Utfordring 3.1

In [5]:
```
import sympy as sp
import numpy as np
from matplotlib import pyplot as plt

# symboler jeg kan få bruk for:

C,c,l,L,a,b,u,u_0,w,z,s,q = sp.symbols('C c l L a b u u_0 w z s q')
```

In [6]:
```
# Stønad er lik 0

# definerer nyttefunksjonen:
def nytte(c,l):
    return c**0.5*l**0.5

# Setter inn i funksjonen der Tone ikke mottar stønad. Det vil si konsum er lik M o
print(f'Finner nyttenivå:')
display(nytte(100,60))

eq = sp.Eq((100+w*40)**0.5*(60-40)**0.5,77.45)

eq_sol = sp.solve(eq,w)

print('Finner reservasjonslønn:')
eq_sol
```

```
Finner nyttenivå:
77.45966692414834
Finner reservasjonslønn:
```

Out[6]: [4.99812812500000]

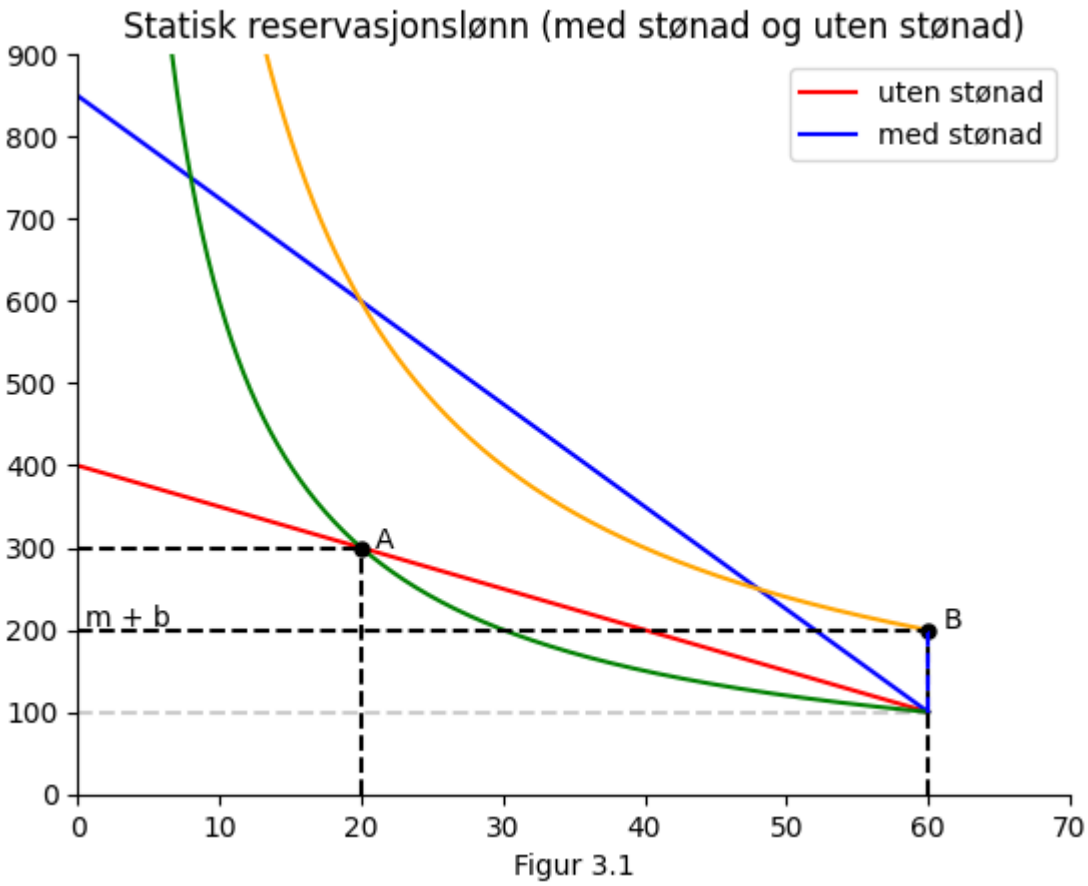
In [7]:
```
# Stønad er lik 100

# Legger til 100 i konsum som følge av tilgang til stønad.
print(f'Finner nyttenivå:')
display(nytte(200,60))

# Setter nyttenivå inn i ligning og løser for w
eq2 = sp.Eq((100+w*40)**0.5*(60-40)**0.5,109.5)

eq2_sol = sp.solve(eq2,w)

print(f'Finner reservasjonslønn:')
eq2_sol
```

```
Finner nyttenivå:
109.54451150103323
Finner reservasjonslønn:
```

