

6. Risikable aktiver

Skriv dit eksamensnummer her: 45

Start med at sætte dit personlige seed nedenfor, hvilket er dit eksamensnummer. Brug af andre medstuderendes seed betragtes som eksamenssnyd!

```
In [1]: seed = 45 # <== Sæt dit seed lig dit eksamensnummer her
```

```
In [2]: import numpy as np
import math
import matplotlib.pyplot as plt
import ipywidgets as wg
from scipy.optimize import fsolve
%matplotlib inline
```

En investor har Bernoulli-nyttefunktion givet ved:

$$u(w) = 1 - \frac{1}{1.005^w}.$$

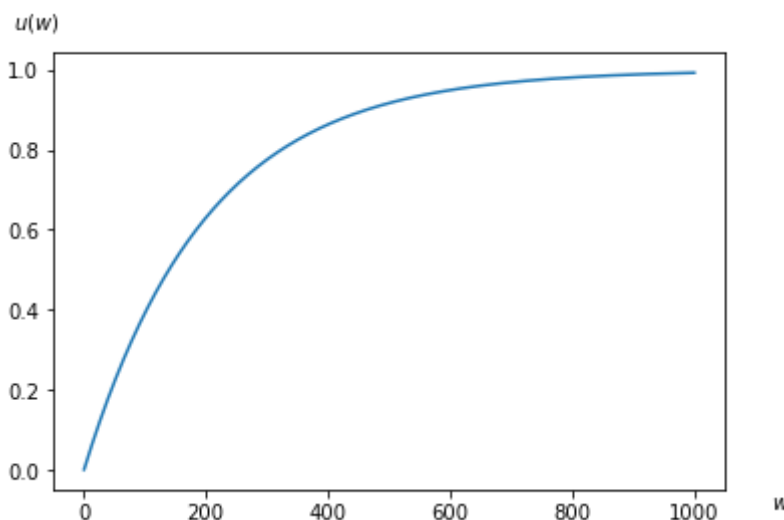
Opgave 6.a)

- Lav et plot af investorens nytte som funktion af $w \in [0, 1000]$
- Argumenter ud for figuren for, om investoren er risikoavers, risikoneutral eller risikoelsker

```
In [3]: # SKRIV DIN LØSNING HER:
def u(w):
    return 1 - 1 / (1.005**w)

W = np.linspace(0, 1000, 1000)
plt.plot(W, u(W))

plt.ylabel('$u(w)$', y=1.1, va='top', labelpad=-15, rotation='horizontal')
plt.xlabel('$w$', x=1.1, ha='right', labelpad=-15);
```



Da grafen er konkav er investoren risikoavers.

Opgave 6.b)

Betragt en aktie med værdi x , der er normalfordelt:

$$x \sim N(500, 200^2).$$

- Træk 10,000 realisationer af x
- Brug de 10,000 træk til at udregne (approximativt) den forventede nytte af x

In [4]: `np.random.seed(seed) # OBS: SLET IKKE DENNE LINJE!`

```
# SKRIV DIN LØSNING HER:
x = np.random.normal(500, 200, 10_000)
print(u(x).mean())
```

0.865101304995566

Opgave 6.c)

Lad y være værdien af en Bitcoin, der er normalfordelt:

$$y \sim N(500, 250^2).$$

Desuden er x og y korrelerede med korrelationskoefficient $\rho \in [-1, 1]$.

Nedenstående funktion trækker 10,000 værdier af x og y med ρ som input:

```
In [5]: # EKSEKVER DENNE CELLE UDEN AT REDIGERE KODEN!
def træk_y_givet_x(rho):
    np.random.seed(seed)

    mean = [500, 500]
    cov = [[200**2, rho*200*250], [rho*200*250, 250**2]]

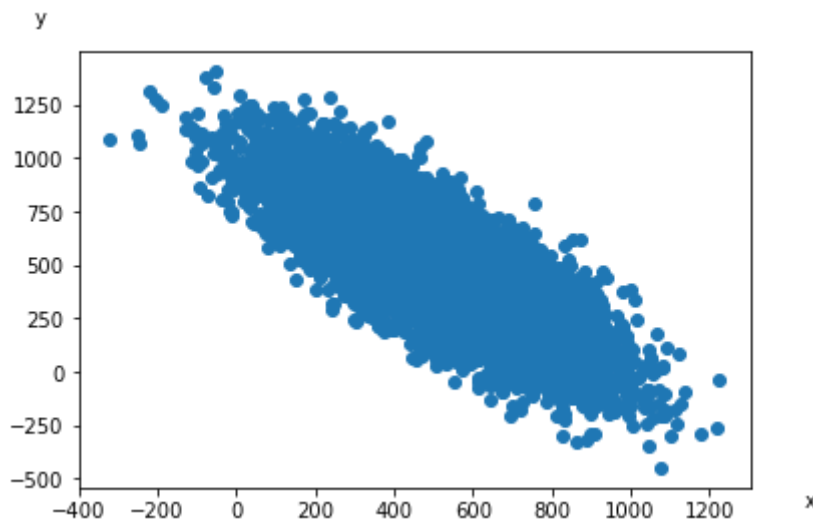
    XY = np.random.multivariate_normal(mean, cov, size=10_000)
    x = XY[:,0]
    y = XY[:,1]

    return x, y
```

- Brug funktionen til at lave et scatterplot, der viser sammenhængen mellem x og y for $\rho = -0.8$
- Beskriv sammenhængen mellem x og y ud fra din figur

```
In [6]: # SKRIV DIN LØSNING HER:
x,y = træk_y_givet_x(-0.8)
plt.ylabel('y', y=1.1, va='top', labelpad=-15, rotation='horizontal')
plt.xlabel('x', x=1.1, ha='right', labelpad=-15);
plt.scatter(x, y)
```

Out[6]: <matplotlib.collections.PathCollection at 0x193c3bf9b80>



Her ses det ud fra figuren, at når x bliver større, så har y en tendens til at blive mindre. Dvs. y (værdien af Bitcoin) afhænger negativt af x (værdien af aktien).

Opgave 6.d)

- Hvis investoren skal vælge enten x eller y , hvad foretrækker investoren? Forklar hvorfor.

```
In [7]: # SKRIV DIN LØSNING HER:
print(np.mean(u(x)))
print(np.mean(u(y)))
```

```
0.8651383027868486
0.8220688887401457
```

Investoren foretrækker x , da nytten er størst ved køb af aktien. Dette giver også god mening, da investoren er risikoavers, og bitcoin har en større spredning end aktien, hvilket gør den mere risikabel.

Opgave 6.e)

Antag nu, at investoren ejer x samt en formue på $w_0 = 1000$. Investoren overvejer at købe én Bitcoin til pris $p \in (0, w_0)$. Da er værdien af investorens portefølje afhængig af købsbeslutningen:

$$w_1 = \begin{cases} w_0 + x & \text{hvis investoren ikke køber Bitcoin} \\ w_0 + x + y - p & \text{hvis investoren køber Bitcoin} \end{cases}.$$

