plt.title("Wykres paskowy - Stripplot")
plt.show()
# 3. SWARMPLOT
plt.figure(figsize=(5, 4))
sns.swarmplot(data=df_cat, x="kategoria", y="wartosc")
plt.title("Wykres roju - Swarmplot")
plt.show()
```