Out[7]: [12.4878125000000]

In [8]:
```
L_0 = np.linspace(0.001,60,1000)
```

```
fig , ax = plt.subplots()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.set_xlim(0,70)
ax.set_ylim(0,900)

# Budsjett med støtte
budsjett = sp.Eq(100+5*(60-L),c)

def budsjett(L):
    return 400 - 5*L

# Budsjett uten støtte
budsjett2 = sp.Eq(100 + 12.5*(60-L),c)

def budsjett2(L):
    return 850 - 12.5*L

ax.plot(L_0,budsjett(L_0), label = 'uten stønad', color = 'red')
ax.plot(L_0,budsjett2(L_0), label = 'med stønad', color = 'blue')
ax.legend(loc = 'best')
ax.set_xlabel('Figur 3.1')

u = C**a*L**b

# Løser med hensyn på L
L_ind_sol=sp.solve(u-u_0,L)[0]
L_ind_sol

# Må gjøre om for å plotte
indiff_L=sp.lambdify((u_0,a,b,C), L_ind_sol)
indiff_L(u_0,a,b,C)
# Plotter
ax.plot(L_0, indiff_L(77.5,0.5,0.5,L_0), color = 'green')
ax.plot(L_0, indiff_L(109.5,0.5,0.5,L_0), color = 'orange')

# Plot notasjon
ax.hlines(100,0,60, color='black',ls='dashed', alpha = 0.2)

ax.plot(60,200, marker  ='o', color= 'black', markersize = 5)
ax.vlines(60,0,200, color='black',ls='dashed')
ax.vlines(60,100,200, color='blue',ls='dashed')

ax.hlines(200,0,60, color='black',ls='dashed')
ax.annotate('B', (60+1,200+1))
ax.annotate('m + b', (0+0.5,200+5))

ax.vlines(20,0,300, color='black',ls='dashed')
ax.hlines(300,0,20, color='black',ls='dashed')
ax.annotate('A',(20+1,300))
ax.plot(20,300, marker  ='o', color= 'black', markersize = 5)

ax.set_title('Statisk reservasjonslønn (med stønad og uten stønad)');
```