- Lav en funktionen, der tager to argumenter (inputs): p (prisen på Bitcoin) og ρ (korrelationen mellem x og y) og returnerer den forventede nytte i de to situationer (køb af Bitcoin/ikke køb af Bitcoin). **OBS:** Den første linje i funktionen skal være `np.random.seed(seed)`. Hint: Brug `træk_y_givet_x`.
- Plot den forventede nytte i de to situationer som funktion af prisen $p \in [0, 700]$
- Hvad er investorens maksimale betalingsvillighed for y ? Indiker dette i figuren med en lodret linje, så betalingsvilligheden kan aflæses på x-aksen.

```
In [8]: # SKRIV DIN LØSNING HER:
```

```

rho = -0.8
def uB(p, rho):
    np.random.seed(seed)
    x, y = træk_y_givet_x(rho)

    noBuy = np.mean(u(1000 + x))

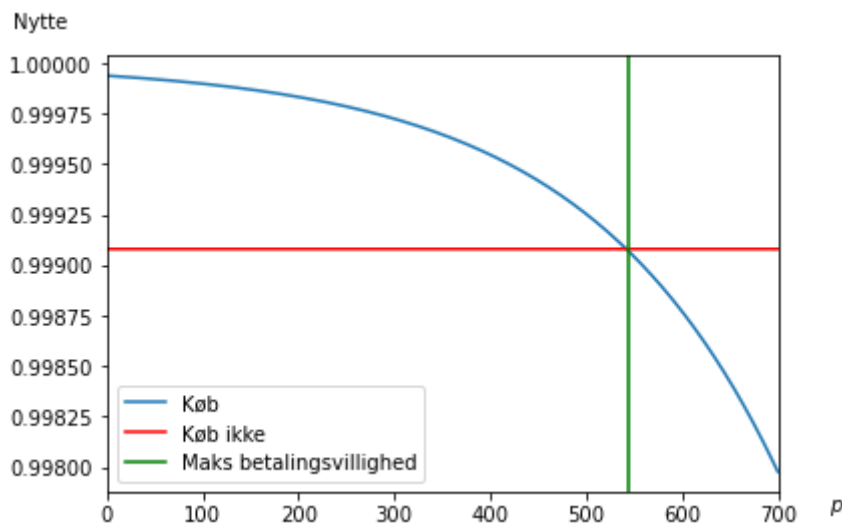
    buy = np.mean(u(1000 + x + y - p))

    return noBuy, buy
P = np.linspace(0, 700, 10000)
forvBuy = [uB(p, rho)[1] for p in P]
forvNoBuy = [uB(p, rho)[0] for p in P]

def maxPay():
    for i in range(P.size):
        if math.isclose((forvBuy[i] - forvNoBuy[i]), 0.0, abs_tol=0.00001):
            equal = P[i]
    return equal
equal = maxPay()

plt.plot(P, forvBuy, label='Køb')
plt.plot(P, forvNoBuy, label='Køb ikke', color = 'r')
plt.axvline(x=equal, label='Maks betalingsvillighed', color='g')
plt.ylabel('Nytte', y=1.1, va='top', labelpad=-15, rotation='horizontal')
plt.xlabel('$p$', x=1.1, ha='right', labelpad=-15)
plt.xlim([0, 700])
plt.legend();

```



Opgave 6.f)

- Lav din figur til en funktion med ρ som input og opret en floatslider fra -1 til 0 i skridt af 0.05 , der styrer denne
- Hvordan ændrer figuren og den maksimale betalingsvillighed sig, når du ændrer på korrelationen ρ ?
- Brug slideren til at finde (ca.) den største værdi af ρ , hvor investoren vil betale mere for y end dens forventede værdi (μ)

```

In [9]: # SKRIV DIN LØSNING HER:
rho_float_slider = wg.FloatSlider(
    value = -0.65,
    min=-1,

```

```

max=0,
step = 0.05,
description = 'rho',
continues_update = False)

def plotRho(rho):
    def maxPay():
        for i in range(P.size):
            if math.isclose((forvBuy[i] - forvNoBuy[i]), 0.0, abs_tol=0.00001):
                equal = P[i]
        return equal
    P = np.linspace(0, 700, 10000)
    forvBuy = [uB(p, rho)[1] for p in P]
    forvNoBuy = [uB(p, rho)[0] for p in P]
    plt.plot(P, forvBuy, label='Køb')
    plt.plot(P, forvNoBuy, label='Køb ikke', color = 'r')
    maxBetal = 'Maks betalingsvillighed=' + str(round(maxPay(), 2))
    plt.axvline(x=maxPay(), label=maxBetal, color='g')
    plt.ylabel('Nytte', y=1.1, va='top', labelpad=-15, rotation='horizontal')
    plt.xlabel('$p$', x=1.1, ha='right', labelpad=-15)
    plt.xlim([0, 700])
    plt.legend();

wg.interact(plotRho, rho = rho_float_slider)

interactive(children=(FloatSlider(value=-0.65, description='rho', max=0.0, min=-1.0, step=0.05), Output()), _d...
Out[9]: <function __main__.plotRho(rho)>

```

Det ses, at når rho bliver større, så bliver den maksimale betalingsvillighed større.

Ved brug af slideren, er den største værdi af rho, hvor investoren vil betale mere for y end dens forventede værdi er ved ca. $\rho = -0,65$, hvor den maksimale betalingsvillighed er på ca. 507