

# 1 Figurer i python med Matplotlib

Matplotlib er et plottingbibliotek som brukes til å lage diagrammer og plott enkelt. Med Matplotlib kan du generere plott figurer med bare noen få linjer med kode.

Før du begynner med dette dokumentet, anbefaler vi at du leser denne korte introduksjonen til matplotlib [matplotlib\\_introduksjon.pdf](#). Som alltid i programmering, er det ikke nødvendig å huske alt, bare å forstå logikken og å kunne finne relevant informasjon raskt. [matplotlib\\_introduksjon.pdf](#) vil introdusere deg for logikken, mens dette dokumentet vil vise hvordan logikken fungerer i praksis og tilby deg kodelapper du kan gjenbruke og tilpasse for å lage dine fremtidige figurer.

## 1.1 Først trenger vi å importere biblioteket

Før vi kan importere biblioteker, må vi installere dem. Se detaljerte instruksjoner om hvordan du installerer python biblioteker i filen [installere\\_pakker\\_vscode.pdf](#).

```
[1]: # Importer biblioteket vi trenger
import matplotlib.pyplot as plt
```

## 1.2 Tilbake til allergier eksempel

Koden nedenfor kommer fra oppgavene [mod4-del1-2-oppgaver-løsning](#). Den teller antallet personer som har en gitt restriksjon.

```
[2]: # En list over matrestriksjoner fra deltakerregistreringen
mat_restriksjoner = [
    "egg", "nøtter", "kjøtt", "svinn", "fisk",
    "laktose", "fisk", "laktose", "kjøtt", "laktose", "kjøtt",
    "nøtter", "melk", "svinn", "egg", "nøtter", "kjøtt", "laktose",
    "fisk", "kjøtt", "laktose", "melk", "egg", "egg", "melk", "svinn",
    "fisk", "nøtter", "nøtter", "melk", "fisk", "laktose",
    "nøtter", "laktose", "laktose", "gluten", "gluten", "kjøtt",
    "kjøtt", "laktose"
]

def restriksjon_teller(restriksjoner):
    """
    Telle hvor mange personer har en gitt restriksjon.

    Returnere en ordbok som teller antallet personer med en gitt
    restriksjon for hver restriksjon i listen.
    """
    # Opprette en ordbok som kommer til å telle hvor mange personer
    # har en gitt restriksjon
    teller = {}

    # Hent ut hver mat restriksjon i listen over restriksjoner
    for mat in restriksjoner:
        # Hvis vi allerede har funnet denne restriksjonen, øker vi antallet
        if mat in teller:
            teller[mat] = teller[mat] + 1
        # Ellers, oppretter vi en ny nøkkel, og initialiserer verdien til 1
        else:
```

```

        teller[mat] = 1
    return teller

teller = restriksjon_teller(mat_restriksjoner)

print(teller)

```

```

{'egg': 4, 'nøtter': 6, 'kjøtt': 7, 'svinn': 3, 'fisk': 5, 'laktose': 9, 'melk':
4, 'gluten': 2}

```

### 1.3 Stolpediagram (Bar plot)

Se [https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.bar.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html)

