

SOK 2030 Mappeoppgave 1 - Kode

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

Kapittel 2

2.1 Monopol og optimal tilpasning

```
# Definerer symboler.  
q = sp.symbols("q", real = True, positive = True)  
  
# Definere nødvendige funksjoner.  
# Definerer marginalinntekten.  
def mr(q):  
    return(120-18*q/50)  
  
# Definerer marginalkostnaden.  
def mc(q):  
    return(10+q/25)  
  
# Definerer etterspørselsfunksjonen.  
def dem(q):  
    return(120-9*q/50)  
  
# Finner likevektspunktet.  
# Setter marginalkostnaden lik marginalinntekten.  
equ = sp.Eq(mr(q), mc(q))
```

```

# Løser med hensyn på q for å finne likevektskvantum.
q_equ = sp.solve(equ, q)[0]

# Legger likevektskvantum inn i etterspørselsfunksjonen for å finne likevektsprisen.
p_equ = dem(q_equ)

# Viser likevekts- pris og kvantum.
print(f"Likevekstkvanum er lik:", round(q_equ, 2))
print(f"Likevekstprisen er lik:", round(p_equ, 2))

```

Likevekstkvanum er lik: 275
Likevekstprisen er lik: 70.50

```

# Figur 2.1.
# Definerer intervall mellom 0 og 800, med 800 punkter.
q_01 = np.linspace(1, 800, 800)

# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_01, dem(q_01), label = "Etterspørsel", color = "#2173B8")
plt.plot(q_01, mc(q_01), label = "Marginalkostnad", color = "#FF8000")
plt.plot(q_01, mr(q_01), label = "Marginalinntekt", color = "#228B22")

# Markerer likevektspunkter.
plt.plot(q_equ, p_equ, marker = "o", color = "red")
plt.hlines(p_equ, xmin = 0, xmax = q_equ, linestyle = "dashed", color = "#D2042D")
plt.vlines(q_equ, ymin = 0, ymax = p_equ, linestyle = "dashed", color = "#D2042D")

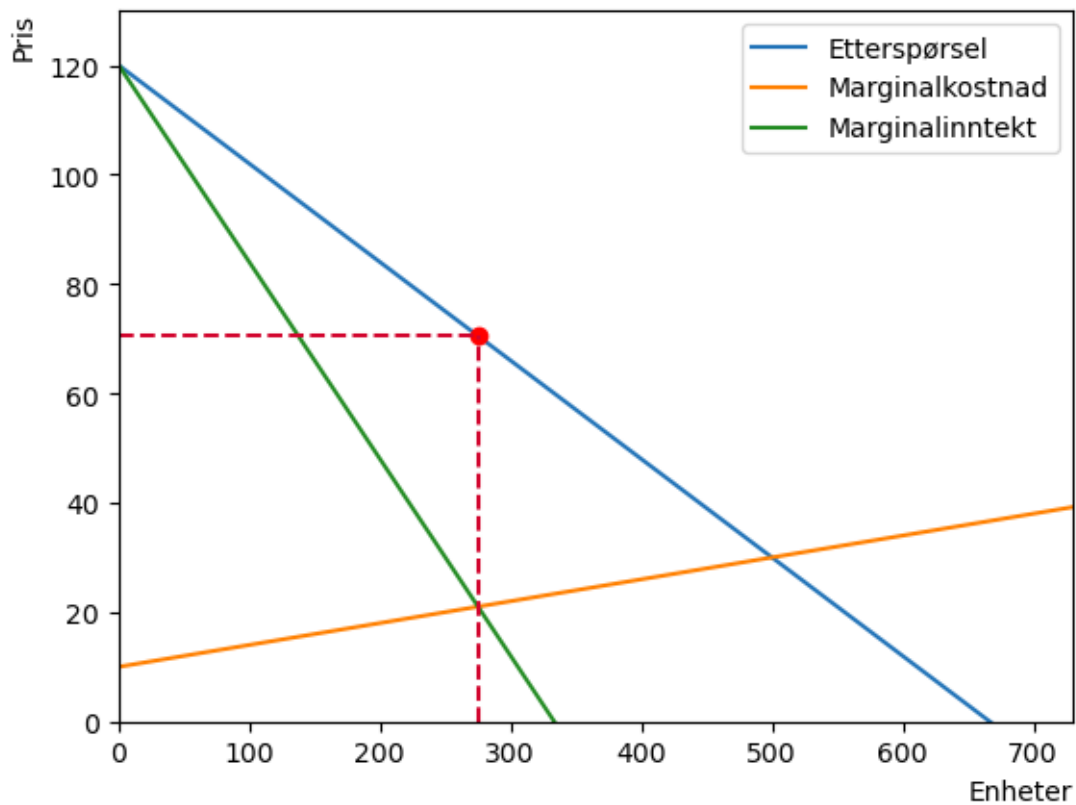
# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Begrenser figuren.
plt.ylim(0, 130)
plt.xlim(0, 730)

# Legger til legend.
plt.legend(loc = "upper right")

```

<matplotlib.legend.Legend at 0x7feae4ee9fd0>



Kapittel 2.4 Effekten av endringer i etterspørselen på markedstilpasning

```
# Definerer etterspørsel.
def dem(q):
    return(120-9*q/50)

# Definerer ny etterspørsel.
def dem_opp(q):
    return(140-9*q/50)

# Definerer ny marginalinntekt.
def mr_n(q):
    return(140-18*q/50)

# Finner likevektspunktet.
# Setter marginalkostnaden lik marginalinntekten.
equ1 = sp.Eq(mr_n(q), mc(q))
```

```

# Løser med hensyn på q for å finne likevektskvantum.
q_equ1 = sp.solve(equ1, q)[0]

# Legger likevektskvantum inn i etterspørselsfunksjonen for å finne likevektsprisen.
p_equ1 = dem_opp(q_equ1)

# Viser likevekts- pris og kvantum.
print(f"Likevekstkvanum er lik:", round(q_equ1, 2))
print(f"Likevekstprisen er lik:", round(p_equ1, 2))

