15 paskaita. Duomenų analizė ir vizualizacija

March 21, 2022

1 Duomenų analizė (su PANDAS)

Pandas dirba su duomenų paketais (DataFrame) Duomenų paketus galima susikurti iš (ir išsaugoti į):

- Žodynų
- Excel failų
- .csv failų
- Pickle failų
- Duomenų bazių ...

1.0.1 Kaip susikurti DataFrame iš tekstinio (csv) failo:

[ ]:
```
import pandas as pd

df = pd.read_csv('countries.csv', delimiter='\t')

print(df)
```

1.0.2 Kaip susikurti DataFrame iš duomenų bazės:

[ ]:
```
import pandas as pd
from sqlalchemy import create_engine

engine = create_engine('sqlite:///darbuotojai3.db')
df = pd.read_sql_table("DARBUOTOJAS", engine)

df
```

1.0.3 Kaip sukurti Pandas duomenų paketą (DataFrame) iš žodyno:

[ ]:
```
import pandas as pd
orai = {
'data': ['7/1/2019', '7/2/2019',
'7/3/2019','7/4/2019','7/5/2019'],
'temperatura': [32, 35, 28, 24, 22],
'vejas': [6,7,2,4,5],
```

```
'oras': ['Lietus', 'Saulėta',
'Saulėta','Saulėta','Debesuota']
}

df = pd.DataFrame(orai)
df
```

1.0.4 Kaip atspausdinti lentelės stulpelio max, min, vidurkį:

[ ]:
```
print(df['temperatura'].max())
print(df['temperatura'].min())
print(df['temperatura'].mean())
```

1.0.5 Kaip gauti lentelės dydį:

[ ]:
```
print(df.shape)
rows, columns = df.shape
print(rows)
print(columns)
```

1.0.6 Kaip atspausdinti dalį lentelės:

Pirmos penkios eilutės:

[ ]:
```
df.head()
```

Pirmos dvi eilutės:

[ ]:
```
df.head(2)
```

Paskutinė eilutė:

[ ]:
```
df.tail(1)
```

Nuo 1 iki 3 eilutės:

[ ]:
```
df[1:3]
```

Nuo 3 iki paskutinės eilutės:

[ ]:
```
df[2:]
```

1.0.7 Kaip atspausdinti norimą elementą:

[ ]:
```
df.iloc[2]
```

[ ]:
```
df.iloc[2:4]
```

1.0.8 Kaip atspausdinti konkrečią vietą pagal koordinates:

[ ]:
```
df.iloc[3,1]
```

1.0.9 Kaip atspausdinti stulpelių pavadinimus:

[ ]:
```
print(df.columns)
```

1.0.10 Kaip atspausdinti konkretų (-ius) stulpelį (-ius):

[ ]:
```
df.data
```

[ ]:
```
df[['data', 'temperatura']]
```

[ ]:
```
df.data[3:5]
```

1.0.11 Kaip iteruoti per visą DataFrame'ą:

[ ]:
```
for index, row in df.iterrows():
  print(index, row)
```

1.0.12 Kaip filtruoti duomenis DataFrame'e:

[ ]:
```
df.loc[df['temperatura'] > 22]
```

1.0.13 Kaip surūšiuoti duomenis:

[ ]:
```
df.sort_values('temperatura', ascending=False)
```

1.0.14 Kaip ieškoti pagal stringo dalį:

[ ]:
```
df.loc[df['oras'].str.contains('Saulė')]
```

1.0.15 Kaip grupuoti duomenis:

Galima naudoti funkcijas count, sum, min, max, mean

Darbuotojai sugrupuoti pagal pareigas, surūšiuota pagal daugiausiai išdirbančių darbuotojų vidurkius:

[ ]:
```
df.groupby(['oras']).mean().sort_values('temperatura',
ascending=False)
```

[ ]:
```
df.groupby(['oras']).count()['data']
```

1.0.16 Kaip atspausdinti lentelės ataskaitą:

[ ]:
```
df.describe()
```

1.0.17 Kaip eksportuoti DataFrame į Excel arba tekstinį failą:

[ ]:
```
df.to_excel('modifikuotas.xlsx', index=False)
```

[ ]:
```
df.to_csv('modifikuotas.txt', index=False, sep='\t')
```

2 Grafikų atvaizdavimas (Matplotlib):

2.0.1 Kaip suformuoti grafiką iš sąrašų:

[ ]:
```
from matplotlib import pyplot as plt

metai = [1978, 1988, 1998, 2008, 2018]
temperatura = [5.8, 5.9, 6.2, 6.9, 7.2]

plt.plot(metai, temperatura)
plt.show()
```