Figur 3.1

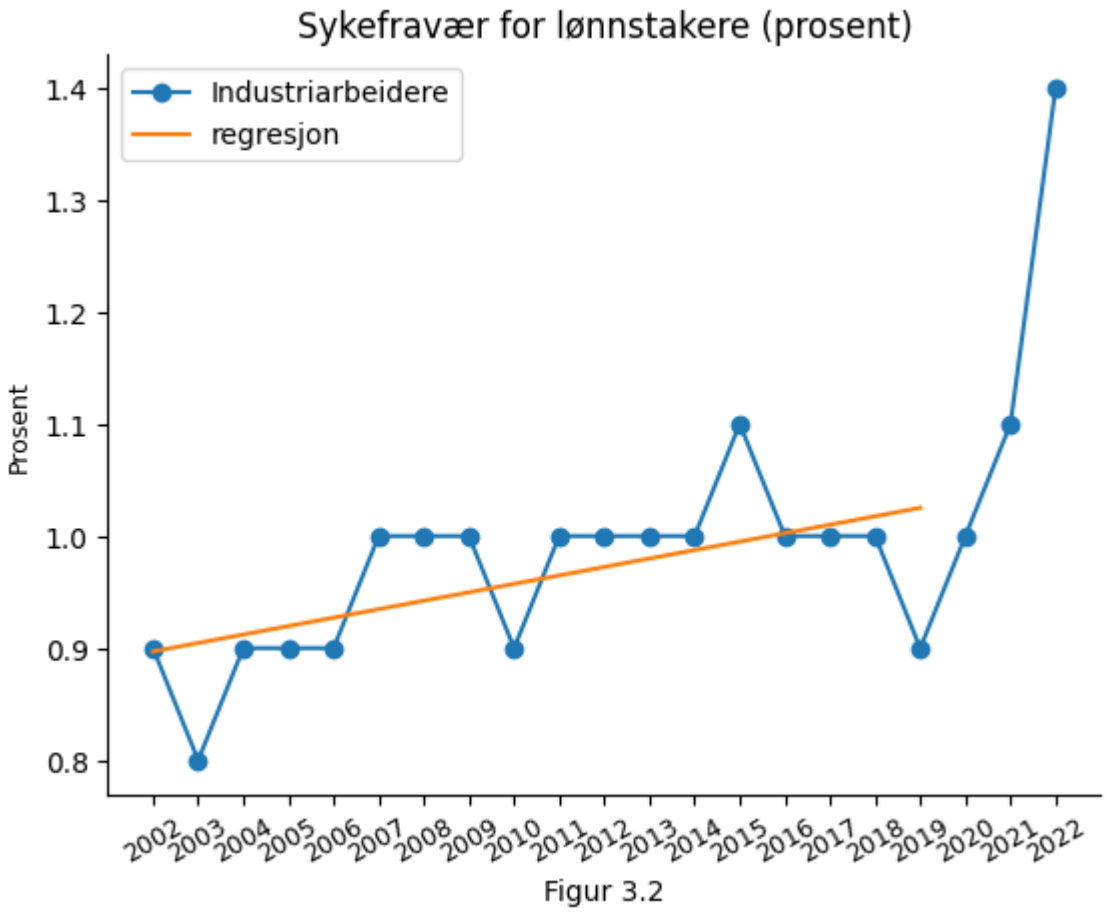
Utfordring 3.2

a)

In [9]:
```
import requests
from pyjstat import pyjstat
from sklearn.linear_model import LinearRegression
import pandas as pd
import warnings
warnings.filterwarnings('ignore')

postUrl = 'https://data.ssb.no/api/v0/no/table/12441/'

apiQuery = {
  "query": [
    {
      "code": "Kjonn",
      "selection": {
        "filter": "item",
        "values": [
          "0"
        ]
      }
    },
    {
      "code": "NACE2007",
      "selection": {
        "filter": "item",
```

```
        "values": [
          "10-33"
        ]
      }
    },
    {
      "code": "Sykefraver2",
      "selection": {
        "filter": "item",
        "values": [
          "E"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
          "2002",
          "2003",
          "2004",
          "2005",
          "2006",
          "2007",
          "2008",
          "2009",
          "2010",
          "2011",
          "2012",
          "2013",
          "2014",
          "2015",
          "2016",
          "2017",
          "2018",
          "2019",
          "2020",
          "2021",
          "2022"
        ]
      }
    }
  ],
  "response": {
    "format": "json-stat2"
  }
}

def apiToDataframe(postUrl, query):

    # postUrl som spørringen skal postes mot
    # Spørringen og endepunktet til API-et kan hentes fra Statistikkbanken.

    res = requests.post(postUrl, json=query)
    # legger resultat i ds. DS har i tillegg en del metadata
```