```

Likevekstkvanum er lik: 325
Likevekstprisen er lik: 81.50

```

# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_01, dem(q_01), label = "Gammel etterspørsel",
         color = "#2173B8", alpha = 0.4)
plt.plot(q_01, mc(q_01), label = "Marginalkostnad",
         color = "#FF8000")
plt.plot(q_01, mr(q_01), label = "Gammel marginalinntekt",
         color = "#228B22", alpha = 0.4)

# Plotter ny etterspørsel og marginalinntekt.
plt.plot(q_01, dem_opp(q_01), label = "Ny etterspørsel", color = "#A020F0")
plt.plot(q_01, mr_n(q_01), label = "Ny marginalinntekt", color = "#E1C16E")

# Markerer likevektspunkter.
plt.plot(q_equ, p_equ, marker = "o",
         color = "red", alpha = 0.4)
plt.plot(q_equ1, p_equ1, marker = "o",
         color = "red")
plt.hlines(p_equ, xmin = 0, xmax = q_equ, linestyle = "dashed",
         color = "#D2042D", alpha = 0.4)
plt.hlines(p_equ1, xmin = 0, xmax = q_equ1, linestyle = "dashed",
         color = "#D2042D")
plt.vlines(q_equ, ymin = 0, ymax = p_equ, linestyle = "dashed",
         color = "#D2042D", alpha = 0.4)
plt.vlines(q_equ1, ymin = 0, ymax = p_equ1, linestyle = "dashed",
         color = "#D2042D")

# Gir aksene titler.

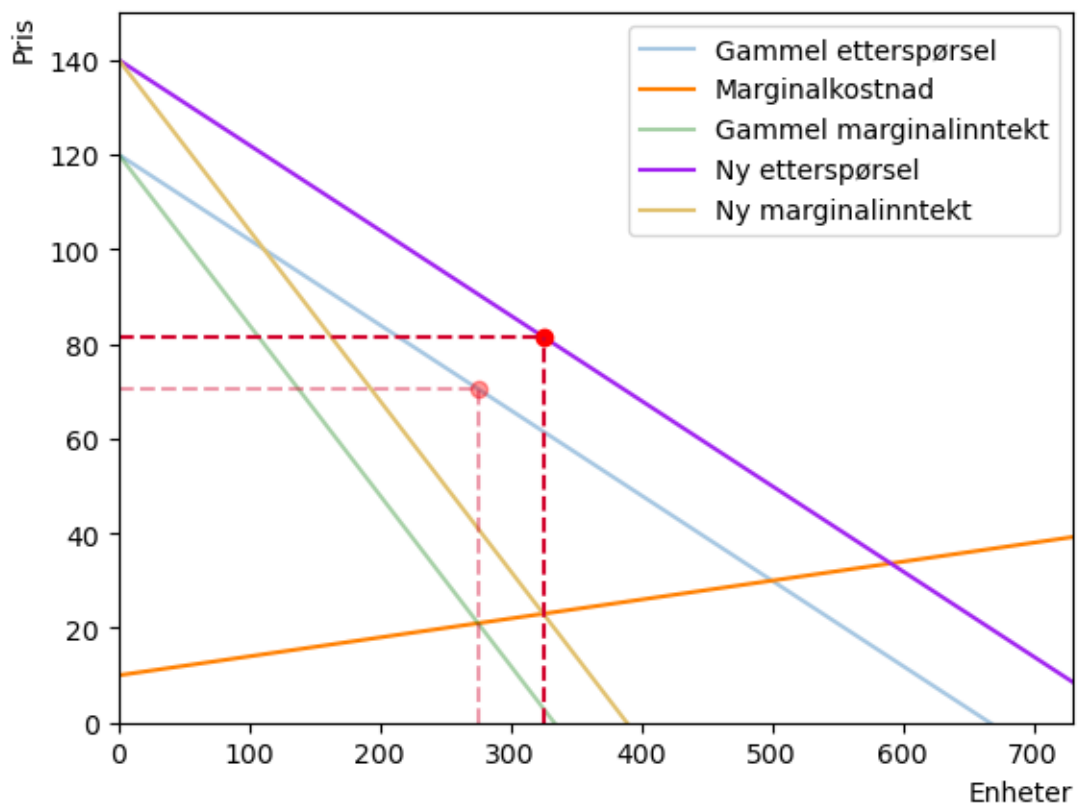
```

```
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Begrenser figuren.
plt.ylim(0, 150)
plt.xlim(0, 730)

# Legger til legend.
plt.legend(loc = "upper right")
```

<matplotlib.legend.Legend at 0x7feae4f68f40>



Kapittel 3

3.1 Optimal tilpasning - Matematisk

```

# Definerer symbol.
Qv, Qs, Q = sp.symbols("Qv Qs Q", real = True, positive = True)

# Definerer etterspørsel vinter.
def dem_v(Qv):
    return 7500-12.5*Qv

# Definerer etterspørsel sommer.
def dem_s(Qs):
    return 2000-2.5*Qs

# Definerer marginalinntekt vinter, utregning i besvarelse.
def mr_v(Qv):
    return 7500-25*Qv

# Definerer marginalinntekt sommer, utregning i besvarelse.
def mr_s(Qs):
    return 2000-5*Qs

# Setter marginalinntekt vinter lik marginalkostnad (800).
equ_mrv = sp.Eq(mr_v(Qv), 800)

# Setter marginalinntekt sommer lik marginalkostnad (800)
equ_mrs = sp.Eq(mr_s(Qs), 800)

# Løser med hensyn på Q.
equ_Qv = sp.solve(equ_mrv)[0]
equ_Qs = sp.solve(equ_mrs)[0]

# Viser likevektskvantum.
print(f"Likevektskvantum på vinter er lik:", equ_Qv)
print(f"Likevektskvantum på sommeren er lik:", equ_Qs)

```

Likevektskvantum på vinter er lik: 268
 Likevektskvantum på sommeren er lik: 240

```

# Finner likevekstprisen.
equ_Pv = dem_v(equ_Qv)
equ_Ps = dem_s(equ_Qs)

```

```
# Viser likevekstprisen.
print(f"Likevekstprisen på vinteren er lik:", equ_Pv)
print(f"Likevekstprisen på sommeren er lik:", equ_Ps)
```

Likevekstprisen på vinteren er lik: 4150.000000000000
 Likevekstprisen på sommeren er lik: 1400.000000000000

```
# Finner profitten til hotellet.
profit_v = (equ_Pv-800)*equ_Qv
profit_s = (equ_Ps-800)*equ_Qs

# Viser profitten til hotellet.
print(f"Profitten til hotellet per døgn på vinteren er lik:", profit_v)
print(f"Profitten til hotellet per døgn på sommeren er lik:", profit_s)
```

