

Python-kode for økonomi beregninger

Skrivet av Mohammed Altawil og Abdulaziz Ibrahim Wali

```
import numpy_financial as npf # Importerer numpy_financial-biblioteket og gir det nytt navn til npf
import matplotlib.pyplot as plt # Importerer matplotlib-biblioteket og gir det nytt navn til plt
import numpy as np # Importerer numpy-biblioteket og gir det nytt navn til np

# Spør om antall perioder

num_periods = int(input("Angi antall perioder: ")) # Spør brukeren om antall perioder og konverterer inndata til et heltall

# Be om den første investeringen

initial_investment = float(input("Angi den første investeringen: ")) # Spør brukeren om den første investeringen og konverterer input til en flyter

# Be om kontantstrømmen for periode 1

cash_flow_first_period = float(input("Angi kontantstrømmen for periode 1: ")) # Spørre brukeren om kontantstrømmen for periode 1 og konvertere input til en flyter

# Spør brukeren om kontantstrømmene er like for alle perioder

same_cash_flows = input("Er kontantstrømmene de samme for alle perioder? (ja/nei): ").lower().strip() # Spørre om kontantstrømmene er like for alle perioder

Hvis kontantstrømmene er like for alle perioder, bruk verdien for alle perioder

cash_flows = [cash_flow_first_period] * (num_periods - 1) # Lager en liste over kontantstrømmer med en enkelt verdi for alle perioder unntatt den første else:

# Hvis kontantstrømmene er forskjellige for hver periode, be om kontantstrømmen for hver periode
cash_flows = [cash_flow_first_period] # Legger til kontantstrømmen for den første perioden i listen
for i in range(1, num_periods): # Løkke for å be om kontantstrøm for hver periode
    periode = i + 1
    cash_flow = float(input(f"Angi kontantstrømmen for periode {periode}: ")) # Ber om kontantstrømmen for gjeldende periode
    cash_flows.append(cash_flow) # Legger til kontantstrømmen i listen
```

```

# Legg til den første investeringen øverst i listen over kontantstrømmer
cash_flows.insert(0, initial_investment) # Legger til den første investeringen øverst i listen over kontantstrømmer

# Beregn IRR for hver periode
irrs = [] # Oppretter en liste for å lagre internrente
for i in range(num_periods + 1): # Løkke for å beregne IRR for hver periode
    irr = npf.irr(cash_flows[:i+1]) * 100 # Beregner internrente ved hjelp av numpy_financial-biblioteket
    irrs.append(irr) # Legger til internrente i listen

# Skriv ut IRR for hver periode
for i, irr in enumerate(irrs): # Løkke for å skrive ut internrente for hver periode
    print(f"IRR for periode {i}: {irr:.2f}%") # Skriver ut internrente for gjeldende periode med en spesifikk formatering

# Tegn IRR for hver periode
plt.plot(range(num_periods + 1), irrs, marker='o') # Tegner internrente på x-aksen med data punkter merket med sirkler
plt.xlabel('Periode') # Merker x-aksen plt.ylabel('IRR
(%)') # Merker y-aksen
plt.title('Internrente (IRR) for hver periode') # Setter tittelen på plottet
plt.grid(True) # Aktiverer rutenett på plottet plt.show() # Viser plottet

# Beregn tilbakebetalingstiden
cumulative_cash_flows = np.cumsum(cash_flows) # Beregner kumulative kontantstrømmer if
np.any(cumulative_cash_flows > 0): # Sjekker om det er positive kontantstrømmer

    positive_cash_flows = np.where(cumulative_cash_flows > 0)[0] # Finner den første perioden med positive
    kontantstrømmer

    tilbakebetalingstid = (positive_cash_flows[0]) / num_periods # Beregner tilbakebetalingstiden else:

    tilbakebetalingstid = None # Setter tilbakebetalingstiden til None hvis det ikke er noen positive kontantstrømmer

# Skriv ut resultatene
if tilbakebetalingstid is not None: # Hvis tilbakebetalingstiden eksisterer

```

```

print(f"Tilbakebetalingstid: {tilbakebetalingstid:.2f} år") # Skriver ut tilbakebetalingstiden else:

print("Ingen positive kontantstrømmer, tilbakebetalingstiden er ikke aktuell.") # Skriver ut en melding hvis
tilbakebetalingstiden ikke er aktuell


# Beregn NPV for hver periode

diskonteringsrente = 0.06 # Setter diskonteringsrenten

npvs = [npf.npv(diskonteringsrente, cash_flows[:i+1]) for i in range(num_periods + 1)] # Beregner NPV for hver
periode


# Tegn NPV for hver periode

plt.plot(range(num_periods + 1), npvs, marker='o') # Tegner NPV for hver periode
plt.xlabel('Periode') # Merker x-aksen plt.ylabel('NPV') # Merker y-aksen
plt.title('Netto nåverdi (NPV) for hver periode') # Setter tittelen på plottet
plt.grid(True) # Aktiverer rutenett på plottet plt.show() # Viser plottet


# Skriv ut NPV for hver periode

for i, npv in enumerate(npvs): # Løkke for å skrive ut NPV for hver periode
    print(f"NPV for periode {i}: {npv:.2f}") # Skriver ut NPV for gjeldende periode med en spesifikk formatering


# Beregn nettoinntekt

totalinntekt = sum(cash_flows) # Beregner totalinntekt total_kostnad =
abs(initial_investment) # Beregner total kostnad nettoinntekt =
totalinntekt - total_kostnad # Beregner nettoinntekt


# Beregn gjennomsnittlig investering

gjennomsnittlig_investering = sum(cash_flows) / num_periods # Beregner gjennomsnittlig investering

```