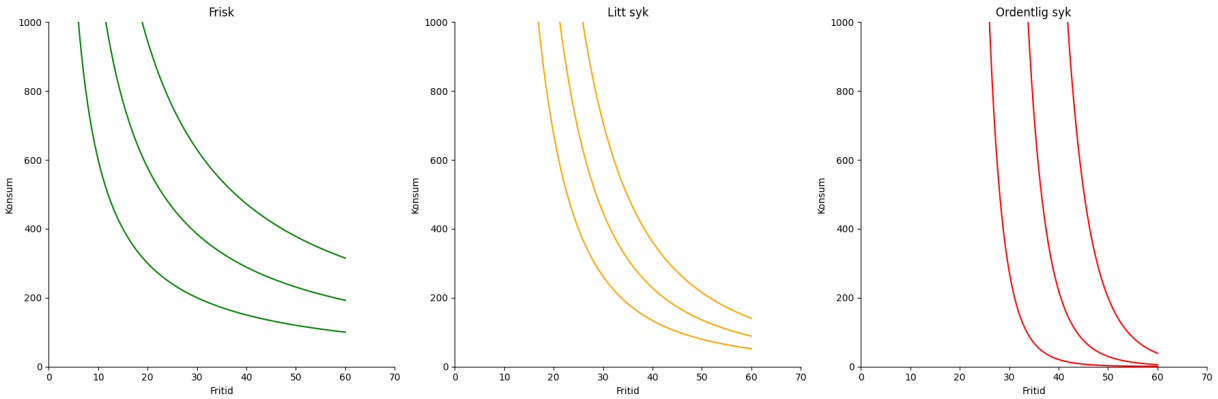
```
    ds = pyjstat.Dataset.read(res.text)
    # skriver resultatet til to dataframes
    # først dataframe med tekst
    df = ds.write('dataframe')
    # deretter dataframe med koder
    df_id = ds.write('dataframe', naming='id')
    # returnerer også ds i tilfelle en trenger metadata
    return df, df_id, ds

egenmelding, df_id, ds = apiToDataframe(postUrl, apiQuery)

fig ,ax = plt.subplots()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_tick_params(labelsize=9, rotation = 30)
ax.set_title('Sykefravær for lønnstakere (prosent)')
ax.set_ylabel('Prosent', size = 9)
ax.set_xlabel('Figur 3.2')
ax.plot(egenmelding['år'],egenmelding['value'], marker = 'o', label = 'Industriarbe

egenmelding2 = egenmelding[egenmelding['år'] < '2020']

# lager regresjon
egenmelding2['år'] = np.arange(len(egenmelding2.index))
X = egenmelding2.loc[:, ['år']]
y = egenmelding2.loc[:, 'value']
model = LinearRegression()
model.fit(X, y)
y_pred = pd.Series(model.predict(X), index=X.index)

ax.plot(egenmelding2['år'],y_pred, label = 'regresjon')
ax.legend(loc = 'best');
```



Figur 3.2

In [10]:
```python
# Nyttefunksjon (CB funksjon)
def I_1(l_1):
    return ((77.45) / (l_1**0.5))**(1/0.5)

def I_2(l_1):
    return (((107.45) / (l_1**0.5))**(1/0.5))

def I_3(l_1):
    return (((137.45) / (l_1**0.5))**(1/0.5))

def I_4(l_1):
    return ((77.45) / (l_1**0.7))**(1/0.3)

def I_5(l_1):
    return ((67.45) / (l_1**0.7))**(1/0.3)

def I_6(l_1):
    return ((57.45) / (l_1**0.7))**(1/0.3)

def I_7(l_1):
    return ((37.45) / (l_1**0.9))**(1/0.1)

def I_8(l_1):
    return ((47.45) / (l_1**0.9))**(1/0.1)

def I_9(l_1):
    return ((57.45) / (l_1**0.9))**(1/0.1)
```

