

# | Mappeoppgave 1 |

## | 1. Innledning |

Næringslivets Hovedorganisasjon (NHO) sier på sine nettsider:

---

"En turistskatt er en avgift som skal bidra til å finansiere fellesgodene i reiselivet. Et fellesgode kan ikke gjøres eksklusivt for den som betaler for det. Eksempel på fellesgoder er offentlige toaletter, skiløyper og merkede stier."

"Turistskatt er et bestemt beløp eller en prosentandel som legges på som en ekstrakostnad på overnatting, eventuelt også andre reiselivstjenester." [1]

---

En turistskatt, eller ett besøksbidrag som regjeringen sier, skal som sagt være med på å finansiere fellesgodene i reiselivet, som bedriftene som belager seg på norsk natur er avhengige av.

Den kan også benyttes som ett redskap for å forhindre overturisme som kan skade norsk natur, og dermed hjelpe til å sikre bærekraftig turisme i fremtiden, samt bevare norsk natur og historiske steder for alle oss som bor her, og kommer til å bo her i fremtiden.

Turisme kan skade norsk natur med forsøpling, og på andre måter. Ett eksempel er brygga i Bergen, hvor turister piller trebiter av vernede bygninger, og urinerer på treverket slik at det råtner. [2]

I EU benytter 18 av de 27 medlemsstatene seg av turistskatt, eller "occupancy taxes", og er gjerne innhentet per person, per natt eller som en prosentandel av hotellromsprisen. [3]

De varierer mellom 0.10 Euro på det laveste, til 7.50 Euro på det høyeste, og man kan se at dette utgjør en relativt liten del av romkostnadene på hoteller.

Disse settes på lokalt nivå (By, kommune, provins), og innhentes av overnattingsstedene.

I Bhutan har de nylig åpnet igjen for turisme, etter COVID-19. Tidligere hadde de en turistskatt på \$65 per natt, og ble organisert gjennom statsanksjonerte reiseoperatører med krav om en guide.

Nå åpner de med en økning i denne skatten til \$200 per natt, men uten krav om guide og man kan booke direkte med hoteller selv og oppholde seg i landet på ett 90 dagers visa.

---

"We are targeting mindful travelers who are sensitive to Bhutan's culture, environment and aspirations," said Dorji Dradhul, director general of the Tourism Council of Bhutan, the industry's regulating agency, in a TV interview on Monday. [4]

Bhutan har lenge beskattet turisme høyt, nettopp for å bevare naturen og skjerme befolkningen fra overturisme. Det skal også nevnes at de har andre tiltak som minimumsgrad av hvor mye av landmassen som skal være skog.

Det anses fremdeles som en test, men om man tar en titt på hvor vakkert landet er, kombinert med hvor lite og utsatt for potensiell overturisme det er, kan man intuitivt sett se en korrelasjon.

## | I denne utredningen skal vi se på |

| Hotellovernattinger i Norge |

| Hotellovernattinger i Tromsø |

| Hotellovernattinger i Vestre-Agder |

| Effekten en skatt kan ha på overnattinger |

| Priselastisitet for overnattinger |

## | 2. Turistnæringen i Norge |

```
In [ ]: # Last inn pakker

import numpy as np
from matplotlib import pyplot as plt
import sympy as sp
import json
import pandas as pd
import requests
from pyjstat import pyjstat

# JSON spørring for dataene

postUrl = "https://data.ssb.no/api/v0/no/table/08403/"

apiQuery = {
    "query": [
        {
            "code": "Region",
            "selection": {
                "filter": "vs:Landet",
                "values": [
                    "0"
                ]
            }
        ]
    ]
}
```

```

    },
    {
        "code": "HotellFormal",
        "selection": {
            "filter": "item",
            "values": [
                "00"
            ]
        }
    },
    {
        "code": "ContentsCode",
        "selection": {
            "filter": "item",
            "values": [
                "Overnattinger"
            ]
        }
    }
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df, df_id, ds
df, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endringer på dataframe

df[['År', 'Måned']] = df.måned.str.split("M", expand=True)

df.rename(columns = {'måned': 'år_måned'}, inplace = True)

```

```

In [ ]: # Lag figuren

m=1000000

fig, ax = plt.subplots(figsize=(18, 6))

# Plot data
ax.plot(df['år_måned'], df['value']/m, color='black', marker='o', linewidth=0.75)

# Sett opp årstall på x-aksen
tick_val = np.arange(0, 456, 12)
tick_lab = [x for x in range(1986, 2024)]

plt.xticks(tick_val, tick_lab, fontsize=8)

# Gi navn til aksene

```

```

ax.set_xlabel('År', fontsize=12)
ax.set_ylabel('Overnattinger (mill)', fontsize=12)

# Sett opp tittel
ax.set_title('Overnattinger i Norge, per måned fra 1986 til 2023', fontsize=16)

# Sett opp grenser på aksene
ax.set_xlim(0, 444)

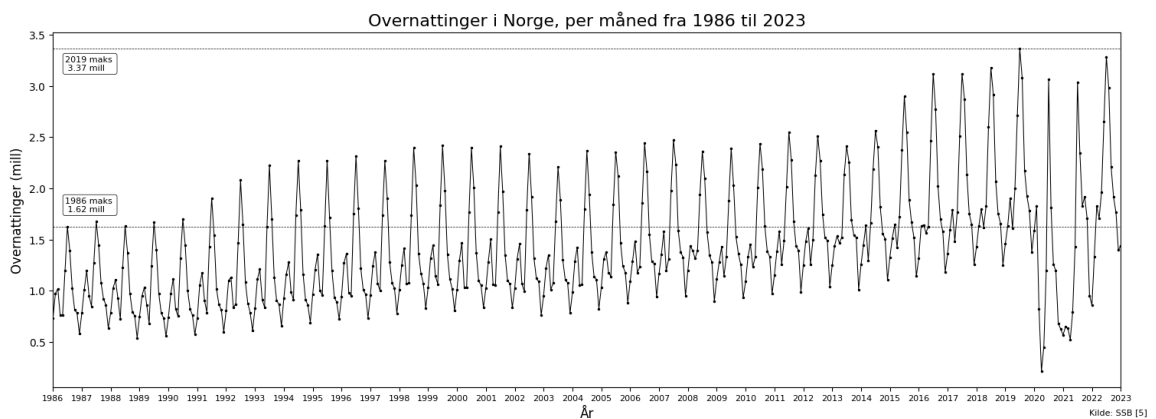
# Vis maksimumsverdier på grafen
ax.hlines(3.367177, 0, 444, color='black', linewidth=0.5, linestyle='--')
ax.text(5, 3.15, '2019 maks \n 3.37 mill', fontsize=8, bbox=dict(facecolor='white'))

ax.hlines(1.627225, 0, 444, color='black', linewidth=0.5, linestyle='--')
ax.text(5, 1.77, '1986 maks \n 1.62 mill', fontsize=8, bbox=dict(facecolor='white'))

# Kilde
plt.annotate('Kilde: SSB [5]', (0,0), (975,-20), fontsize=8, xycoords='axes fraction')

plt.show()

```



```

In [ ]: # Lag ny dataframe med summen av overnattinger per år
df_1 = df[['År', 'value']]
df_1 = df_1.groupby('År').sum()
df_1 = df_1.reset_index()

```

```

In [ ]: # Lag figuren
fig2, ax = plt.subplots(figsize=(18, 6))

# Plot data
ax.plot(df_1['År'], df_1['value']/m, color='black', marker='o', linewidth=0.75,

# Sett opp årstall på x-aksen
tick_val = np.arange(0, 38, 1)
tick_lab = [x for x in range(1986, 2024)]

plt.xticks(tick_val, tick_lab, fontsize=8)

# Gi navn til aksene
ax.set_xlabel('År', fontsize=12)
ax.set_ylabel('Overnattinger (mill)', fontsize=12)

# Sett opp tittel
ax.set_title('Overnattinger i Norge, per år fra 1986 til 2023', fontsize=16)

# Sett opp grenser på aksene