Profitten til hotellet per døgn på vinteren er lik: 897800.0000000000
 Profitten til hotellet per døgn på sommeren er lik: 144000.0000000000

3.1 Optimal tilpasning - Grafisk

```
# Figur 3.1
# Definerer intervall mellom 0 og 800, med 800 punkter.
q_02 = np.linspace(1, 800, 800)
# Definerer intervall mellom 800 og 800, med 800 punkter.
q_x = np.linspace(800, 800, 800)

# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_02, dem_v(q_02), label = "Etterspørsel")
plt.plot(q_02, mr_v(q_02), label = "Marginalinntekt")
plt.plot(q_02, q_x, label = "Marginalkostnad")

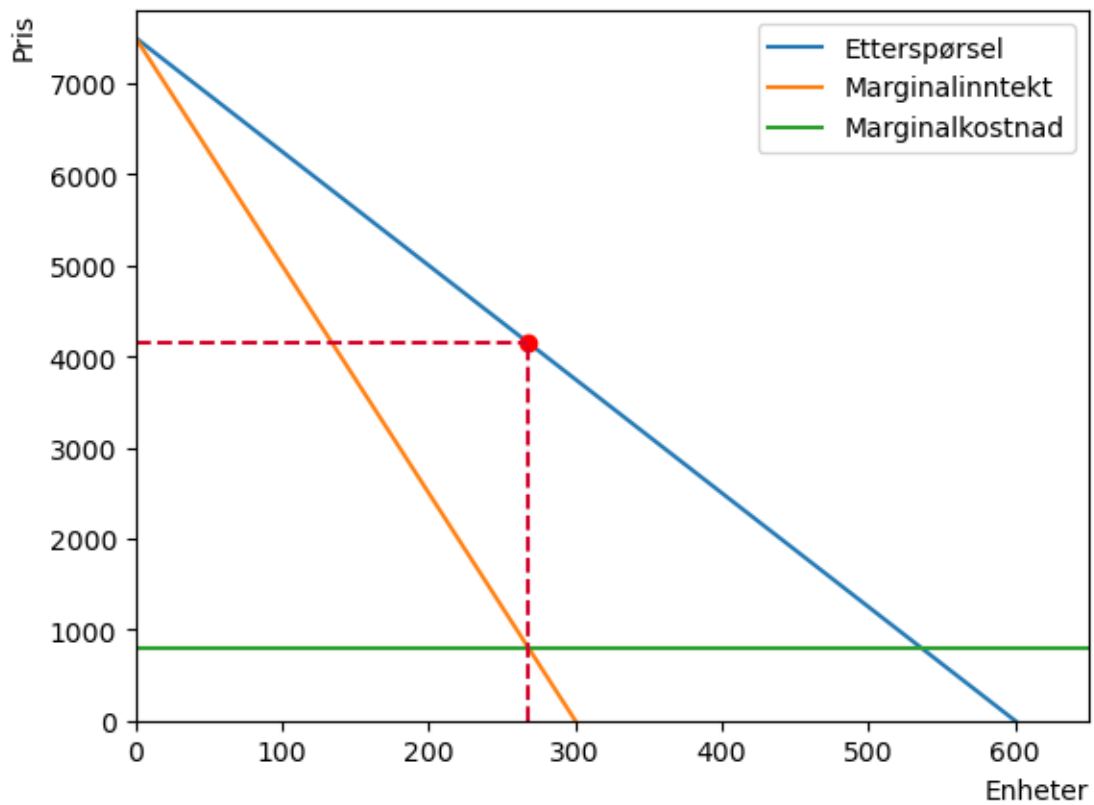
# Markerer likevektspunkter.
plt.plot(equ_Qv, equ_Pv, marker = "o",
         color = "red")
plt.hlines(equ_Pv, xmin = 0, xmax = equ_Qv, linestyle = "dashed",
          color = "#D2042D")
plt.vlines(equ_Qv, ymin = 0, ymax = equ_Pv, linestyle = "dashed",
          color = "#D2042D")
```

```
# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Avgrenser figuren.
plt.xlim(0, 650)
plt.ylim(0, 7800)

# Legger til legend.
plt.legend(loc = "upper right")
```

<matplotlib.legend.Legend at 0x7feadc4b2fa0>



Figur 3.2

Plotter etterspørsel, marginal- kostnad og inntekt.


```

plt.plot(q_02, dem_s(q_02), label = "Etterspørsel")
plt.plot(q_02, mr_s(q_02), label = "Marginalinntekt")
plt.plot(q_02, q_x, label = "Marginalkostnad")

# Markerer likevektspunkter.
plt.plot(equ_Qs, equ_Ps, marker = "o",
         color = "red")
plt.hlines(equ_Ps, xmin = 0, xmax = equ_Qs, linestyle = "dashed",
         color = "#D2042D")
plt.vlines(equ_Qs, ymin = 0, ymax = equ_Ps, linestyle = "dashed",
         color = "#D2042D")

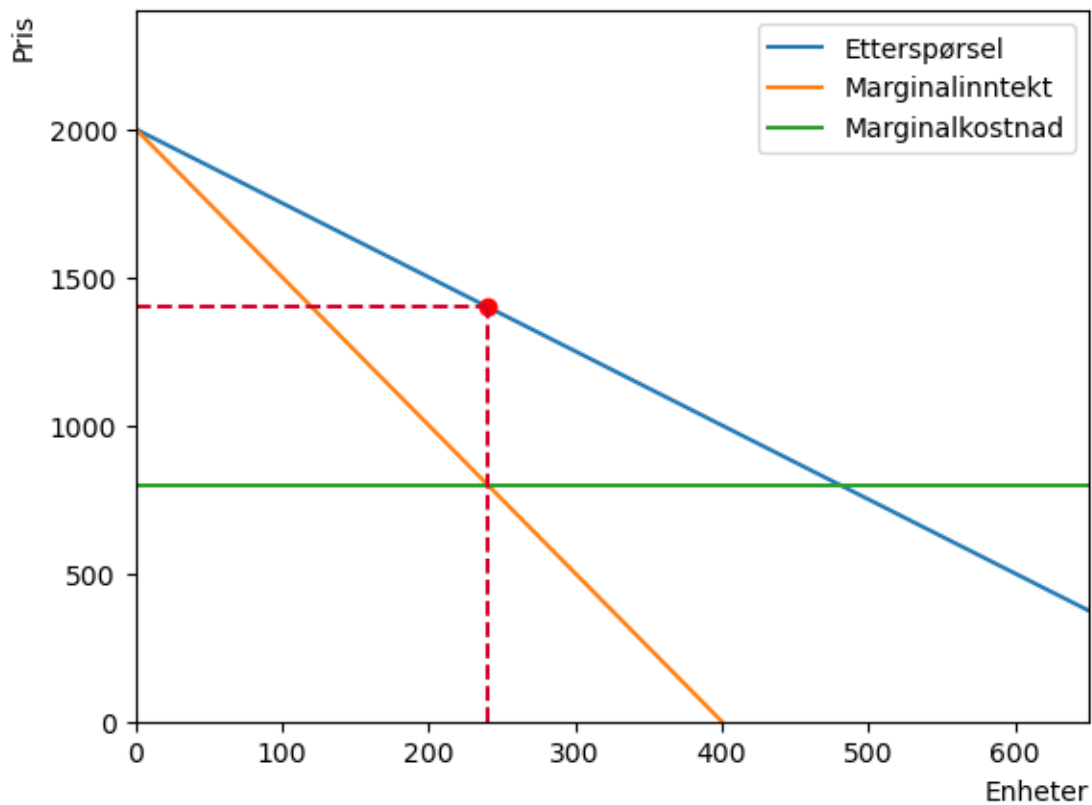
# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Avgrenser figuren.
plt.xlim(0, 650)
plt.ylim(0, 2400)

# Legger til legend.
plt.legend(loc = "upper right")

