Visualisering med Python

Alex

Visualisering med Python - varför?

- **Insikter**: Upptäcka mönster, trender och relationer i data.
- **Kommunikation**: Tydlig och effektiv kommunikation av komplex data.
- **Beslutsfattande**: Understödja informerade beslut med överskådlig presenteration.
- **Identifiera avvikelser**: Upptäcka avvikelser och "avstickare" i datasetet.
- **Förenkling**: Sammanfatta stora mängder data på ett lättbegripligt sätt.

Det finns många bibliotek

- **Matplotlib**: Basbibliotek för 2D-figurer med stöd för många olika diagramtyper.
- **Seaborn**: Bygger på Matplotlib men erbjuder ett mer estetiskt tilltalande och lättanvänt API för statistiska grafer.
- **Plotly**: Interaktiva visualiseringar som fungerar bra för webbaserade applikationer.
- **Pandas**: Har inbyggda funktioner som fungerar bra för snabb och enkel visualisering av DataFrames.
- **Bokeh**: Skapar interaktiva visualiseringar för webben med en enkel syntax.
- mfl

Vi kommer främst fokusera på biblioteket `Matplotlib` i denna kurs

```
import matplotlib.pyplot as plt
```

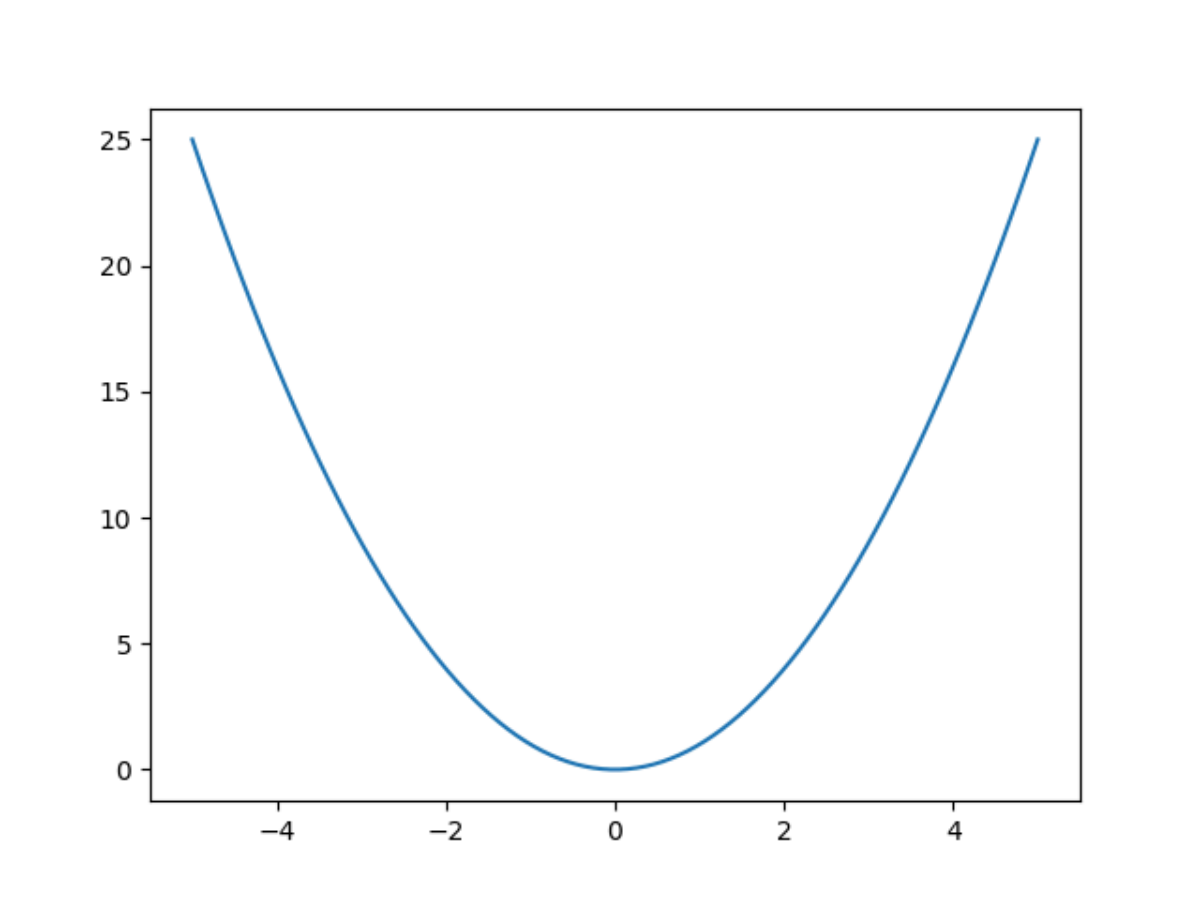
Exempelanvändning

```
import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(-5, 5, 100)
y = x**2

plt.plot(x, y, '-')
plt.show()
```