```

```

ax.set_xlim(0, 37)

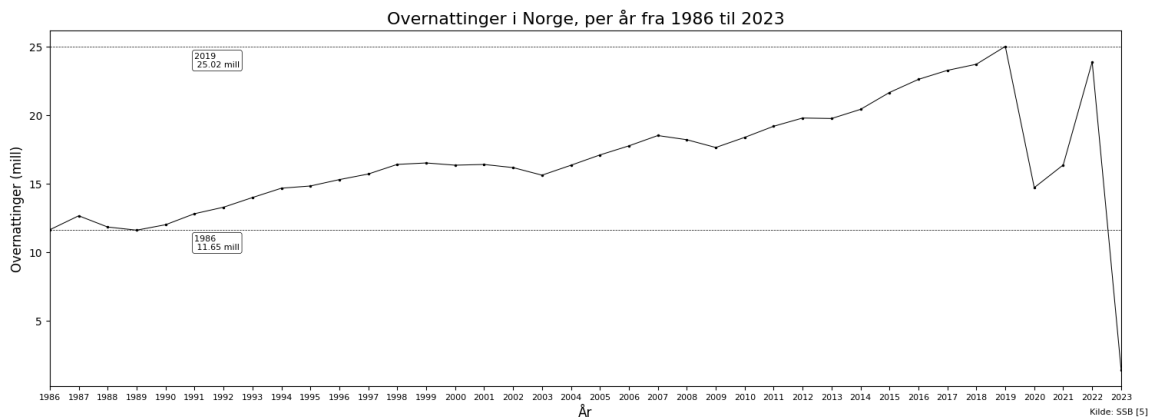
# Vis maksimumsverdi og startverdi på grafen
ax.hlines(df_1['value'].max()/m, 0, 37, color='black', linewidth=0.5, linestyle='--')
ax.text(5, 23.5, '2019 \n 25.02 mill', fontsize=8, bbox=dict(facecolor='white',

ax.hlines(11653357/m, 0, 37, color='black', linewidth=0.5, linestyle='--')
ax.text(5, 10.2, '1986 \n 11.65 mill', fontsize=8, bbox=dict(facecolor='white',

# Kilde
plt.annotate('Kilde: SSB [5]', (0,0), (975,-20), fontsize=8, xycoords='axes frac

plt.show()

```



Vi kan se av de to foregående figurene at hotellovernattinger i Norge har økt jevnt siden 1986, sett at man ser litt bort fra tallene under COVID-19.

Fra 1986 til 2019 ser vi mer enn en dobling i antall overnattinger, samt en dobling i de som kommer i måneden med høyest overnattingstall.

Legg merke til at figurene ikke starter på null.

```

In [ ]: # Lag ny dataframe med gjennomsnittet av overnattinger per måned
df_2 = df[['Måned', 'value']]
df_2 = df_2.groupby('Måned').mean()
df_2 = df_2.reset_index()

```

```

In [ ]: # Lag figuren
fig3, ax = plt.subplots(figsize=(18, 6))

# Plot data
ax.plot(df_2['Måned'], df_2['value']/m, color='black', marker='o', linewidth=0.7

# Sett opp måneder på x-aksen
tick_val = np.arange(0, 12, 1)
tick_lab = ['Januar', 'Februar', 'Mars', 'April', 'Mai', 'Juni', 'Juli', 'August

plt.xticks(tick_val, tick_lab, fontsize=8)

# Gi navn til aksene
ax.set_xlabel('Måned', fontsize=12)
ax.set_ylabel('Overnattinger (mill)', fontsize=12)

# Sett opp tittel
ax.set_title('Overnattinger i Norge, per måned gjennomsnitt (1986 - 2023)', font

```

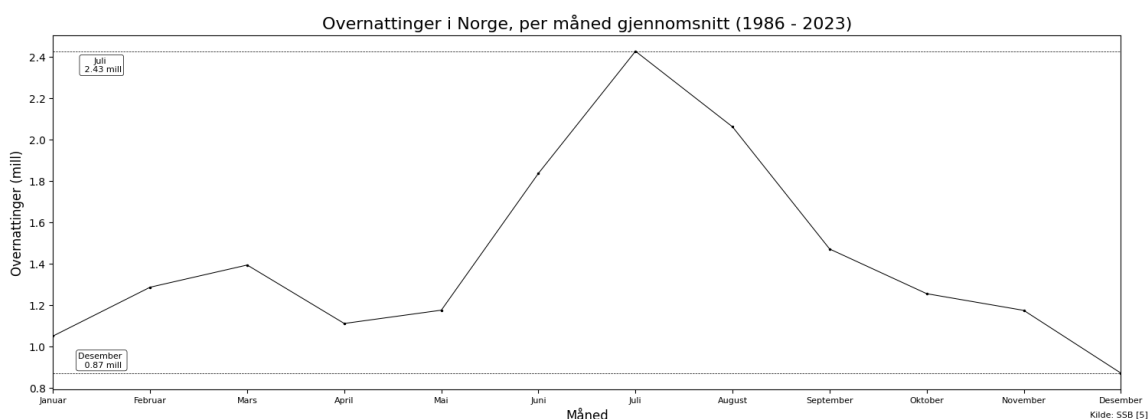
```
# Sett opp grenser på aksene
ax.set_xlim(0, 11)

# Vis maksimumsverdi og minimumsverdi på grafen
ax.hlines(df_2['value'].max()/m, 0, 11, color='black', linewidth=0.5, linestyle='solid')
ax.text(0.5, 2.325, 'Juli \n 2.43 mill', fontsize=8, bbox=dict(facecolor='white', edgecolor='black'))

ax.hlines(df_2['value'].min()/m, 0, 11, color='black', linewidth=0.5, linestyle='solid')
ax.text(0.5, 0.9, 'Desember \n 0.87 mill', fontsize=8, bbox=dict(facecolor='white', edgecolor='black'))

# Kilde
plt.annotate('Kilde: SSB [5]', (0,0), (975,-20), fontsize=8, xycoords='axes fraction')

plt.show()
```



Gjennomsnittlige overnattinger per måned, fra 1986 til 2023, viser tydelig at det er sommermånedene og spesielt Juli som ser mest trafikk.

Juli som den måneden med mest overnattinger ser 2.8 ganger så mange overnattinger som Desember med færrest overnattinger.

Dette tyder på at turismen hovedsakelig utspiller seg i sommermånedene.

```
In [ ]: # JSON spørring til API
postUrl = "https://data.ssb.no/api/v0/no/table/06921/"

apiQuery = {
  "query": [
    {
      "code": "TransportA",
      "selection": {
        "filter": "item",
        "values": [
          "1",
          "2",
          "3",
          "4",
          "5"
        ]
      }
    },
    {
      "code": "Reisetype",
      "selection": {
```

```

        "filter": "item",
        "values": [
            "00"
        ]
    }
}
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_3 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_3, df_id, ds
df_3, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframe
df_3[['År', 'Kvartal']] = df_3.kvartal.str.split("K", expand=True)

df_3.rename(columns = {'kvartal': 'år_kvartal'}, inplace = True)

# Omgjør dataframe med transportmåte og antall overnattinger
df_3 = df_3[['transportmåte', 'År', 'value']]
df_3 = df_3.groupby(['transportmåte', 'År']).sum()
df_3 = df_3.reset_index()

```

```

In [ ]: fig4, ax = plt.subplots(figsize=(18, 6), sharey=True)

ax.plot(df_3[df_3['transportmåte'] == 'Fly']['År'], df_3[df_3['transportmåte'] == 'Fly']['value'])
ax.plot(df_3[df_3['transportmåte'] == 'Båt, ferge, cruiseskip og fritidsbåt']['År'], df_3[df_3['transportmåte'] == 'Båt, ferge, cruiseskip og fritidsbåt']['value'])
ax.plot(df_3[df_3['transportmåte'] == 'Jernbane']['År'], df_3[df_3['transportmåte'] == 'Jernbane']['value'])
ax.plot(df_3[df_3['transportmåte'] == 'Buss (turbuss og rutebuss)']['År'], df_3[df_3['transportmåte'] == 'Buss (turbuss og rutebuss)']['value'])
ax.plot(df_3[df_3['transportmåte'] == 'Personbil, motorsykkel, campingbil osv.']['År'], df_3[df_3['transportmåte'] == 'Personbil, motorsykkel, campingbil osv.']['value'])

ax.set_xlabel('År', fontsize=12)
ax.set_ylabel('Antall reiser (mill)', fontsize=12)

ax.set_title('Antall reiser i Norge, per transportmåte (1986 - 2023)', fontsize=12)

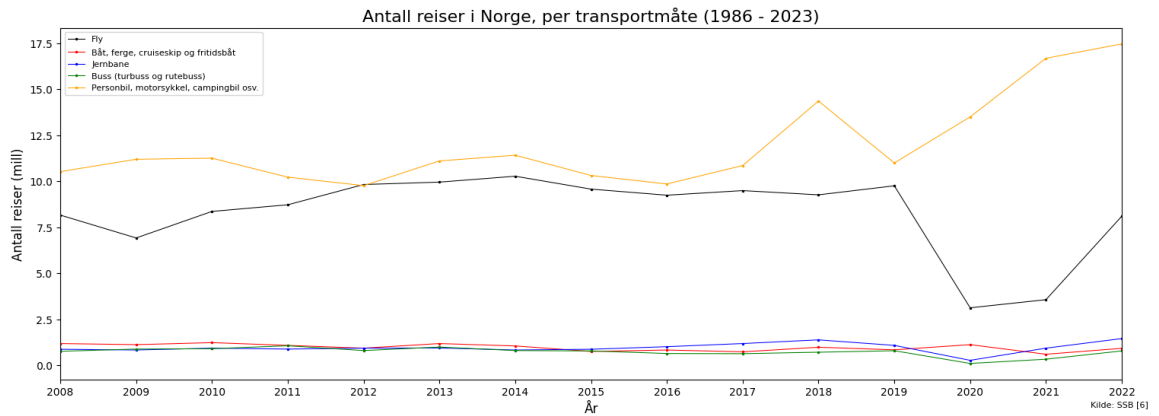
ax.set_xlim(0, 14)

ax.legend(loc='upper left', fontsize=8)

plt.annotate('Kilde: SSB [6]', (0,0), (975,-20), fontsize=8, xycoords='axes fraction')

plt.show()

```



I figuren over ser vi reiser i Norge, etter transportmetode. Denne tyder på at majoriteten av reisene fra turister hovedsaklig vil utløse overnattingsbehov ved norske fasiliteter.