```

<matplotlib.legend.Legend at 0x7feadc4a1370>



3.2 Lønnsomheten - Matematisk

```
# Definerer etterspørselsfunksjon.
def dem_t(q):
    return 2916.66-2.08*q

# Definerer marginalinntekt.
def mr_t(q):
    return 2916.66-4.16*q

# Finner likevektspunktet.
# Setter marginalkostnaden lik marginalinntekten.
equ_t = sp.Eq(mr_t(q), 800)

# Løser med hensyn på q for å finne likevektskvantum.
q_equ_t = sp.solve(equ_t, q)[0]
```

```
# Legger likevektskvantum inn i etterspørselsfunksjonen for å finne likevektsprisen.
p_equ_t = dem_t(q_equ_t)

# Viser likevekts- pris og kvantum.
print(f"Likevekstkvantum er lik:", round(q_equ_t, 2))
print(f"Likevekstprisen er lik:", round(p_equ_t, 2))
```

Likevekstkvantum er lik: 508.81
Likevekstprisen er lik: 1858.33

```
# Legger restriksjon på modellen slik at Q = 290.
p_ny = 2916.66-2.08*290
q_ny = 290
```

3.2 Lønnsomheten - Grafisk

```
# Figur 3.3

# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_02, dem_t(q_02), label = "Etterspørsel")
plt.plot(q_02, mr_t(q_02), label = "Marginalinntekt")
plt.plot(q_02, q_x, label = "Marginalkostnad")

# Markerer likevektspunkter.
plt.plot(q_ny, p_ny, marker = "o",
         color = "red")
plt.hlines(p_ny, xmin = 0, xmax = q_ny, linestyle = "dashed",
         color = "#D2042D")
plt.vlines(q_ny, ymin = 0, ymax = p_ny, linestyle = "dashed",
         color = "#D2042D")

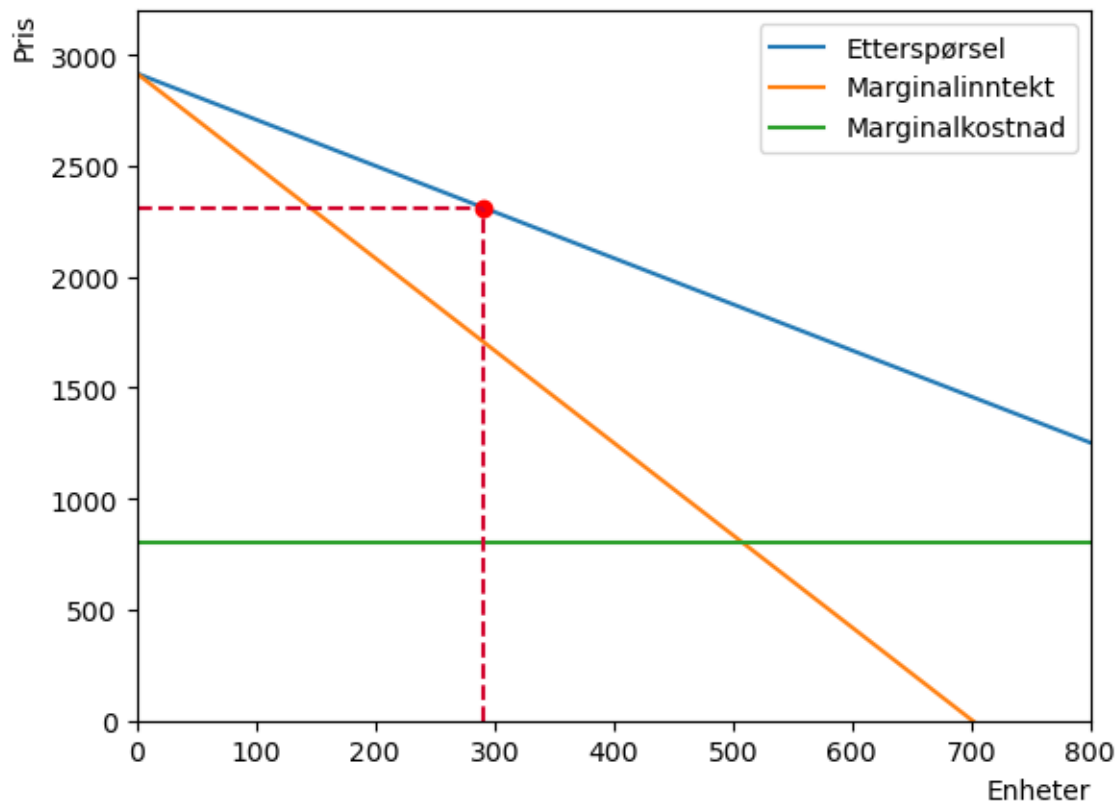
# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Avgrenser figuren.
plt.xlim(0, 800)
plt.ylim(0, 3200)

# Legger til legend.
```

```
plt.legend(loc = "upper right")
```

