DSB\_S190523

Data Processing and Visualization Practice

Q1. **Can you print only the first three lines?**

*# TODO: write your code here*

data **=** {'X': [8.651201, 10.984780, 13.378190, 14.879830, 8.694768],

'Y': [17.891600, 18.517851, 17.394541, 17.809919, 14.905470]}

df **=** pd**.**DataFrame(data)

*# Printing the first three lines*

print(df**.**head(3))

X Y

0 8.651201 17.891600

1 10.984780 18.517851

2 13.378190 17.394541

Man kan vælge flere tilfældige rækker. Hvis dataen er sorteret på en måde, kan sampling faktisk give et upartisk billede af datasættet.

Man kan også finde ud hvor mange rækker der er i datasættet ved at køre:

len(pump\_df)

Men df giver ikke antallet af rækker, dog fortæller det om antallet af elementer.

pump\_df**.**size

Man kan også se på formen af datasættet, og hvilke kolonner der er i datasættet:

pump\_df**.**shape

(13, 2)

pump\_df**.**columns

Index(['X', 'Y'], dtype='object')

pump\_df**.**describe()

X Y

|  |  |  |
| --- | --- | --- |
| Count | 13.000000 | 13.000000 |
| Mean | 12.504677 | |  | | --- | | 11.963446 |  |  | | --- | |  | |
| std | 3.376869 | |  | | --- | | 4.957821 |  |  | | --- | |  | |
| Min | 8.651201 | |  | | --- | | 5.046838 |  |  | | --- | |  | |
| 25% | |  | | --- | | 8.999440 |  |  | | --- | |  | | 7.958250 |
| 50% | |  | | --- | | 12.571360 |  |  | | --- | |  | | |  | | --- | | 11.727170 |  |  | | --- | |  | |
| 75% | |  | | --- | | 14.879830 |  |  | | --- | |  | | 17.394541 |
| max | |  | | --- | | 18.914391 |  |  | | --- | |  | | 18.517851 |

Man kan også sætte data op som en liste:

pump\_df[:2]

|  |  |  |
| --- | --- | --- |
|  | X | Y |
| 0 | 8.651201 | |  | | --- | | 10.984780 |  |  | | --- | |  | |
| 1 | |  | | --- | | 10.984780 |  |  | | --- | |  | | 18.517851 |

Man kan også filtrere rækkerne, alt efter nogle specifikke betingelser

Q2. Vi skal indlæse datasættet

Indlæs dødsdatasættet

death\_url = "https://raw.githubusercontent.com/yy/dviz-course/master/data/deaths.csv" death\_df = pd.read\_csv(death\_url)

# Vis de første 2 rækker print(death\_df.head(2))

|  |  |  |
| --- | --- | --- |
|  | X | Y |
| 0 | 13.588010 | |  | | --- | | 11.09560 |  |  | | --- | |  | |
| 1 | 9.878124 | 12.55918 |

Derefter skal den vise det samlede antal rækker:

total\_rows = len(death\_df)

print(f"Total number of rows: {total\_rows}")

total number of rows: 578

Man kan også lave nogle visualiseringer vha. pandas.

death\_df.plot()

I princippet, er det egentlig ikke det vi har spurgt om. Fordi vi har bedt den om at plotte data, så den vil finde ud af hvad er denne type data. Man kan også run den her kommando hvis man ikke ser noget.

%matplotlib inline

Men vi ønsker at tegne et scatter plot ved at bruge y og x som koordinater:

death\_df.plot(x='X , y='Y', kind='scatter', label='Deaths')

Så får vi følgende plot:

Et billede, der indeholder skærmbillede

Automatisk genereret beskrivelse

Man kan også reducere størrelsen på prikkerne.

Man kan bruge denne kommando:

death\_df.plot(x='X', y='Y', kind='scatter', label='Deaths', s=2, c='black')

Hvor man bade reducere størrelsen på prikkerne, men samtidige ændre den til sort.