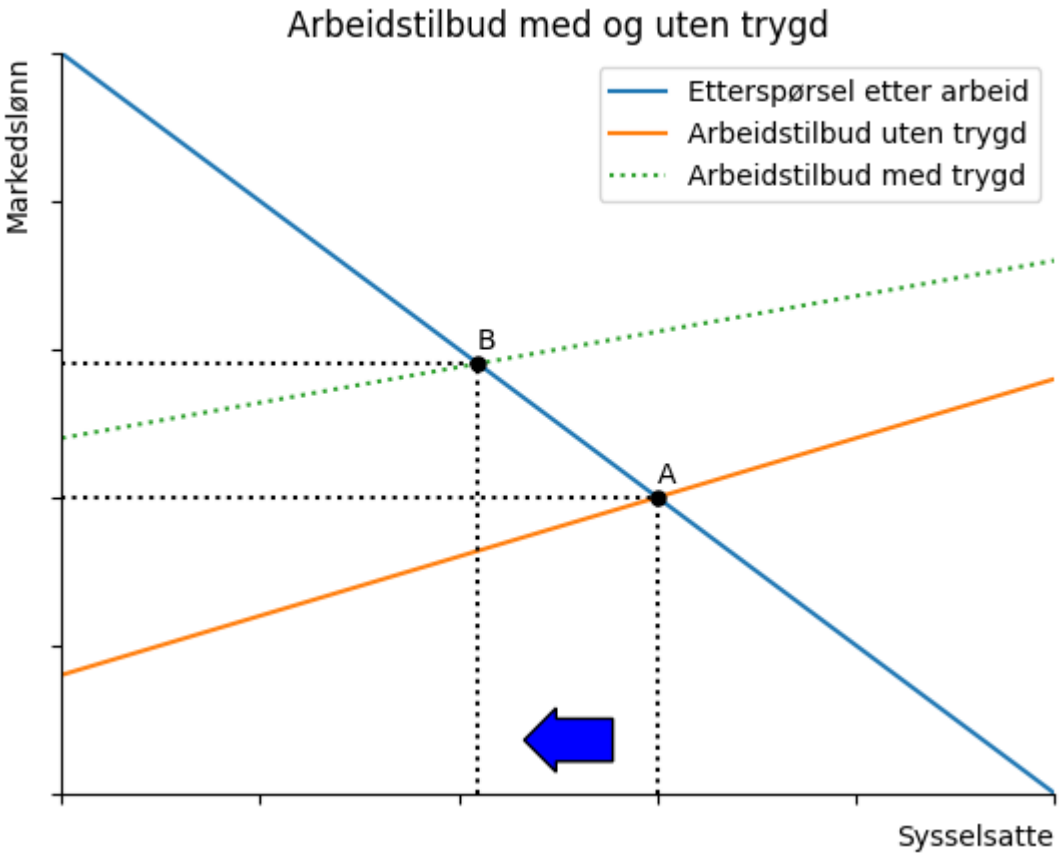
```python
l_1 = np.linspace(0.01, 60, 1000)

# Lager graf
fig, axes = plt.subplots(1, 3, figsize=(18, 6))  # 1 rad med 3 kolonner

# Plot for frisk
axes[0].plot(l_1, I_1(l_1), color="green")
axes[0].plot(l_1, I_2(l_1), color="green")
axes[0].plot(l_1, I_3(l_1), color="green")
axes[0].set_title("Frisk")
axes[0].set_xlim(0, 70)
axes[0].set_ylim(0, 1000)
axes[0].set_xlabel("Fritid")
axes[0].set_ylabel("Konsum")
axes[0].spines['top'].set_color('none')
axes[0].spines['right'].set_color('none')

# Plot for litt syk
axes[1].plot(l_1, I_4(l_1), color="orange")
axes[1].plot(l_1, I_5(l_1), color="orange")
axes[1].plot(l_1, I_6(l_1), color="orange")
axes[1].set_title("Litt syk")
axes[1].set_xlim(0, 70)
axes[1].set_ylim(0, 1000)
axes[1].set_xlabel("Fritid")
axes[1].set_ylabel("Konsum")
axes[1].spines['top'].set_color('none')
axes[1].spines['right'].set_color('none')

# Plot for ordentlig syk
axes[2].plot(l_1, I_7(l_1), color="red")
axes[2].plot(l_1, I_8(l_1), color="red")
axes[2].plot(l_1, I_9(l_1), color="red")
axes[2].set_title("Ordentlig syk")
axes[2].set_xlim(0, 70)
axes[2].set_ylim(0, 1000)
axes[2].set_xlabel("Fritid")
axes[2].set_ylabel("Konsum")
axes[2].spines['top'].set_color('none')
axes[2].spines['right'].set_color('none')

plt.tight_layout()
plt.show()
```