<matplotlib.legend.Legend at 0x7feadc41a250>



3.3 Velferd - Matematisk

```
# Finner konsumentoverskuddet.  
cs_v = (268*(7500-4150))/2  
cs_s = (240*(2000-1400))/2  
cs_t = (290*(2916.66-2313))/2
```

```
# Finner produsentoverskuddet.  
ps_v = (268*(4150-800))  
ps_s = (240*(1400-800))  
ps_t = (290*(2313-800))
```

```

# Skriver ut svaret.
print(f"Konsumentoverskuddet på vinteren er lik:", cs_v)
print(f"Konsumentoverskuddet på sommeren er lik:", cs_s)
print(f"Konsumentoverskuddet med uniform prising er lik:", cs_t)
print()
print(f"Produsentoverskuddet på vinteren er lik:", ps_v)
print(f"Produsentoverskuddet på sommeren er lik:", ps_s)
print(f"Produsentoverskuddet med uniform prising er lik:", ps_t)

```

Konsumentoverskuddet på vinteren er lik: 448900.0
 Konsumentoverskuddet på sommeren er lik: 72000.0
 Konsumentoverskuddet med uniform prising er lik: 87530.699999999998

Produsentoverskuddet på vinteren er lik: 897800
 Produsentoverskuddet på sommeren er lik: 144000
 Produsentoverskuddet med uniform prising er lik: 438770

```

# Finner velferdsoverskuddet.
ws_v = cs_v + ps_v
ws_s = cs_s + ps_s
ws_t = cs_t + ps_t

# Skriver ut svaret.
print(f"Total velferd på vinteren er lik:", ws_v)
print(f"Total velferd på sommeren er lik:", ws_s)
print(f"Total velferd med universell prising er lik:", ws_t)

```

Total velferd på vinteren er lik: 1346700.0
 Total velferd på sommeren er lik: 216000.0
 Total velferd med universell prising er lik: 526300.7

```

# Dødevektstapet.

# Setter etterspørsel lik marginalkostnad.
equ_v1 = sp.Eq(dem_v(q), 800)
equ_s1 = sp.Eq(dem_s(q), 800)
equ_t1 = sp.Eq(dem_t(q), 800)

# Løser med hensyn på q for å finne likevektskvantum.

```

```

q_equ1_v1 = sp.solve(equ_v1, q)[0]
q_equ1_s1 = sp.solve(equ_s1, q)[0]
q_equ1_t1 = sp.solve(equ_t1, q)[0]

# Legger likevektskvantum inn i etterspørselsfunksjonen for å finne likevektsprisen.
p_equ1_v1 = dem_v(q_equ1_v1)
p_equ1_s1 = dem_v(q_equ1_s1)
p_equ1_t1 = dem_v(q_equ1_t1)

# Dødevektstapet.
dl_v = ((q_equ1_v1 - 268)*(4150-800))/2
dl_s = ((q_equ1_s1 - 240)*(1400-800))/2
dl_t = ((q_equ1_t1 - 290)*(2313-800))/2

# Skriver ut svaret.
print(f"Dødevektstapet på vinteren er lik:", dl_v)
print(f"Dødevektstapet på sommeren er lik:", dl_s)
print(f"Dødevektstapet med universell prising er lik:", dl_t)

```

Dødevektstapet på vinteren er lik: 448900.0000000000
Dødevektstapet på sommeren er lik: 72000.00000000000
Dødevektstapet med universell prising er lik: 550448.3125000000

3.3 Velferd - Grafisk

```

# Figur 3.4

# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_02, dem_v(q_02), label = "Etterspørsel")
plt.plot(q_02, q_x, label = "Marginalkostnad")

# Markerer likevektspunkter.
plt.plot(equ_Qv, equ_Pv, marker = "o",
         color = "red")
plt.hlines(equ_Pv, xmin = 0, xmax = equ_Qv, linestyle = "dashed",
         color = "#D2042D")
plt.vlines(equ_Qv, ymin = 0, ymax = equ_Pv, linestyle = "dashed",
         color = "#D2042D")

# Legger inn konsument- og produsentoverskudd og dødevektstap.

```

```

plt.fill_between(q_02, dem_v(q_02), 4150, where = (q_02 <= 268),
                 color = "green", alpha = 0.4, label = "Konsumentoverskudd")
plt.fill_between(q_02, 800, 4150, where = (q_02 <= 268),
                 color = "red", alpha = 0.4, label = "Produsentoverskudd")
plt.fill_between(q_02, 800, dem_v(q_02), where = (q_02 >= 268) & (q_02 <= 540),
                 color = "blue", alpha = 0.4, label = "Dødevæktstap")

# Legger til tekst.
plt.text(30, 5000, "448 900")
plt.text(30, 2000, "897 800")
plt.text(300, 2000, "448 900")

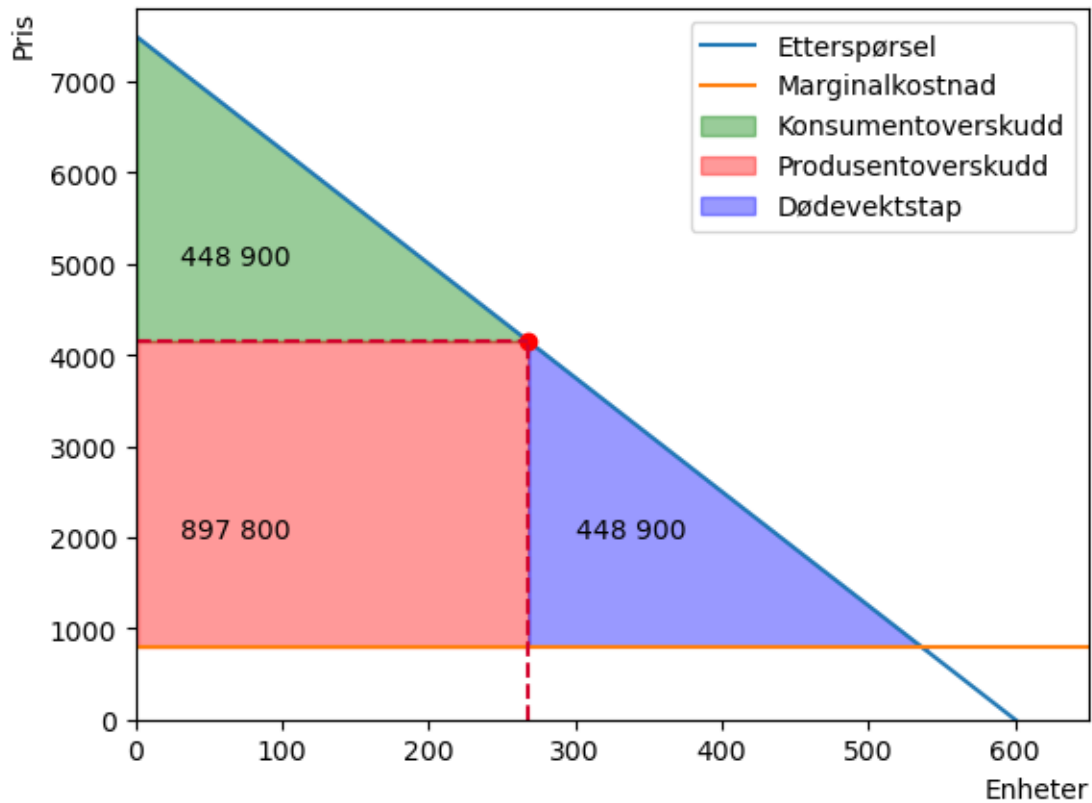
# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Avgrenser figuren.
plt.xlim(0, 650)
plt.ylim(0, 7800)

# Legger til legend.
plt.legend(loc = "upper right")