Unntaket vil være cruiseship, men her kan beskatning pålegges av havner der skipene legger til kai.

Merk at i denne figuren er cruiseship inlemmet med båt, ferge og fritidsbåt, og vil med dette være lavere enn grafen tilsier.

```
In [ ]: # JSON spørring til API
        postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

        apiQuery = {
            "query": [
                {
                    "code": "Region",
                    "selection": {
                        "filter": "vs:LandetNyn",
                        "values": [
                            "ØN"
                        ]
                    }
                },
                {
                    "code": "ContentsCode",
                    "selection": {
                        "filter": "item",
                        "values": [
                            "KapasitetRom"
                        ]
                    }
                }
            ],
            {
                "code": "Tid",
                "selection": {
                    "filter": "item",
                    "values": [
                        "2016M01",
                        "2016M02",
                        "2016M03",
                        "2016M04",
                        "2016M05",
                        "2016M06",
                        "2016M07",
                        "2016M08",
```



"2016M09",  
"2016M10",  
"2016M11",  
"2016M12",  
"2017M01",  
"2017M02",  
"2017M03",  
"2017M04",  
"2017M05",  
"2017M06",  
"2017M07",  
"2017M08",  
"2017M09",  
"2017M10",  
"2017M11",  
"2017M12",  
"2018M01",  
"2018M02",  
"2018M03",  
"2018M04",  
"2018M05",  
"2018M06",  
"2018M07",  
"2018M08",  
"2018M09",  
"2018M10",  
"2018M11",  
"2018M12",  
"2019M01",  
"2019M02",  
"2019M03",  
"2019M04",  
"2019M05",  
"2019M06",  
"2019M07",  
"2019M08",  
"2019M09",  
"2019M10",  
"2019M11",  
"2019M12",  
"2020M01",  
"2020M02",  
"2020M03",  
"2020M04",  
"2020M05",  
"2020M06",  
"2020M07",  
"2020M08",  
"2020M09",  
"2020M10",  
"2020M11",  
"2020M12",  
"2021M01",  
"2021M02",  
"2021M03",  
"2021M04",  
"2021M05",  
"2021M06",  
"2021M07",  
"2021M08",

```

        "2021M09",
        "2021M10",
        "2021M11",
        "2021M12",
        "2022M01",
        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
}
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_4 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_4, df_id, ds
df_4, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_4[['År', 'Måned']] = df_4.måned.str.split("M", expand=True)

df_4.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

# JSON spørring til API
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
    "query": [
        {
            "code": "Region",
            "selection": {
                "filter": "vs:LandetNyn",
                "values": [
                    "0N"
                ]
            }
        }
    ],
    {

```

```
"code": "ContentsCode",
"selection": {
  "filter": "item",
  "values": [
    "Losji0ms"
  ]
},
{
  "code": "Tid",
  "selection": {
    "filter": "item",
    "values": [
      "2016M01",
      "2016M02",
      "2016M03",
      "2016M04",
      "2016M05",
      "2016M06",
      "2016M07",
      "2016M08",
      "2016M09",
      "2016M10",
      "2016M11",
      "2016M12",
      "2017M01",
      "2017M02",
      "2017M03",
      "2017M04",
      "2017M05",
      "2017M06",
      "2017M07",
      "2017M08",
      "2017M09",
      "2017M10",
      "2017M11",
      "2017M12",
      "2018M01",
      "2018M02",
      "2018M03",
      "2018M04",
      "2018M05",
      "2018M06",
      "2018M07",
      "2018M08",
      "2018M09",
      "2018M10",
      "2018M11",
      "2018M12",
      "2019M01",
      "2019M02",
      "2019M03",
      "2019M04",
      "2019M05",
      "2019M06",
      "2019M07",
      "2019M08",
      "2019M09",
      "2019M10",
      "2019M11",
```

```

        "2019M12",
        "2020M01",
        "2020M02",
        "2020M03",
        "2020M04",
        "2020M05",
        "2020M06",
        "2020M07",
        "2020M08",
        "2020M09",
        "2020M10",
        "2020M11",
        "2020M12",
        "2021M01",
        "2021M02",
        "2021M03",
        "2021M04",
        "2021M05",
        "2021M06",
        "2021M07",
        "2021M08",
        "2021M09",
        "2021M10",
        "2021M11",
        "2021M12",
        "2022M01",
        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
}],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_5 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_5, df_id, ds
df_5, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen

```

```
df_5[['År', 'Måned']] = df_5.måned.str.split("M", expand=True)

df_5.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

# Lag en ny dataframe med gjennomsnittet av alle månedene i årene
df_41 = df_4[['variabel', 'År', 'value']]
df_41 = df_41.groupby(['variabel', 'År']).mean()

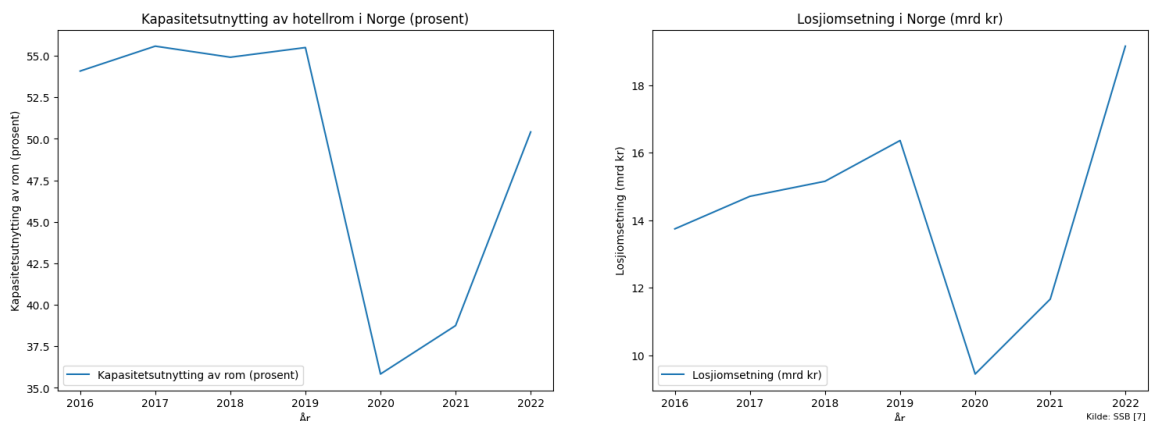
# Lag en ny dataframe med gjennomsnittet av alle månedene i årene
df_51 = df_5[['variabel', 'År', 'value']]
df_51 = df_51.groupby(['variabel', 'År']).sum()/m
```

```
In [ ]: # Lag figuren med to axer
fig5, (ax1, ax2) = plt.subplots(1,2 , figsize=(18,6), sharex=True)

# Plot dataene, Legg til labels og tittel, og legg til Legend
ax1.plot(df_41.loc['Kapasitetsutnyttning av rom (prosent)', 'value'], label='Kapasitetsutnyttning av rom (prosent)')
ax2.plot(df_51.loc['Losjtomsetning (1 000 kr)', 'value'], label='Losjtomsetning (1 000 kr)')
ax1.set_title('Kapasitetsutnyttning av hotellrom i Norge (prosent)')
ax2.set_title('Losjtomsetning i Norge (mrd kr)')
ax1.set_ylabel('Kapasitetsutnyttning av rom (prosent)')
ax2.set_ylabel('Losjtomsetning (mrd kr)')
ax1.set_xlabel('År')
ax2.set_xlabel('År')
ax1.legend(loc='lower left')
ax2.legend(loc='lower left')

# Kilde
plt.annotate('Kilde: SSB [7]', (0,0), (400,-20), fontsize=8, xycoords='axes fraction')

plt.show()
```



Kapasitetsutnyttningen og losjtomsetningen ser ut til å ha kommet seg tilbake etter COVID-19.

Utnyttelse og omsetning vil variere mer basert på region, da forskjellige regioner har forskjellig popularitet.