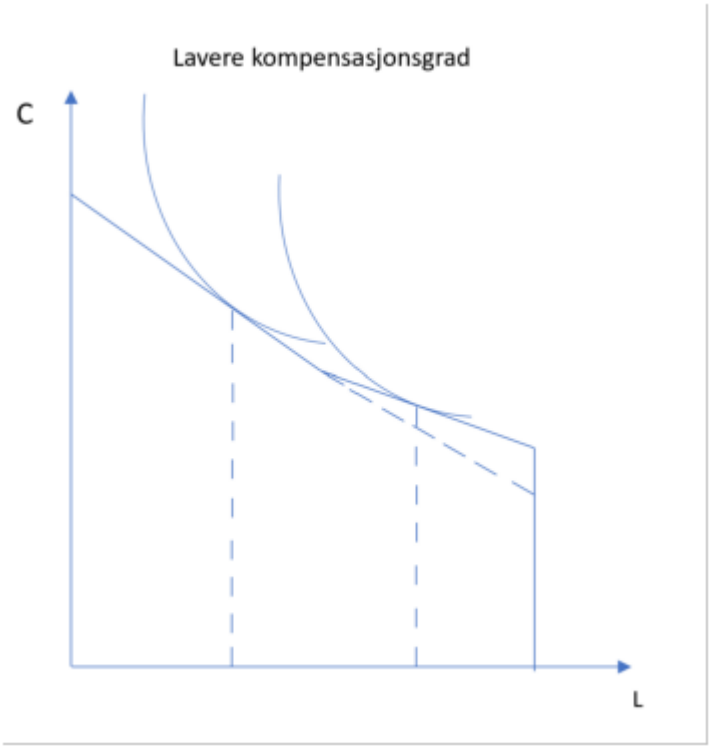
In [11]:

```
fig, ax = plt.subplots()
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.set_xlim(0,50)
ax.set_ylim(0,250)

def etterspørsel(x):
    return 250 - 5*x

def tilbud(x):
    return 40 + 2*x

def tilbud2(x):
    return 120 + 1.2*x

x = np.linspace(0.001,50,1000)

ax.plot(x,etterspørsel(x), label = 'Etterspørsel etter arbeid')
ax.plot(x,tilbud(x), label = 'Arbeidstilbud uten trygd')
ax.plot(x,tilbud2(x), linestyle = 'dotted', label = 'Arbeidstilbud med trygd')
ax.legend(loc = 'best')
ax.set_title('Arbeidstilbud med og uten trygd')
ax.set_xlabel('Sysselsatte', loc = 'right')
ax.set_ylabel('Markedslønn', loc = 'top')

# løser likevekt for hånd

ax.vlines(30,0,100, color = 'black', ls = 'dotted')
ax.vlines(20.9,0,145, color = 'black', ls = 'dotted')
ax.hlines(145,0,20.9, color = 'black', ls = 'dotted')
ax.hlines(100,0,30, color = 'black', ls = 'dotted')

ax.set_xticklabels([])
ax.set_yticklabels([])

ax.annotate('A',(30,100+5))
ax.plot(30,100, marker  ='o', color= 'black', markersize = 5)

ax.annotate('B',(20.9,145+5))
ax.plot(20.9,145, marker  ='o', color= 'black', markersize = 5)

t = ax.text(26, 18, "      ",
            ha="center", va="center", rotation=540, size=12,
```

```
            bbox=dict(boxstyle="rarrow,pad=0.1",
                      fc="blue"));
```



In [55]:
```
import matplotlib.image as mpimg
import matplotlib.pyplot as plt

# Read Images
img = mpimg.imread('Kompensasjonsgrad.png')

# Output Images
plt.imshow(img)
plt.box()
plt.axis('off');
```



Appendiks

Jeg har ikke brukt KI i besvarelsen min.

Kildeliste

Boeri, T. & van Ours, J. The Economics of Imperfect Labor Markets, Third Edition, Princeton.

Videoforelesning, Andrea Mannberg (2022). SOK 2008: Arbeidsledighetsrate. Link: https://www.youtube.com/watch?v=PIK38cpDgOY&ab_channel=AndreaMannberg

Forelesningsnotater, Mikko Moilanen (2023). SOK 2008: Disinsentiveffekter. Link: https://uit-sok-2008-h23.github.io/assets/kap_6_insentiv_studenter.html

Forelesningsnotater, Andrea Mannberg (2023). SOK 2008: Arbeidsledighetstrygd. Link: https://uit-sok-2008-h23.github.io/assets/F4.3_Arbeidsledighetstrygd_2023.pdf