```

<matplotlib.legend.Legend at 0x7feadc34beb0>



Figur 3.5

```
# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_02, dem_s(q_02), label = "Etterspørsel")
plt.plot(q_02, q_x, label = "Marginalkostnad")

# Markerer likevektspunkter.
plt.plot(equ_Qs, equ_Ps, marker = "o",
         color = "red")
plt.hlines(equ_Ps, xmin = 0, xmax = equ_Qs, linestyle = "dashed",
         color = "#D2042D")
plt.vlines(equ_Qs, ymin = 0, ymax = equ_Ps, linestyle = "dashed",
         color = "#D2042D")

# Legger inn konsument- og produsentoverskudd og dødevektstap.
plt.fill_between(q_02, dem_s(q_02), 1400, where = (q_02 <= 240),
                color = "green", alpha = 0.4, label = "Konsumentoverskudd")
```



```

plt.fill_between(q_02, 800, 1400, where = (q_02 <= 240),
                 color = "red", alpha = 0.4, label = "Produsentoverskudd")
plt.fill_between(q_02, 800, dem_s(q_02), where = (q_02 >= 240) & (q_02 <= 485),
                 color = "blue", alpha = 0.4, label = "Dødevektstap")

# Legger til tekst.
plt.text(30, 1600, "72 000")
plt.text(30, 1000, "144 000")
plt.text(260, 1000, "72 000")

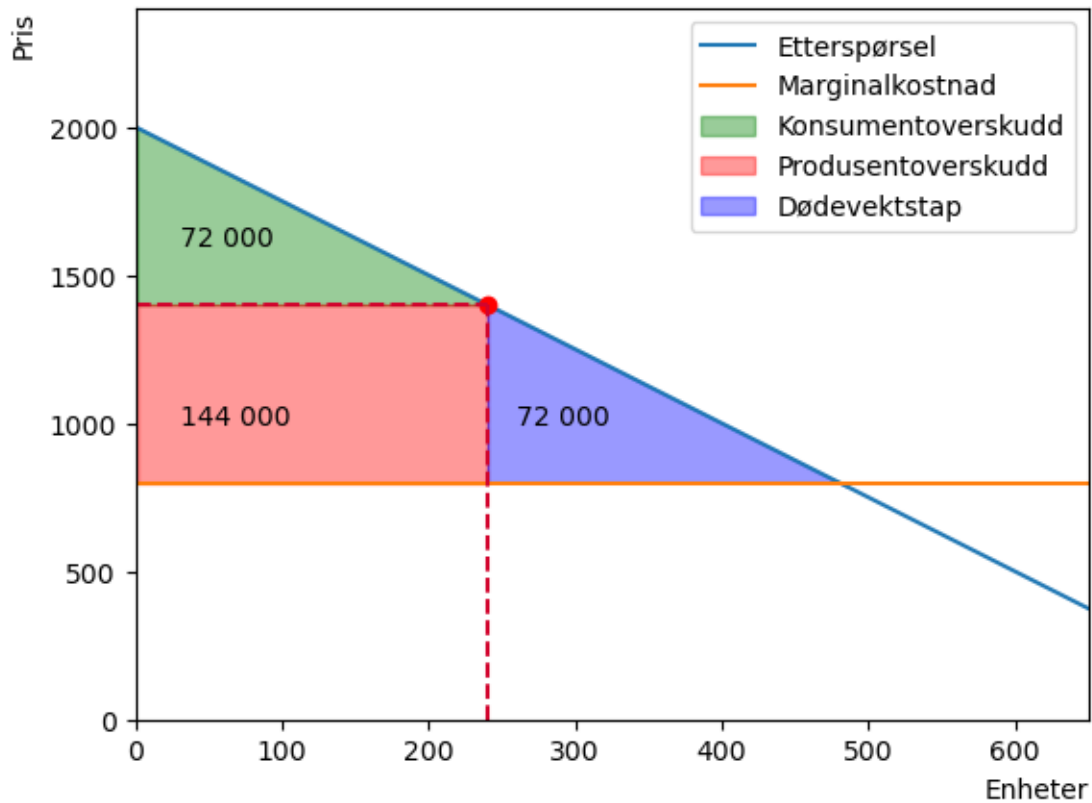
# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Avgrenser figuren.
plt.xlim(0, 650)
plt.ylim(0, 2400)

# Legger til legend.
plt.legend(loc = "upper right")

```

<matplotlib.legend.Legend at 0x7feadc323c70>



Figur 3.6

```
# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_02, dem_t(q_02), label = "Etterspørsel")
plt.plot(q_02, q_x, label = "Marginalkostnad")

# Markerer likevektspunkter.
plt.plot(q_ny, p_ny, marker = "o",
         color = "red")
plt.hlines(p_ny, xmin = 0, xmax = q_ny, linestyle = "dashed",
         color = "#D2042D")
plt.vlines(q_ny, ymin = 0, ymax = p_ny, linestyle = "dashed",
         color = "#D2042D")

# Legger inn konsument- og produsentoverskudd og dødeveststap.
plt.fill_between(q_02, dem_t(q_02), 2313, where = (q_02 <= 290),
                color = "green", alpha = 0.4, label = "Konsumentoverskudd")
```

```

plt.fill_between(q_02, 800, 2313, where = (q_02 <= 290),
                 color = "red", alpha = 0.4, label = "Produsentoverskudd")
plt.fill_between(q_02, 800, dem_t(q_02), where = (q_02 >= 290) & (q_02 <= 800),
                 color = "blue", alpha = 0.4, label = "Dødevektstap")

# Legger til tekst.
plt.text(30, 2500, "87 530")
plt.text(30, 1000, "438 770")
plt.text(320, 1000, "550 448")

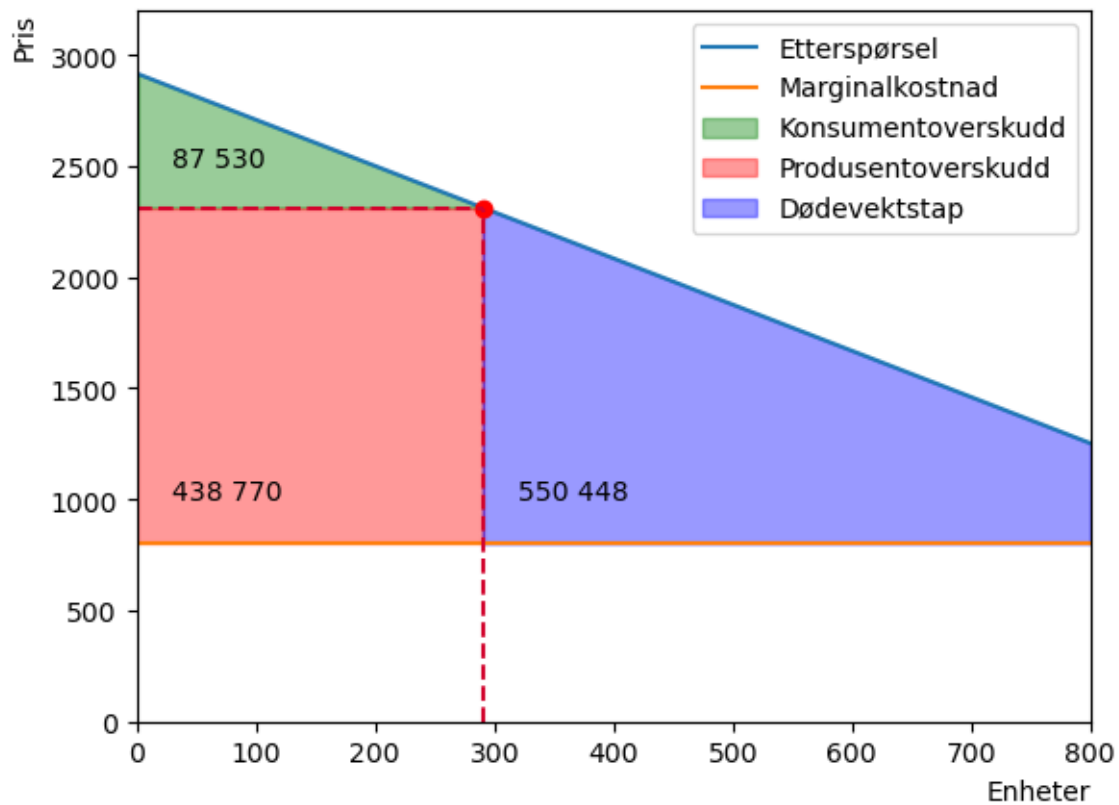
# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Avgrenser figuren.
plt.xlim(0, 800)
plt.ylim(0, 3200)

# Legger til legend.
plt.legend(loc = "upper right")

```

<matplotlib.legend.Legend at 0x7feadcdf3fd0>



3.4 Økning i etterspørsel - Matematisk

```
# Definerer ny etterspørsel vinter.
def dem_v_ny(Qv):
    return 8500-12.5*Qv

# Definerer ny etterspørsel sommer.
def dem_s_ny(Qs):
    return 3200-2.5*Qs

# Definerer marginalinntekt vinter, utregning i besvarelse.
def mr_v_ny(Qv):
    return 8500-25*Qv

# Definerer marginalinntekt sommer, utregning i besvarelse.
def mr_s_ny(Qs):
    return 3200-5*Qs
```

```

# Setter marginalinntekt vinter lik marginalkostnad (800).
equ_mrv_ny = sp.Eq(mr_v_ny(Qv), 800)

# Setter marginalinntekt sommer lik marginalkostnad (800)
equ_mrs_ny = sp.Eq(mr_s_ny(Qs), 800)

# Løser med hensyn på Q.
equ_Qv_ny = sp.solve(equ_mrv_ny)[0]
equ_Qs_ny = sp.solve(equ_mrs_ny)[0]

# Viser likevektskvantum.
print(f"Likevektskvantum på vinter er lik:", equ_Qv_ny)
print(f"Likevektskvantum på sommeren er lik:", equ_Qs_ny)

```

Likevektskvantum på vinter er lik: 308
Likevektskvantum på sommeren er lik: 480

```

# Finner likevekstprisen.
equ_Pv_ny = dem_v_ny(equ_Qv_ny)
equ_Ps_ny = dem_s_ny(equ_Qs_ny)

# Viser likevekstprisen.
print(f"Likevekstprisen på vinteren er lik:", equ_Pv_ny)
print(f"Likevekstprisen på sommeren er lik:", equ_Ps_ny)

```

Likevekstprisen på vinteren er lik: 4650.000000000000
Likevekstprisen på sommeren er lik: 2000.000000000000

```

# Finner profitten til hotellet.
profit_v_ny = (equ_Pv_ny-800)*equ_Qv_ny
profit_s_ny = (equ_Ps_ny-800)*equ_Qs_ny

# Viser profitten til hotellet.
print(f"Dekningsbidraget til hotellet per døgn på vinteren er lik:", profit_v_ny)
print(f"Dekningsbidraget til hotellet per døgn på sommeren er lik:", profit_s_ny)

```

Dekningsbidraget til hotellet per døgn på vinteren er lik: 1185800.00000000
Dekningsbidraget til hotellet per døgn på sommeren er lik: 576000.000000000

```

# Tvinger modellen til Q = 290.
equ_Pv_ny_290 = dem_v_ny(290)
equ_Ps_ny_290 = dem_s_ny(290)

# Viser likevekstprisen.
print(f"Likevekstprisen på vinteren med Q = 290 er lik:", equ_Pv_ny_290)
print(f"Likevekstprisen på sommeren med Q = 290 er lik:", equ_Ps_ny_290)

```

Likevekstprisen på vinteren med Q = 290 er lik: 4875.0

Likevekstprisen på sommeren med Q = 290 er lik: 2475.0

3.4 Økning i etterspørsel - Grafisk

```

# Figur 3.7
# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_02, dem_v(q_02), label = "Etterspørsel", alpha = 0.3)
plt.plot(q_02, mr_v(q_02), label = "Marginalinntekt", alpha = 0.3)
plt.plot(q_02, q_x, label = "Marginalkostnad")
plt.plot(q_02, dem_v_ny(q_02), label = "Ny Etterspørsel", color = "purple")
plt.plot(q_02, mr_v_ny(q_02), label = "Ny Marginalinntekt")

# Markerer likevektspunkter.
plt.plot(equ_Qv, equ_Pv, marker = "o",
         color = "red", alpha = 0.3)
plt.plot(290, equ_Pv_ny_290, marker = "o",
         color = "red")
plt.hlines(equ_Pv, xmin = 0, xmax = equ_Qv, linestyle = "dashed",
          color = "#D2042D", alpha = 0.3)
plt.hlines(equ_Pv_ny_290, xmin = 0, xmax = 290, linestyle = "dashed",
          color = "#D2042D")
plt.vlines(equ_Qv, ymin = 0, ymax = equ_Pv, linestyle = "dashed",
          color = "#D2042D", alpha = 0.3)
plt.vlines(290, ymin = 0, ymax = equ_Pv_ny_290, linestyle = "dashed",
          color = "#D2042D")

# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

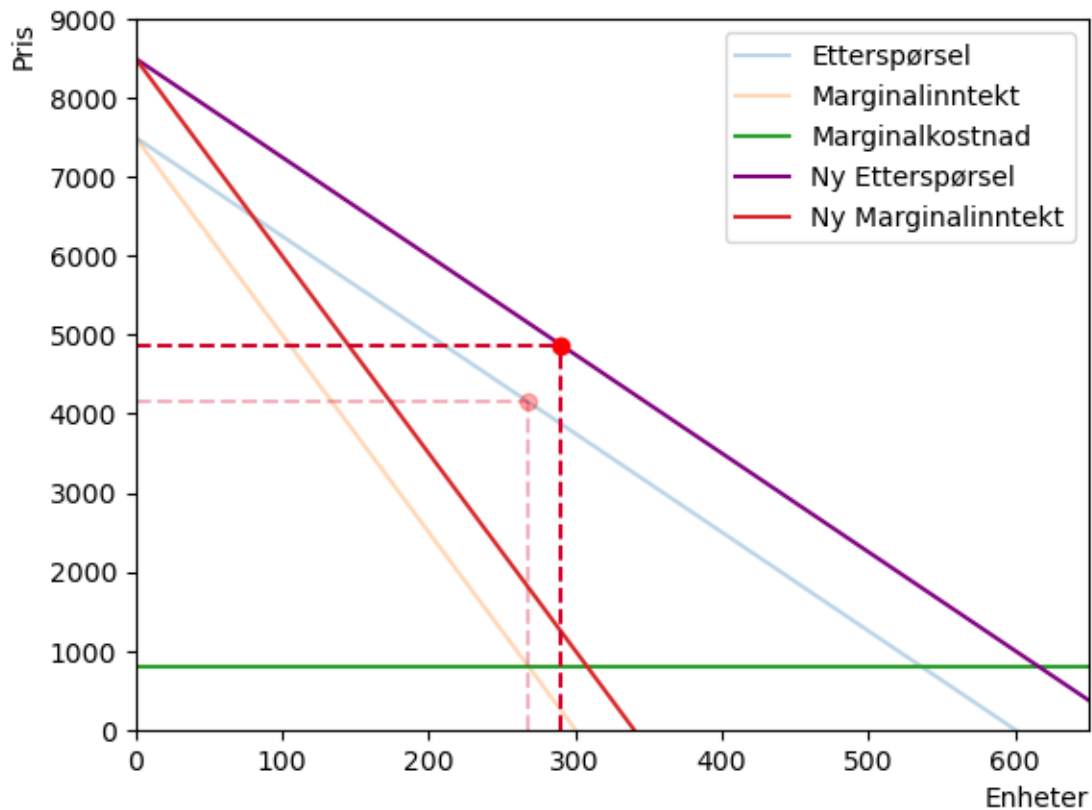
# Avgrenser figuren.

```

```
plt.xlim(0, 650)
plt.ylim(0, 9000)

# Legger til legend.
plt.legend(loc = "upper right")
```

<matplotlib.legend.Legend at 0x7feadc2fd550>



```
# Figur 3.8
# Plotter etterspørsel, marginal- kostnad og inntekt.
plt.plot(q_02, dem_s(q_02), label = "Etterspørsel")
plt.plot(q_02, mr_s(q_02), label = "Marginalinntekt")
plt.plot(q_02, q_x, label = "Marginalkostnad")
plt.plot(q_02, dem_s_ny(q_02), label = "Ny Etterspørsel", color = "purple")
plt.plot(q_02, mr_s_ny(q_02), label = "Ny Marginalinntekt")
```

```

# Markerer likevektspunkter.
plt.plot(equ_Qs, equ_Ps, marker = "o",
         color = "red", alpha = 0.3)
plt.plot(290, equ_Ps_ny_290, marker = "o",
         color = "red")
plt.hlines(equ_Ps, xmin = 0, xmax = equ_Qs, linestyle = "dashed",
          color = "#D2042D", alpha = 0.3)
plt.hlines(equ_Ps_ny_290, xmin = 0, xmax = 290, linestyle = "dashed",
          color = "#D2042D")
plt.vlines(equ_Qs, ymin = 0, ymax = equ_Ps, linestyle = "dashed",
          color = "#D2042D", alpha = 0.3)
plt.vlines(290, ymin = 0, ymax = equ_Ps_ny_290, linestyle = "dashed",
          color = "#D2042D")

# Gir aksene titler.
plt.ylabel("Pris", loc = "top")
plt.xlabel("Enheter", loc = "right")

# Avgrenser figuren.
plt.xlim(0, 800)
plt.ylim(0, 3500)

# Legger til legend.
plt.legend(loc = "upper right")

```

<matplotlib.legend.Legend at 0x7fead6d11fd0>