Det er jo da også de mer populære destinasjonene som vil kunne tjene mest på en turistskatt, samt muligens ha mest behov for det.

```
In [ ]: # JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"
```

```
apiQuery = {
  "query": [
    {
      "code": "Region",
      "selection": {
        "filter": "vs:ReiselivsRegioner",
        "values": [
          "54101"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "KapasitetRom"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
          "2016M01",
          "2016M02",
          "2016M03",
          "2016M04",
          "2016M05",
          "2016M06",
          "2016M07",
          "2016M08",
          "2016M09",
          "2016M10",
          "2016M11",
          "2016M12",
          "2017M01",
          "2017M02",
          "2017M03",
          "2017M04",
          "2017M05",
          "2017M06",
          "2017M07",
          "2017M08",
          "2017M09",
          "2017M10",
          "2017M11",
          "2017M12",
          "2018M01",
          "2018M02",
          "2018M03",
          "2018M04",
          "2018M05",
          "2018M06",
          "2018M07",
          "2018M08",
          "2018M09",
          "2018M10",

```

```
        "2018M11",
        "2018M12",
        "2019M01",
        "2019M02",
        "2019M03",
        "2019M04",
        "2019M05",
        "2019M06",
        "2019M07",
        "2019M08",
        "2019M09",
        "2019M10",
        "2019M11",
        "2019M12",
        "2020M01",
        "2020M02",
        "2020M03",
        "2020M04",
        "2020M05",
        "2020M06",
        "2020M07",
        "2020M08",
        "2020M09",
        "2020M10",
        "2020M11",
        "2020M12",
        "2021M01",
        "2021M02",
        "2021M03",
        "2021M04",
        "2021M05",
        "2021M06",
        "2021M07",
        "2021M08",
        "2021M09",
        "2021M10",
        "2021M11",
        "2021M12",
        "2022M01",
        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
},
],
"response": {
    "format": "json-stat2"
}
}
```

```
def apiToDataframe(postUrl, query):
```

```

res = requests.post(postUrl, json=query)

ds = pyjstat.Dataset.read(res.text)

df_42 = ds.write('dataframe')

df_id = ds.write('dataframe', naming='id')

return df_42, df_id, ds
df_42, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_42[['År', 'Måned']] = df_42.måned.str.split("M", expand=True)

df_42.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, in

#-----

# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
  "query": [
    {
      "code": "Region",
      "selection": {
        "filter": "vs:ReiselivsRegioner",
        "values": [
          "54101"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "LosjiOms"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
          "2016M01",
          "2016M02",
          "2016M03",
          "2016M04",
          "2016M05",
          "2016M06",
          "2016M07",
          "2016M08",
          "2016M09",
          "2016M10",
          "2016M11",
          "2016M12",
          "2017M01",

```



"2017M02",  
"2017M03",  
"2017M04",  
"2017M05",  
"2017M06",  
"2017M07",  
"2017M08",  
"2017M09",  
"2017M10",  
"2017M11",  
"2017M12",  
"2018M01",  
"2018M02",  
"2018M03",  
"2018M04",  
"2018M05",  
"2018M06",  
"2018M07",  
"2018M08",  
"2018M09",  
"2018M10",  
"2018M11",  
"2018M12",  
"2019M01",  
"2019M02",  
"2019M03",  
"2019M04",  
"2019M05",  
"2019M06",  
"2019M07",  
"2019M08",  
"2019M09",  
"2019M10",  
"2019M11",  
"2019M12",  
"2020M01",  
"2020M02",  
"2020M03",  
"2020M04",  
"2020M05",  
"2020M06",  
"2020M07",  
"2020M08",  
"2020M09",  
"2020M10",  
"2020M11",  
"2020M12",  
"2021M01",  
"2021M02",  
"2021M03",  
"2021M04",  
"2021M05",  
"2021M06",  
"2021M07",  
"2021M08",  
"2021M09",  
"2021M10",  
"2021M11",  
"2021M12",  
"2022M01",

```

        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
},
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_52 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_52, df_id, ds
df_52, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_52[['År', 'Måned']] = df_52.måned.str.split("M", expand=True)

df_52.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
    "query": [
        {
            "code": "Region",
            "selection": {
                "filter": "vs:ReiselivsRegioner",
                "values": [
                    "42105"
                ]
            }
        },
        {
            "code": "ContentsCode",
            "selection": {
                "filter": "item",
                "values": [
                    "KapasitetRom"
                ]
            }
        }
    ]
}

```

```
    ]
  }
},
{
  "code": "Tid",
  "selection": {
    "filter": "item",
    "values": [
      "2016M01",
      "2016M02",
      "2016M03",
      "2016M04",
      "2016M05",
      "2016M06",
      "2016M07",
      "2016M08",
      "2016M09",
      "2016M10",
      "2016M11",
      "2016M12",
      "2017M01",
      "2017M02",
      "2017M03",
      "2017M04",
      "2017M05",
      "2017M06",
      "2017M07",
      "2017M08",
      "2017M09",
      "2017M10",
      "2017M11",
      "2017M12",
      "2018M01",
      "2018M02",
      "2018M03",
      "2018M04",
      "2018M05",
      "2018M06",
      "2018M07",
      "2018M08",
      "2018M09",
      "2018M10",
      "2018M11",
      "2018M12",
      "2019M01",
      "2019M02",
      "2019M03",
      "2019M04",
      "2019M05",
      "2019M06",
      "2019M07",
      "2019M08",
      "2019M09",
      "2019M10",
      "2019M11",
      "2019M12",
      "2020M01",
      "2020M02",
      "2020M03",
      "2020M04",
```

```

        "2020M05",
        "2020M06",
        "2020M07",
        "2020M08",
        "2020M09",
        "2020M10",
        "2020M11",
        "2020M12",
        "2021M01",
        "2021M02",
        "2021M03",
        "2021M04",
        "2021M05",
        "2021M06",
        "2021M07",
        "2021M08",
        "2021M09",
        "2021M10",
        "2021M11",
        "2021M12",
        "2022M01",
        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
},
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_43 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_43, df_id, ds
df_43, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_43[['År', 'Måned']] = df_43.måned.str.split("M", expand=True)

df_43.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

```

```
# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
  "query": [
    {
      "code": "Region",
      "selection": {
        "filter": "vs:ReiselivsRegioner",
        "values": [
          "42105"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "Losji0ms"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
          "2016M01",
          "2016M02",
          "2016M03",
          "2016M04",
          "2016M05",
          "2016M06",
          "2016M07",
          "2016M08",
          "2016M09",
          "2016M10",
          "2016M11",
          "2016M12",
          "2017M01",
          "2017M02",
          "2017M03",
          "2017M04",
          "2017M05",
          "2017M06",
          "2017M07",
          "2017M08",
          "2017M09",
          "2017M10",
          "2017M11",
          "2017M12",
          "2018M01",
          "2018M02",
          "2018M03",
          "2018M04",
          "2018M05",
          "2018M06",
          "2018M07",

```

```
"2018M08",
"2018M09",
"2018M10",
"2018M11",
"2018M12",
"2019M01",
"2019M02",
"2019M03",
"2019M04",
"2019M05",
"2019M06",
"2019M07",
"2019M08",
"2019M09",
"2019M10",
"2019M11",
"2019M12",
"2020M01",
"2020M02",
"2020M03",
"2020M04",
"2020M05",
"2020M06",
"2020M07",
"2020M08",
"2020M09",
"2020M10",
"2020M11",
"2020M12",
"2021M01",
"2021M02",
"2021M03",
"2021M04",
"2021M05",
"2021M06",
"2021M07",
"2021M08",
"2021M09",
"2021M10",
"2021M11",
"2021M12",
"2022M01",
"2022M02",
"2022M03",
"2022M04",
"2022M05",
"2022M06",
"2022M07",
"2022M08",
"2022M09",
"2022M10",
"2022M11",
"2022M12"
]
}
},
],
"response": {
  "format": "json-stat2"
}
```

```

}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_53 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_53, df_id, ds
df_53, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_53[['År', 'Måned']] = df_53.måned.str.split("M", expand=True)

df_53.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

# Lag nye dataframer

df_42 = df_42[['variabel', 'År', 'value']]
df_42 = df_42.groupby(['variabel', 'År']).mean()

df_52 = df_52[['variabel', 'År', 'value']]
df_52 = df_52.groupby(['variabel', 'År']).sum()/1000

df_43 = df_43[['variabel', 'År', 'value']]
df_43 = df_43.groupby(['variabel', 'År']).mean()

df_53 = df_53[['variabel', 'År', 'value']]
df_53 = df_53.groupby(['variabel', 'År']).sum()/1000