2.0.2 Kaip apipavidalinti grafiką:

[ ]:
```
from matplotlib import pyplot as plt

metai = [1978, 1988, 1998, 2008, 2018]
temperatura = [5.8, 5.9, 6.2, 6.9, 7.2]

plt.title("Vidutinė temperatūra")
plt.xlabel("Metai")
plt.ylabel("Laipsniai (Celcijaus)")

plt.plot(metai, temperatura)
plt.show()
```

2.0.3 Kaip pridėti papildomą kreivę:

[ ]:
```
from matplotlib import pyplot as plt

metai = [1978, 1988, 1998, 2008, 2018]
temperatura = [5.8, 5.9, 6.2, 6.9, 7.2]
ispanija = [15.8, 15.9, 16.2, 16.9, 17.2]

plt.title("Vidutinė temperatūra")
plt.xlabel("Metai")
```

```
plt.ylabel("Laipsniai (Celcijaus)")

plt.plot(metai, temperatura)
plt.plot(metai, ispanija)
plt.show()
```

2.0.4 Kaip įvardinti kreives:

[ ]:
```
from matplotlib import pyplot as plt

metai = [1978, 1988, 1998, 2008, 2018]
temperatura = [5.8, 5.9, 6.2, 6.9, 7.2]
ispanija = [15.8, 15.9, 16.2, 16.9, 17.2]

plt.title("Vidutinė temperatūra")
plt.xlabel("Metai")
plt.ylabel("Laipsniai (Celcijaus)")

plt.plot(metai, temperatura)
plt.plot(metai, ispanija)
plt.legend(["Vilnius", "Ispanija"])
plt.show()
```

2.0.5 Kaip atspausdinti grafiką iš Pandas duomenų paketo:

[ ]:
```
import pandas as pd
from matplotlib import pyplot as plt

orai = {
'data': ['7/1/2019', '7/2/2019',
'7/3/2019','7/4/2019','7/5/2019'],
'temperatura': [32, 35, 28, 24, 22],
'vejas': [6,7,2,4,5],
'oras': ['Lietus', 'Saulėta',
'Saulėta','Saulėta','Debesuota']}

df = pd.DataFrame(orai)

plt.plot(df.data, df.temperatura)
plt.show()
```

2.0.6 Kitokio tipo grafikas:

[ ]:
```
plt.plot(df.data, df.temperatura, "o")
```

2.0.7 Kaip atspausdinti grafiką iš atrinkto duomenų paketo:

[ ]:
```
import pandas as pd

data = pd.read_csv('countries.csv')

data
```

[ ]:
```
import pandas as pd
from matplotlib import pyplot as plt

data = pd.read_csv('countries.csv')
# iš https://www.csdojo.io/data

us = data[data.country == 'United States']
china = data[data.country == 'China']

plt.plot(us.year, us.population / 10**6)
plt.plot(china.year, china.population / 10**6)
plt.legend(['JAV', 'Kinija'])
plt.xlabel('Metai')
plt.ylabel('Populiacija (mln.)')
plt.show()
```

[ ]:
```
import pandas as pd
from matplotlib import pyplot as plt

data = pd.read_csv('countries.csv')

us = data[data.country == 'United States']
china = data[data.country == 'China']

plt.plot(us.year, us.population/us.population.iloc[0] * 100)
plt.plot(china.year, china.population/
china.population.iloc[0] * 100)
plt.legend(['USA', 'China'])
plt.xlabel('Metai')
plt.ylabel('Populiacijos kilimas (pirmi metai = 100 proc.)')
plt.show()
```

2.0.8 Užduotys:

1 užduotis

- Parsisiųsti countries.csv failą iš https://www.csdojo.io/data
- Susikurti iš jo duomenų paketą (Pandas DataFrame)
- Atspausdinti tik metų stulpelį
- Atspausdinti norimos šalies didžiausius/mažiausius/vidutinius gyventojų kiekius

- Atspausdinti bendrą lentelės ataskaitą
- Išsaugokite sukurtą DataFrame į duomenų bazę.
- Grafike atvaizduoti kelių skirtingų šalių populiacijos pokytį per metus
- Padaryti, kad grafikas turėtų pavadinimą, būtų įvardintos x/y ašys bei kreivės.

2 užduotis: Į Pandas DataFrame įdėkite duomenų bazę darbuotojai3.db ir savo nuožiūra išbandykite šioje paskaitoje pateiktas Pandas galimybes.

[ ]: