# SOK-2030 Seminar 3 - Monopol og produktdifferensiering

```
# Laster inn pakker.
import numpy as np
import sympy as sp
from sympy import *
from sympy.solvers import solve
from matplotlib import pyplot as plt
```

Kaffebønna er lokalisert i Tromsø med flere utsalgssteder. I første omgang antar vi at Kaffebønna er monopolist i sitt marked hvor kundene er horisontalt differenisert (den linære byen). Anta at Kaffebønna har 30 000 kunder ($N = 30.000$) som er jevnt fordelt langs lokaliseringsområdet til utsalgsstedene, og at hver kunde kjøper en kopp kaffe per dag. Kostnaden ved å produsere en kopp kaffe er NOK 5, og bedriften har faste kostnader per utsalgssted på NOK 25 000. Transportkostnadene for kundene er på NOK 50.
Hva er optimalt antall utsalgssteder for Kaffebønna i Tromsø?

### Optimal valg av utsalgssteder.

```
# Definerer symboler.
V, t, n, c, F, N = symbols("V t n c F N")
```

Profitten ved $N$ kunder og $n$ utsalgssteder er lik $(V - \frac{t}{2n} - c) \times N - nF$.
Vi skriver det på følgende vis:

```
# Definerer profittfunksjon.
def profit(V, t, c, n, F, N):
    return (V-(t/(2*n))-c)*N-n*F

# Printer funksjonen.
```

```
profit(V, t, c, n, F, N)
```

$$-Fn + N\left(V - c - \frac{t}{2n}\right)$$

For å finne optimalt antall utsalgssteder så deriverer vi profittfunksjonen med hensyn på antall utsalgssteder ($n$).

```
# Deriverer med hensyn på n.
d_profit = sp.diff(profit(V, t, c, n, F, N), n)
d_profit
```

$$-F + \frac{Nt}{2n^2}$$

Videre setter vi den deriverte lik 0, og finner optimalt nivå på antall utsalgsteder ($n$).

```
# Setter den deriverte lik 0.
foc = sp.Eq(d_profit, 0)
foc
```

$$-F + \frac{Nt}{2n^2} = 0$$

Videre må vi løse med hensyn på antall utsalgssteder ($n$).

```
# Løser med hensyn på antall utsalgssteder.
n_max = solve(foc, n)[1]
n_max
```

$$\frac{\sqrt{2}\sqrt{\frac{Nt}{F}}}{2}$$

```
# Definerer tallverdier fra oppgavetekst.
num_dict = {N:30000,F:25000,t:50}

# N - Antall kunder
# F - Faste kostnader per utsalgssted
# t - Transportkostnader
```

Dersom vi nå putter inn verdiene fra oppgaveteksten får vi at:

```
# Legger inn verdiene fra oppgaveteksten.
n_max.subs(num_dict)
```

$\sqrt{30}$

```
# Finner tallsvar.
round(n_max.subs(num_dict), 2)
```

5.48

Altså er $30^{0.5} = 5.48$ optimalt antall serveringsteder for kaffebønna.

### Samfunnsøkonomisk optimalt antall utsalg.

```
# Definerer totalkostnadsfunksjon.
def totalcost(t, n, F, N):
    return ((N*t)/(4*n)+n*F)

totalcost(t, n, F, N)
```

$$Fn + \frac{Nt}{4n}$$

```
# Deriverer total kostnadsfunksjon med hensyn på n.
d_totalcost = sp.diff(totalcost(t, n, F, N), n)

display(d_totalcost)
```

$$F - \frac{Nt}{4n^2}$$

```
# Setter den deriverte lik 0.
foc1 = sp.Eq(d_totalcost, 0)

foc1
```

$$F - \frac{Nt}{4n^2} = 0$$

```
# Løser med hensyn på n.
n_min = solve(foc1, n)[1]

n_min
```

$$\frac{\sqrt{\frac{Nt}{F}}}{2}$$

```python
# Bruker tallene fra oppgaven til å finne samf. optimal.
n_min.subs(num_dict)
```

$\sqrt{15}$

```python
# Finner tallsvar.
round(n_min.subs(num_dict), 2)
```

3.87