Fönster öppnas, och ser ut så här:



Spara figur

```
plt.savefig('figur.png')
```

Ange eventuellt sökväg till bilden:

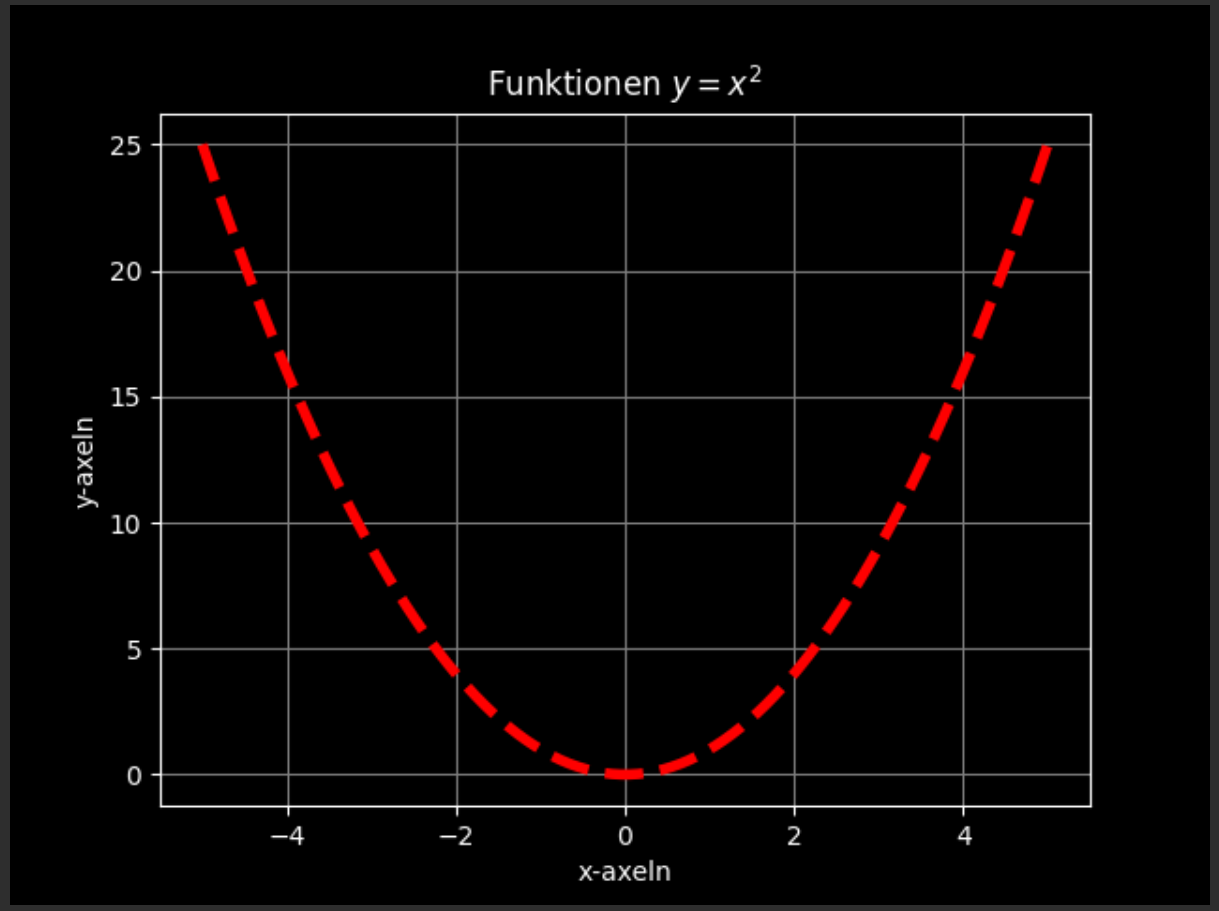
```
plt.savefig('path/to/figure/figur.png')
```

Några modifikationer

```
plt.style.use('dark_background')
plt.plot(x, y, '-', linewidth=4, linestyle='dashed', color='red')
plt.title(r'Funktionen $y = x^2$')
plt.xlabel('x-axeln')
plt.ylabel('y-axeln')
plt.grid(True, color='gray')
plt.show()
```

Plot på nästa sida →

Modifierad figur med samma data



Resten av presentationen sker i ipynb-filen.

Figurskapandet möjliggör också att titta lite närmare på olika NumPy-funktioner