```

```

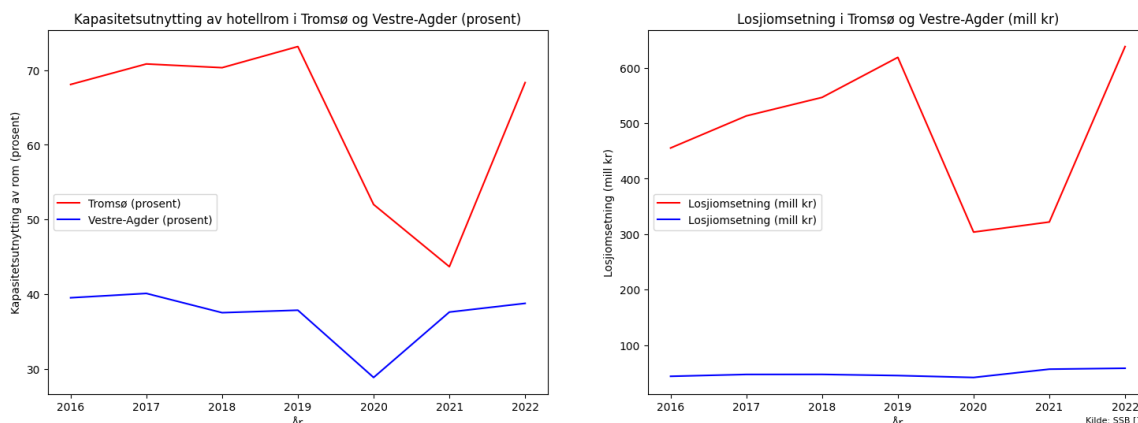
In [ ]: # Lag figurer
fig6, (ax1, ax2) = plt.subplots(1,2 , figsize=(18,6), sharex=True)

# Plot figurer, Legg til labels og tittel, og endre på x-aksen
ax1.plot(df_42.loc['Kapasitetsutnyttning av rom (prosent)', 'value'], label='Tromsø')
ax1.plot(df_43.loc['Kapasitetsutnyttning av rom (prosent)', 'value'], label='Vest-Agder')
ax2.plot(df_52.loc['Losjiosetning (1 000 kr)', 'value'], label='Losjiosetning i Tromsø')
ax2.plot(df_53.loc['Losjiosetning (1 000 kr)', 'value'], label='Losjiosetning i Vest-Agder')
ax1.set_title('Kapasitetsutnyttning av hotellrom i Tromsø og Vestre-Agder (prosent)')
ax2.set_title('Losjiosetning i Tromsø og Vestre-Agder (mill kr)')
ax1.set_ylabel('Kapasitetsutnyttning av rom (prosent)')
ax2.set_ylabel('Losjiosetning (mill kr)')
ax1.set_xlabel('År')
ax2.set_xlabel('År')
ax1.legend(loc='center left')
ax2.legend(loc='center left')

# Legg til kilde
plt.annotate('Kilde: SSB [7]', (0,0), (400,-20), fontsize=8, xycoords='axes fraction')

plt.show()

```



I figuren over ser vi en god fremstilling av forskjellene mellom reisedestinasjoner i Norge.

Tromsø har både langt høyere kapasitetsutnyttelse, samt langt høyere losjionsetning enn Vestre-Agder.

Foregående data vil være viktig for denne analysen, da de illustrerer volatiliteten i norsk reiseliv og forskjellene på regioner i Norge, som kan brukes til å evaluere både hvilke regioner som kan dra nytte av en turistskatt, samt hvilke tidsrom.

### | 3. Samfunnsøkonomiske effekter av en turistskatt |

Hensikten med en turistskatt er å få turister til å være med på å finansiere fellesgodene som turister benytter seg av i Norge, samt verne om naturen slik at denne ikke blir ødelagt for profit. Den vil også ha den effekten å kunne dempe etterspørselen etter turisme til Norge, som igjen vil føre til mindre slitasje og skade.

```
In [ ]: # JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
  "query": [
    {
      "code": "Region",
      "selection": {
        "filter": "vs:LandetNyn",
        "values": [
          "ØN"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "PrisRom"
        ]
      }
    }
  ],
}
```



```
{
  "code": "Tid",
  "selection": {
    "filter": "item",
    "values": [
      "2016M01",
      "2016M02",
      "2016M03",
      "2016M04",
      "2016M05",
      "2016M06",
      "2016M07",
      "2016M08",
      "2016M09",
      "2016M10",
      "2016M11",
      "2016M12",
      "2017M01",
      "2017M02",
      "2017M03",
      "2017M04",
      "2017M05",
      "2017M06",
      "2017M07",
      "2017M08",
      "2017M09",
      "2017M10",
      "2017M11",
      "2017M12",
      "2018M01",
      "2018M02",
      "2018M03",
      "2018M04",
      "2018M05",
      "2018M06",
      "2018M07",
      "2018M08",
      "2018M09",
      "2018M10",
      "2018M11",
      "2018M12",
      "2019M01",
      "2019M02",
      "2019M03",
      "2019M04",
      "2019M05",
      "2019M06",
      "2019M07",
      "2019M08",
      "2019M09",
      "2019M10",
      "2019M11",
      "2019M12",
      "2020M01",
      "2020M02",
      "2020M03",
      "2020M04",
      "2020M05",
      "2020M06",
      "2020M07",
```

```

        "2020M08",
        "2020M09",
        "2020M10",
        "2020M11",
        "2020M12",
        "2021M01",
        "2021M02",
        "2021M03",
        "2021M04",
        "2021M05",
        "2021M06",
        "2021M07",
        "2021M08",
        "2021M09",
        "2021M10",
        "2021M11",
        "2021M12",
        "2022M01",
        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_44 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_44, df_id, ds
df_44, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_44[['År', 'Måned']] = df_44.måned.str.split("M", expand=True)

df_44.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

```

```
apiQuery = {
  "query": [
    {
      "code": "Region",
      "selection": {
        "filter": "vs:LandetNyn",
        "values": [
          "ØN"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "LosjiOmsRom"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
          "2016M01",
          "2016M02",
          "2016M03",
          "2016M04",
          "2016M05",
          "2016M06",
          "2016M07",
          "2016M08",
          "2016M09",
          "2016M10",
          "2016M11",
          "2016M12",
          "2017M01",
          "2017M02",
          "2017M03",
          "2017M04",
          "2017M05",
          "2017M06",
          "2017M07",
          "2017M08",
          "2017M09",
          "2017M10",
          "2017M11",
          "2017M12",
          "2018M01",
          "2018M02",
          "2018M03",
          "2018M04",
          "2018M05",
          "2018M06",
          "2018M07",
          "2018M08",
          "2018M09",
          "2018M10",

```

```
    "2018M11",
    "2018M12",
    "2019M01",
    "2019M02",
    "2019M03",
    "2019M04",
    "2019M05",
    "2019M06",
    "2019M07",
    "2019M08",
    "2019M09",
    "2019M10",
    "2019M11",
    "2019M12",
    "2020M01",
    "2020M02",
    "2020M03",
    "2020M04",
    "2020M05",
    "2020M06",
    "2020M07",
    "2020M08",
    "2020M09",
    "2020M10",
    "2020M11",
    "2020M12",
    "2021M01",
    "2021M02",
    "2021M03",
    "2021M04",
    "2021M05",
    "2021M06",
    "2021M07",
    "2021M08",
    "2021M09",
    "2021M10",
    "2021M11",
    "2021M12",
    "2022M01",
    "2022M02",
    "2022M03",
    "2022M04",
    "2022M05",
    "2022M06",
    "2022M07",
    "2022M08",
    "2022M09",
    "2022M10",
    "2022M11",
    "2022M12"
  ]
}
},
],
"response": {
  "format": "json-stat2"
}
}
```

```
def apiToDataframe(postUrl, query):
```

```

res = requests.post(postUrl, json=query)

ds = pyjstat.Dataset.read(res.text)

df_54 = ds.write('dataframe')

df_id = ds.write('dataframe', naming='id')

return df_54, df_id, ds
df_54, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_54[['År', 'Måned']] = df_54.måned.str.split("M", expand=True)

df_54.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, in

#-----

# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
  "query": [
    {
      "code": "Region",
      "selection": {
        "filter": "vs:ReiselivsRegioner",
        "values": [
          "54101"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "PrisRom"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
          "2016M01",
          "2016M02",
          "2016M03",
          "2016M04",
          "2016M05",
          "2016M06",
          "2016M07",
          "2016M08",
          "2016M09",
          "2016M10",
          "2016M11",
          "2016M12",
          "2017M01",

```

"2017M02",  
"2017M03",  
"2017M04",  
"2017M05",  
"2017M06",  
"2017M07",  
"2017M08",  
"2017M09",  
"2017M10",  
"2017M11",  
"2017M12",  
"2018M01",  
"2018M02",  
"2018M03",  
"2018M04",  
"2018M05",  
"2018M06",  
"2018M07",  
"2018M08",  
"2018M09",  
"2018M10",  
"2018M11",  
"2018M12",  
"2019M01",  
"2019M02",  
"2019M03",  
"2019M04",  
"2019M05",  
"2019M06",  
"2019M07",  
"2019M08",  
"2019M09",  
"2019M10",  
"2019M11",  
"2019M12",  
"2020M01",  
"2020M02",  
"2020M03",  
"2020M04",  
"2020M05",  
"2020M06",  
"2020M07",  
"2020M08",  
"2020M09",  
"2020M10",  
"2020M11",  
"2020M12",  
"2021M01",  
"2021M02",  
"2021M03",  
"2021M04",  
"2021M05",  
"2021M06",  
"2021M07",  
"2021M08",  
"2021M09",  
"2021M10",  
"2021M11",  
"2021M12",  
"2022M01",

```

        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
},
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_45 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_45, df_id, ds
df_45, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_45[['År', 'Måned']] = df_45.måned.str.split("M", expand=True)

df_45.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
    "query": [
        {
            "code": "Region",
            "selection": {
                "filter": "vs:ReiselivsRegioner",
                "values": [
                    "54101"
                ]
            }
        },
        {
            "code": "ContentsCode",
            "selection": {
                "filter": "item",
                "values": [
                    "LosjiOmsRom"
                ]
            }
        }
    ]
}

```

```
]
}
},
{
  "code": "Tid",
  "selection": {
    "filter": "item",
    "values": [
      "2016M01",
      "2016M02",
      "2016M03",
      "2016M04",
      "2016M05",
      "2016M06",
      "2016M07",
      "2016M08",
      "2016M09",
      "2016M10",
      "2016M11",
      "2016M12",
      "2017M01",
      "2017M02",
      "2017M03",
      "2017M04",
      "2017M05",
      "2017M06",
      "2017M07",
      "2017M08",
      "2017M09",
      "2017M10",
      "2017M11",
      "2017M12",
      "2018M01",
      "2018M02",
      "2018M03",
      "2018M04",
      "2018M05",
      "2018M06",
      "2018M07",
      "2018M08",
      "2018M09",
      "2018M10",
      "2018M11",
      "2018M12",
      "2019M01",
      "2019M02",
      "2019M03",
      "2019M04",
      "2019M05",
      "2019M06",
      "2019M07",
      "2019M08",
      "2019M09",
      "2019M10",
      "2019M11",
      "2019M12",
      "2020M01",
      "2020M02",
      "2020M03",
      "2020M04",
```



```

        "2020M05",
        "2020M06",
        "2020M07",
        "2020M08",
        "2020M09",
        "2020M10",
        "2020M11",
        "2020M12",
        "2021M01",
        "2021M02",
        "2021M03",
        "2021M04",
        "2021M05",
        "2021M06",
        "2021M07",
        "2021M08",
        "2021M09",
        "2021M10",
        "2021M11",
        "2021M12",
        "2022M01",
        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
},
],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_55 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_55, df_id, ds
df_55, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_55[['År', 'Måned']] = df_55.måned.str.split("M", expand=True)

df_55.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

```

```
# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
  "query": [
    {
      "code": "Region",
      "selection": {
        "filter": "vs:ReiselivsRegioner",
        "values": [
          "42105"
        ]
      }
    },
    {
      "code": "ContentsCode",
      "selection": {
        "filter": "item",
        "values": [
          "PrisRom"
        ]
      }
    },
    {
      "code": "Tid",
      "selection": {
        "filter": "item",
        "values": [
          "2016M01",
          "2016M02",
          "2016M03",
          "2016M04",
          "2016M05",
          "2016M06",
          "2016M07",
          "2016M08",
          "2016M09",
          "2016M10",
          "2016M11",
          "2016M12",
          "2017M01",
          "2017M02",
          "2017M03",
          "2017M04",
          "2017M05",
          "2017M06",
          "2017M07",
          "2017M08",
          "2017M09",
          "2017M10",
          "2017M11",
          "2017M12",
          "2018M01",
          "2018M02",
          "2018M03",
          "2018M04",
          "2018M05",
          "2018M06",
          "2018M07",

```

```
"2018M08",
"2018M09",
"2018M10",
"2018M11",
"2018M12",
"2019M01",
"2019M02",
"2019M03",
"2019M04",
"2019M05",
"2019M06",
"2019M07",
"2019M08",
"2019M09",
"2019M10",
"2019M11",
"2019M12",
"2020M01",
"2020M02",
"2020M03",
"2020M04",
"2020M05",
"2020M06",
"2020M07",
"2020M08",
"2020M09",
"2020M10",
"2020M11",
"2020M12",
"2021M01",
"2021M02",
"2021M03",
"2021M04",
"2021M05",
"2021M06",
"2021M07",
"2021M08",
"2021M09",
"2021M10",
"2021M11",
"2021M12",
"2022M01",
"2022M02",
"2022M03",
"2022M04",
"2022M05",
"2022M06",
"2022M07",
"2022M08",
"2022M09",
"2022M10",
"2022M11",
"2022M12"
]
}
},
"response": {
  "format": "json-stat2"
}
```

```

}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_46 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_46, df_id, ds
df_46, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_46[['År', 'Måned']] = df_46.måned.str.split("M", expand=True)

df_46.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, in

#-----

# JSON spørring
postUrl = "https://data.ssb.no/api/v0/no/table/13156/"

apiQuery = {
    "query": [
        {
            "code": "Region",
            "selection": {
                "filter": "vs:ReiselivsRegioner",
                "values": [
                    "42105"
                ]
            }
        },
        {
            "code": "ContentsCode",
            "selection": {
                "filter": "item",
                "values": [
                    "LosjiOmsRom"
                ]
            }
        },
        {
            "code": "Tid",
            "selection": {
                "filter": "item",
                "values": [
                    "2016M01",
                    "2016M02",
                    "2016M03",
                    "2016M04",
                    "2016M05",
                    "2016M06",
                    "2016M07",
                    "2016M08",
                    "2016M09",
                    "2016M10",

```

"2016M11",  
"2016M12",  
"2017M01",  
"2017M02",  
"2017M03",  
"2017M04",  
"2017M05",  
"2017M06",  
"2017M07",  
"2017M08",  
"2017M09",  
"2017M10",  
"2017M11",  
"2017M12",  
"2018M01",  
"2018M02",  
"2018M03",  
"2018M04",  
"2018M05",  
"2018M06",  
"2018M07",  
"2018M08",  
"2018M09",  
"2018M10",  
"2018M11",  
"2018M12",  
"2019M01",  
"2019M02",  
"2019M03",  
"2019M04",  
"2019M05",  
"2019M06",  
"2019M07",  
"2019M08",  
"2019M09",  
"2019M10",  
"2019M11",  
"2019M12",  
"2020M01",  
"2020M02",  
"2020M03",  
"2020M04",  
"2020M05",  
"2020M06",  
"2020M07",  
"2020M08",  
"2020M09",  
"2020M10",  
"2020M11",  
"2020M12",  
"2021M01",  
"2021M02",  
"2021M03",  
"2021M04",  
"2021M05",  
"2021M06",  
"2021M07",  
"2021M08",  
"2021M09",  
"2021M10",

```

        "2021M11",
        "2021M12",
        "2022M01",
        "2022M02",
        "2022M03",
        "2022M04",
        "2022M05",
        "2022M06",
        "2022M07",
        "2022M08",
        "2022M09",
        "2022M10",
        "2022M11",
        "2022M12"
    ]
}
}],
"response": {
    "format": "json-stat2"
}
}

def apiToDataframe(postUrl, query):

    res = requests.post(postUrl, json=query)

    ds = pyjstat.Dataset.read(res.text)

    df_56 = ds.write('dataframe')

    df_id = ds.write('dataframe', naming='id')

    return df_56, df_id, ds
df_56, df_id, ds = apiToDataframe(postUrl, apiQuery)

# Endre på dataframen
df_56[['År', 'Måned']] = df_56.måned.str.split("M", expand=True)

df_56.rename(columns = {'måned': 'år_måned', 'statistikkvariabel': 'variabel'}, inplace=True)

#-----

# Lag nye dataframer

df_44 = df_56[['variabel', 'år_måned', 'value']]

df_54 = df_56[['variabel', 'år_måned', 'value']]

df_45 = df_56[['variabel', 'år_måned', 'value']]

df_55 = df_56[['variabel', 'år_måned', 'value']]

df_46 = df_56[['variabel', 'år_måned', 'value']]

df_56 = df_56[['variabel', 'år_måned', 'value']]

```

```

In [ ]: # Lag figurer
fig7, ax = plt.subplots(figsize=(18,6))