Et billede, der indeholder skærmbillede, diagram

Automatisk genereret beskrivelse

Det kan også visualiseres:

death\_df.plot(x='X', y='Y', s=2, c='black', kind='scatter', label='Deaths')

pump\_df.plot(x='X', y='Y', kind='scatter', c='red', s=8, label='Pumps')

Et billede, der indeholder skærmbillede, diagram

Automatisk genereret beskrivelse

Et billede, der indeholder skærmbillede, design

Automatisk genereret beskrivelse

Men vi har også to separate plots. Grunden til at vi burger plot metoden er for at skabe en ny figur I stedet for at placere dem i en enkelte figur. Vi kan oprette en akse. Og hvis man burger plot metoden, kan man få en aksen:

ax = death\_df.plot(x='X', y='Y', s=2, c='black', kind='scatter', label='Deaths')

Et billede, der indeholder skærmbillede, diagram

Automatisk genereret beskrivelse

Man kan også bruge denne plot-kommando, for at sige til den hvor den skal tegne punkterne:

ax = death\_df.plot(x='X', y='Y', s=2, c='black', alpha=0.5, kind='scatter', label='Deaths')

pump\_df.plot(x='X', y='Y', kind='scatter', c='red', s=8, label='Pumps', ax=ax)

Et billede, der indeholder kort, skærmbillede, diagram

Automatisk genereret beskrivelse

Derudover ved at bruge subplots(), kan man skabe en tom figur, og den vil blot returnere figuren og aksen til dig:

import matplotlib.pyplot as plt

fig, ax = plt.subplots()

Et billede, der indeholder Rektangel, skærmbillede, hvid, billedramme

Automatisk genereret beskrivelse

Q3 :Voronoi driagram

Hvor der kan bruges disse :

from scipy.spatial import Voronoi, voronoi\_plot\_2d

Vi skal producere et Voronoi diagram:

# du får brug for dette

points = pump\_df.values

points

array([[ 8.6512012, 17.8915997],

[10.9847803, 18.5178509],

[13.37819 , 17.3945408],

[14.8798304, 17.8099194],

[ 8.694768 , 14.9054699],

[ 8.8644161, 12.75354 ],

[12.5713596, 11.72717 ],

[10.6609697, 7.428647 ],

[13.5214596, 7.95825 ],

[16.4348907, 9.2521296],

[18.9143906, 9.7378187],

[16.0051098, 5.0468378],

[ 8.9994402, 5.1010232]])

# TODO: din kode her

import numpy as np

import matplotlib.pyplot as plt

from scipy.spatial import Voronoi, voronoi\_plot\_2d

# Antag at du har dødskoordinaterne i en DataFrame kaldet 'death\_df'

deaths\_coordinates = death\_df[['X', 'Y']].values

# Antag at du har pumpkoordinaterne i en numpy-array kaldet 'points'

points = np.array([[ 8.6512012, 17.8915997],

[10.9847803, 18.5178509],

[13.37819 , 17.3945408],

[14.8798304, 17.8099194],

[ 8.694768 , 14.9054699],

[ 8.8644161, 12.75354 ],

[12.5713596, 11.72717 ],

[10.6609697, 7.428647 ],

[13.5214596, 7.95825 ],

[16.4348907, 9.2521296],

[18.9143906, 9.7378187],

[16.0051098, 5.0468378],

[ 8.9994402, 5.1010232]])

# Kombiner døds- og pumpkoordinater

all\_coordinates = np.vstack((deaths\_coordinates, points))

# Opret Voronoi-diagram

vor = Voronoi(all\_coordinates)

# Plot Voronoi-diagram

fig, ax = plt.subplots()

voronoi\_plot\_2d(vor, ax=ax, show\_vertices=False, line\_colors='black', line\_width=2, line\_alpha=0.6, point\_size=5)

# Plot dødsfald

ax.scatter(deaths\_coordinates[:, 0], deaths\_coordinates[:, 1], c='red', label='Deaths', marker='x')

# Plot pumper

ax.scatter(points[:, 0], points[:, 1], c='blue', label='Pumps', marker='o

# Tilpas plottet

ax.set\_title('Voronoi Diagram med Dødsfald og Pumper')

ax.legend()

plt.show()

Et billede, der indeholder skærmbillede, linje/række

Automatisk genereret beskrivelse

Dernæst skal vi gemme figuren:

import matplotlib.pyplot as plt

plt.plot([1,2,3], [4,2,3])

plt.savefig('foo.png')

Et billede, der indeholder linje/række, design

Automatisk genereret beskrivelse

Derudover skal vi også gennem diagrammet:

import numpy as np

import matplotlib.pyplot as plt

from scipy.spatial import Voronoi, voronoi\_plot\_2d