```

```

# Plot data
ax.plot(df_44['år_måned'], df_44['value'], label='Pris per rom (kr)')
ax.plot(df_54['år_måned'], df_54['value'], label='Losjtomsetning per tilgjengeleg rom (kr)')

# Sett opp x-aksen
tick_val = np.arange(0, 84, 12)
tick_lab = [x for x in range(2016, 2023)]

plt.xticks(tick_val, tick_lab, fontsize=8)

# Sett opp tittel, labels og limits
ax.set_title('Pris per rom (kr) og Losjtomsetning per tilgjengeleg rom (kr) i he
ax.set_xlabel('År', fontsize=14)
ax.set_ylabel('Pris i kr', fontsize=14)
ax.set_xlim(0, 84)

# Kilde
plt.annotate('Kilde: SSB [7]', (0,0), (950,-20), fontsize=8, xycoords='axes frac

ax.legend()

plt.show()

fig8, ax = plt.subplots(figsize=(18,6))

ax.plot(df_45['år_måned'], df_45['value'], label='Pris per rom (kr)')
ax.plot(df_55['år_måned'], df_55['value'], label='Losjtomsetning per tilgjengeleg rom (kr)')

tick_val = np.arange(0, 84, 12)
tick_lab = [x for x in range(2016, 2023)]

plt.xticks(tick_val, tick_lab, fontsize=8)

ax.set_title('Pris per rom (kr) og Losjtomsetning per tilgjengeleg rom (kr) i Tr
ax.set_xlabel('År', fontsize=14)
ax.set_ylabel('Pris i kr', fontsize=14)
ax.set_xlim(0, 84)

plt.annotate('Kilde: SSB [7]', (0,0), (950,-20), fontsize=8, xycoords='axes frac

ax.legend()

plt.show()

fig9, ax = plt.subplots(figsize=(18,6))

ax.plot(df_46['år_måned'], df_46['value'], label='Pris per rom (kr)')
ax.plot(df_56['år_måned'], df_56['value'], label='Losjtomsetning per tilgjengeleg rom (kr)')

tick_val = np.arange(0, 84, 12)
tick_lab = [x for x in range(2016, 2023)]

plt.xticks(tick_val, tick_lab, fontsize=8)

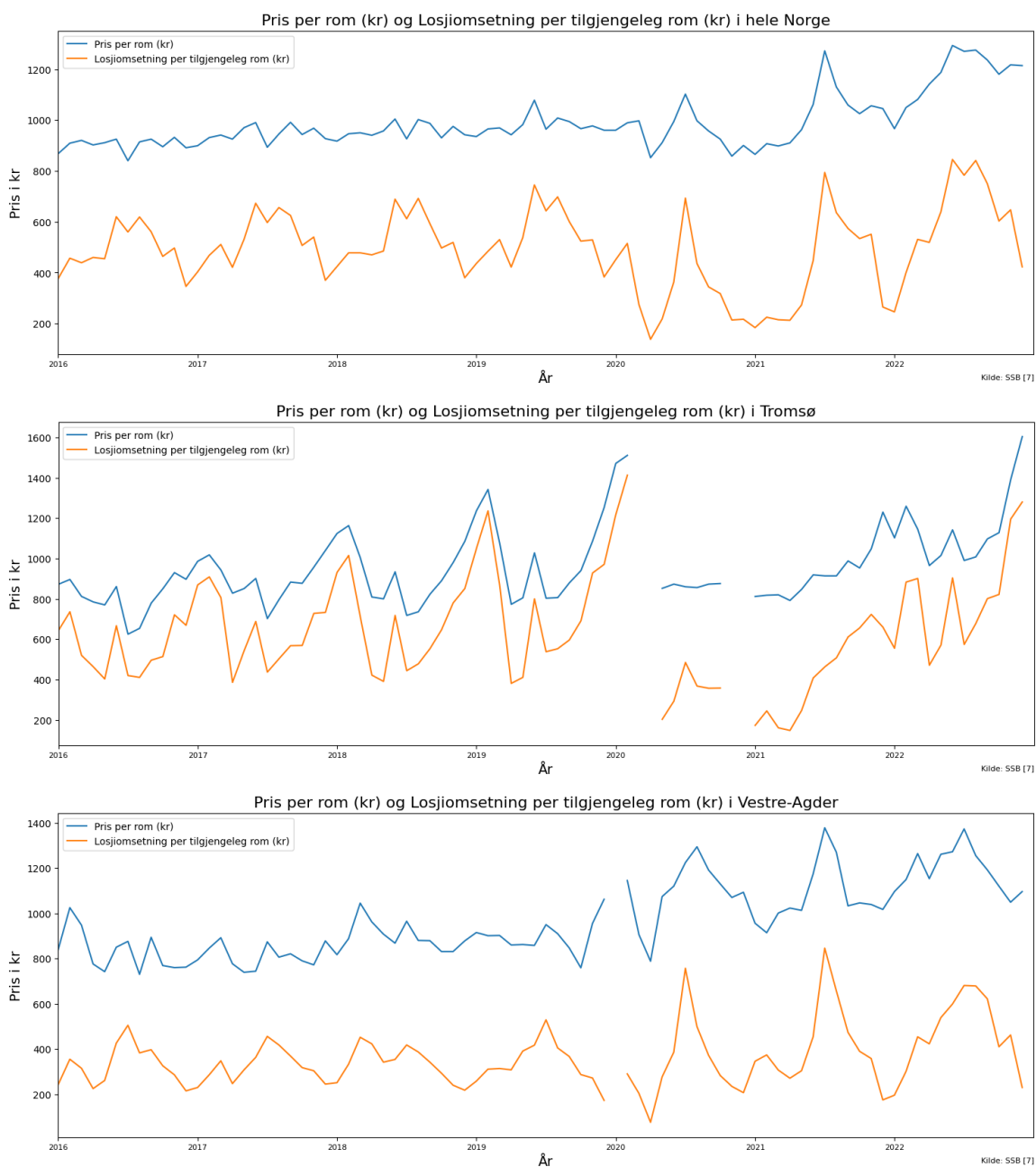
ax.set_title('Pris per rom (kr) og Losjtomsetning per tilgjengeleg rom (kr) i Ve
ax.set_xlabel('År', fontsize=14)
ax.set_ylabel('Pris i kr', fontsize=14)
ax.set_xlim(0, 84)

plt.annotate('Kilde: SSB [7]', (0,0), (950,-20), fontsize=8, xycoords='axes frac

```

```
ax.legend()
```

```
plt.show()
```



I den første av de tre foregående figurene kan vi se tydelig at etterspørselen stiger i sommermånedene når turistene kommer. Dette vitner om at priselastisiteten faller sammenliknet med vintermånedene.

Dette kan føre til en mindre negativ effekt på markedslukevekten i sommermånedene enn i vintermånedene, ved en introduksjon av turistskatt.

Vi ser også i den andre figuren at Tromsø får flere overnattinger i vintermånedene i motsetning til hele landet totalt sett, noe som tyder på lokal kontroll over turistskatt vil være mest hensiktsmessig.

I tredje figur ser vi at Vestre-Agder opplever noe økning i sommermånedene, da dette er ett mindre populært feriested. Derfor kan turistskatt her ha mer negativ effekt enn andre



mer populære destinasjoner.

### Illustrasjon med antagelser

In [ ]: *# Definer funksjoner*

```
def etterspørsel(p,T):
    return 2000 - 20*(p+T)

def tilbud(p,t):
    return 0+20*(p-t)
```

In [ ]: `p_num=np.linspace(0.1,100,100)`

```
# Plot figuren
fig11, ax = plt.subplots(figsize=(18,6))

# Plot data
ax.plot(etterspørsel(p_num,0), label='Etterspørsel', color='red')
ax.plot(tilbud(p_num,0), label='Tilbud', color='blue')

# Sett opp x-aksen
ax.set_xticks(np.arange(0, 110, 10))

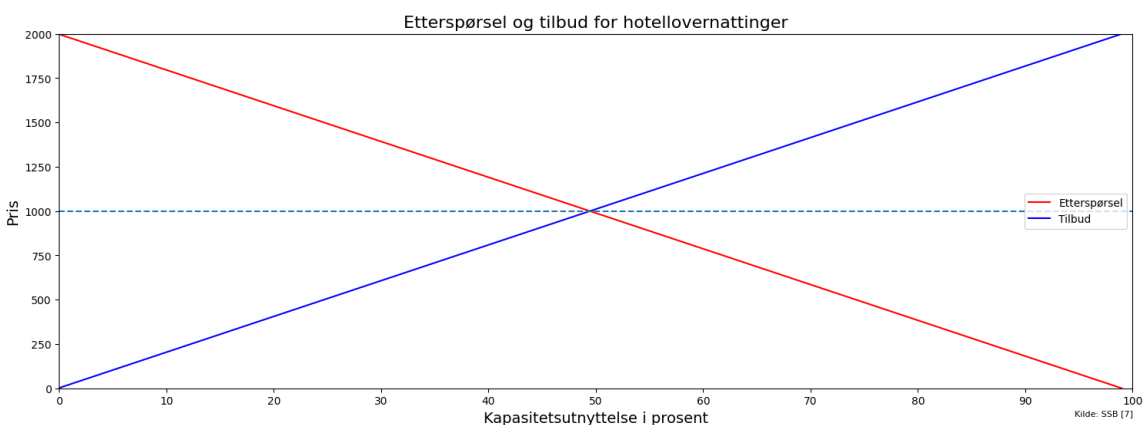
# Sett opp tittel, labels og limits
ax.set_title('Etterspørsel og tilbud for hotellovernattinger', fontsize=16)
ax.set_xlabel('Kapasitetsutnyttelse i prosent', fontsize=14)
ax.set_ylabel('Pris', fontsize=14)
ax.set_xlim(0, 100)
ax.set_ylim(0, 2000)

# Sett opp linje
ax.hlines(1000, 0, 100, linestyle='dashed')

# Kilde
plt.annotate('Kilde: SSB [7]', (0,0), (950,-20), fontsize=8, xycoords='axes fraction')

ax.legend()

plt.show()
```



I foregående figur vil jeg forsøke å illustrere markedet slik det er i dag. Jeg har antatt en tilbuds- og en etterspørselsfunksjon, hvor vi vil finne likevekten for overnattinger med en pris på 1 000 kr, og en kapasitetsutnyttelse på 49.5%.

Etterspørselsfunksjonen viser oss at jo lavere prisen blir, jo flere vil benytte seg av overnattingene, derfor heller denne nedover.

Mens tilbudsfunksjonen viser oss at jo høyere pris gjestene er villige til å betale, jo flere rom er hotellene villige til å tilby.

Det kan nevnes at funksjoner med full informasjon, og som inkluderer alle typer rom, vil kunne fremstille betraktelig høyere priser. Men her ser i på ett aggregert gjennomsnitt, altså alle rom samlet med gjennomsnittspriser.

```
In [ ]: # Lag figuren
fig12, ax = plt.subplots(figsize=(18,6))

# Plot data
ax.plot(etterspørsel(p_num,0), label='Etterspørsel', color='red')
ax.plot(tilbud(p_num,0), label='Tilbud', color='blue')
ax.plot(etterspørsel(p_num,10), label='Etterspørsel med 10% skatt', color='green')
ax.plot(tilbud(p_num,-10), label='Tilbud med 10% skatt', color='orange')

# Sett opp x-aksen
ax.set_xticks(np.arange(30, 70, 5))

# Sett opp tittel, Labels og Limits
ax.set_title('Etterspørsel og tilbud for hotellovernattinger', fontsize=16)
ax.set_xlabel('Kapasitetsutnyttelse i prosent', fontsize=14)
ax.set_ylabel('Pris', fontsize=14)
ax.set_xlim(30, 60)
ax.set_ylim(600, 1400)

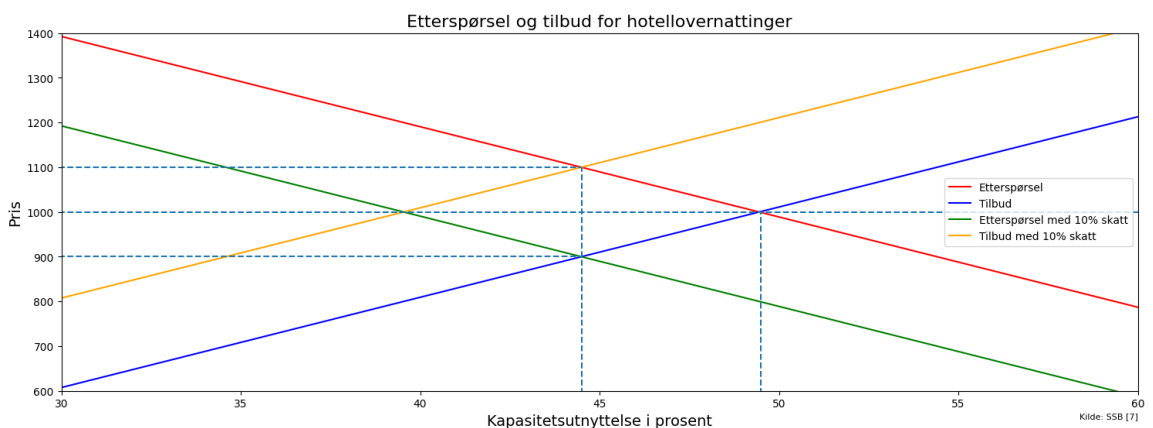
# Sett opp linjer
ax.hlines(1000, 0, 100, linestyles='dashed')
ax.hlines(1100, 0, 44.5, linestyles='dashed')
ax.hlines(900, 0, 44.5, linestyles='dashed')

ax.vlines(49.5, 0, 1000, linestyles='dashed')
ax.vlines(44.5, 0, 1100, linestyles='dashed')

# Kilde
plt.annotate('Kilde: SSB [7]', (0,0), (950,-20), fontsize=8, xycoords='axes frac')

ax.legend()

plt.show()
```



I foregående figur ser vi hva som kan være effekten av en 10% turistskatt på overnattinger. Prisen vil stige, og antall overnattinger vil falle.

Prisen gjestene vil betale blir 1100 kr, mens inntektene til hotellene blir 900 kr, og det vil etterspørres 44.5% av kapasiteten.

---

"Panelet består av 12 avreiseland som vi observerer i perioden 1990 til 2007."

"Begge gruppene er uelastiske i etterspørselen etter en norgesferie med en priselastisitet for nabolandene på -0.7 og for avreiselandene på -0.8."

"Disse tolv landene står til sammen for 87 prosent av alle utenlandske overnattinger på norske hotell/overnattingsbedrifter i perioden 1990 til 2007."

"Landene vi bruker observasjoner for er Danmark, Sverige, Finland, Tyskland, Frankrike, Storbritannia, USA, Italia, Nederland, Spania, Japan og Sveits."

- Andreas Slettebak Wangen, Trondheim, 1. juni 2009,  
Masteroppgave [8]

---

Som vi kan se fra Wagen sin masteroppgave, riktignok med noe eldre informasjon, er gjestene delvis uelastiske i sin beslutning om å konsumere overnattinger i Norge.

Priselastisiteten måler endringen i etterspørselen etter ett gode ved en 1% endring i pris.

En elastisitet mellom -1 og 0, ved en 1% økning i pris, betyr at det vil bli ett likt eller mindre enn 1% fall i etterspørselen.

Med priselastisiteter på -0.7 og -0.8 vil en 1% økning i pris tyde på at vi kan forvente ett fall i etterspørselen på 0.7% og 0.8% respektivt fra de landene som ble undersøkt.

## | 4. Konklusjon |

Det virker tydelig at en turistskatt kan oppnå formålet sitt med å sikre bærekraftig turisme, samt verne om norsk natur og fellesgoder.

Vi ser at det er rom i markedet for prisøkningen en beskatning vil medføre, da store deler av turisme etterspørselen vil falle mindre enn prisøkningen.

Aggregert sett ser Norge ut som om dette er tilfelle i sin helhet, men vi ser også at vi har regioner som vil dra mer nytte av det enn andre, samt noen som potensielt kan svekkes av dette der etterspørselen ikke er like høy.

Mitt råd til Nærings- og fiskeridepartementet vil derfor være å gi kommunene tilgang på turistskatt, slik at lokalsamfunnene kan evaluere om dette er noe de trenger.

Kildereferanser:

[1] NHO Reiseliv

<https://www.nhoreiseliv.no/vi-mener/barekraftig-reiseliv/turistskatt/>

[2] Dagbladet, hentet 23/03-2023

<https://www.dagbladet.no/nyheter/turistene-tisser-pa-verdensarven/71412248>

[3] EU sine nettsider

[https://single-market-economy.ec.europa.eu/sectors/tourism/eu-funding-and-businesses/business-portal/financing-your-business/tourism-related-taxes-across-eu\\_en#specifictaxes](https://single-market-economy.ec.europa.eu/sectors/tourism/eu-funding-and-businesses/business-portal/financing-your-business/tourism-related-taxes-across-eu_en#specifictaxes)

[4] NIKKEI Asia, hentet 23/03-2023

<https://asia.nikkei.com/Business/Travel-Leisure/Bhutan-reopens-with-200-tourist-tax-in-test-for-sustainable-travel>

[5] Statistisk Sentralbyrå, hentet 23/03-2023

<https://www.ssb.no/statbank/table/08403/tableViewLayout1/>

[6] Statistisk Sentralbyrå, hentet 23/03-2023

<https://www.ssb.no/statbank/table/06921/tableViewLayout1/>

[7] Statistisk Sentralbyrå, hentet 23/03-2023

<https://www.ssb.no/statbank/table/13156/tableViewLayout1/>

[8] Andreas Slettebak Wangen, Trondheim, 1. juni 2009, Masteroppgave, hentet 23/03/2023

[https://www.ntnu.no/c/document\\_library/get\\_file?uuid=db48cbdc-9552-4d10-b108-67f04dc03d6d&groupId=10449](https://www.ntnu.no/c/document_library/get_file?uuid=db48cbdc-9552-4d10-b108-67f04dc03d6d&groupId=10449)

Koden benyttet er hovedsaklig min egen, fra tidligere oppgaver, noe fra forelesningsnotater og noen snutter fra <https://stackoverflow.com/>