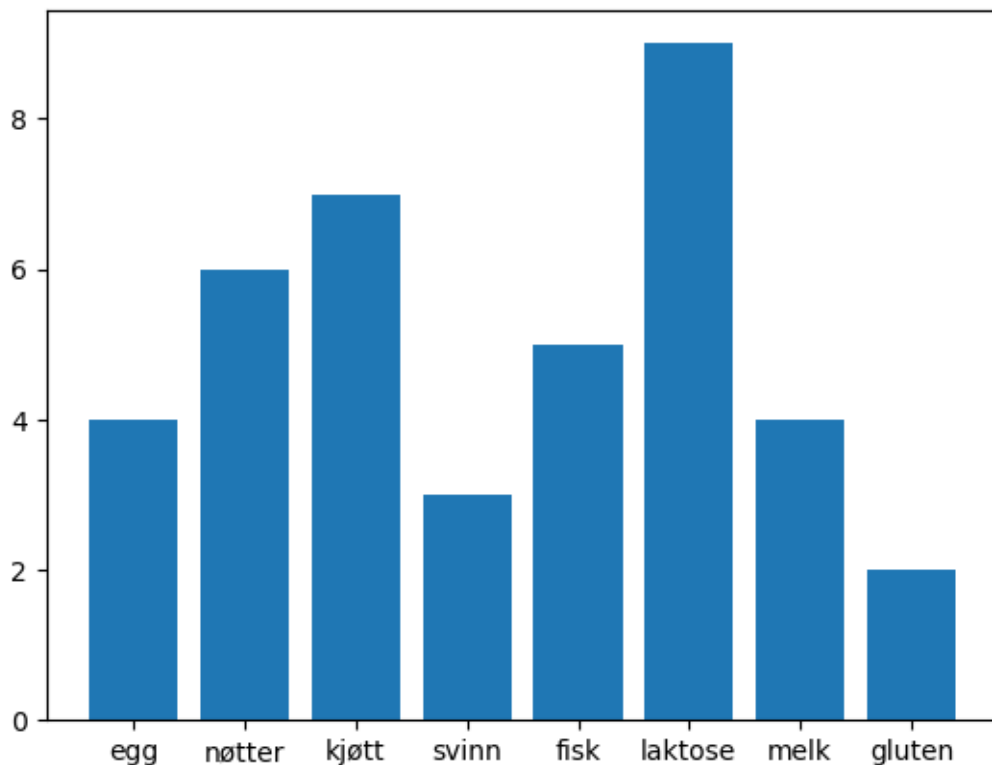
#### 1.3.1 Stolpediagram med en ordbok

```

[3]: # Opprett en tom figur som inneholder en tom akse
fig, ax = plt.subplots()

# Syntaksen: ax.bar(x-verdiene, y-verdiene)
ax.bar(teller.keys(), teller.values())
# plt.show() vise plottet som vi har laget
# plt.show() stopper programmet opp og venter på at vinduet med plottet skal
# lukkes før det fortsetter å kjøre resten av koden.
plt.show()

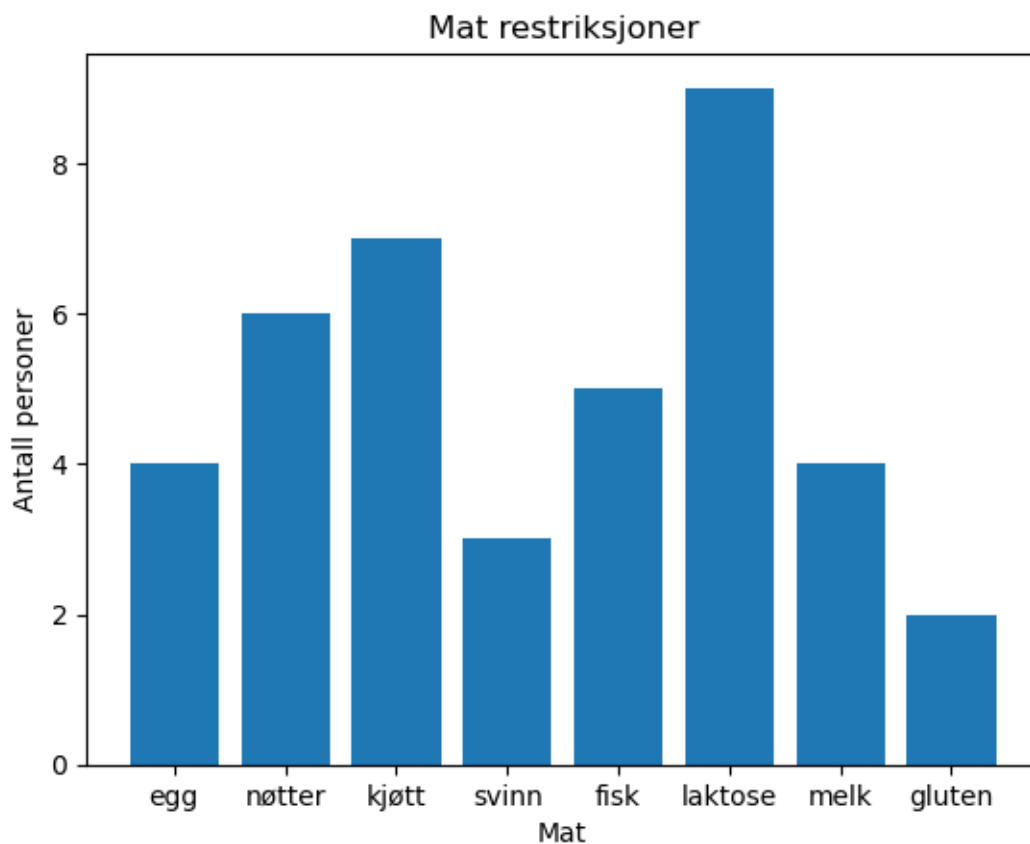
```



I cellen ovenfor har vi brukt de linjene som er strengt nødvendige for å lage et søylediagram. Men det er viktig

å legge til tekst i figurene våre for å gjøre dem lette å lese og forstå. Derfor legger vi til noen valgfrie linjer i cellen nedenfor for å kommentere figuren vår.

```
[4]: # ----- Bruk dataene våre til å lage en figur -----  
# Opprett en tom figur som inneholder en tom akse  
fig, ax = plt.subplots()  
  
# Syntaksen: ax.bar(x-verdiene, y-verdiene)  
ax.bar(teller.keys(), teller.values())  
  
# ----- Legg til tekst til figuren -----  
# Sett x-akse etikett  
ax.set_xlabel("Mat")  
# Sett y-akse etikett  
ax.set_ylabel("Antall personer")  
# Sett tittelen til figuren  
ax.set_title("Mat restriksjoner")  
  
# ----- Lagr figuren -----  
fig.savefig("mat_restriksjoner_figur")
```



### 1.3.2 Stolpediagram med lister

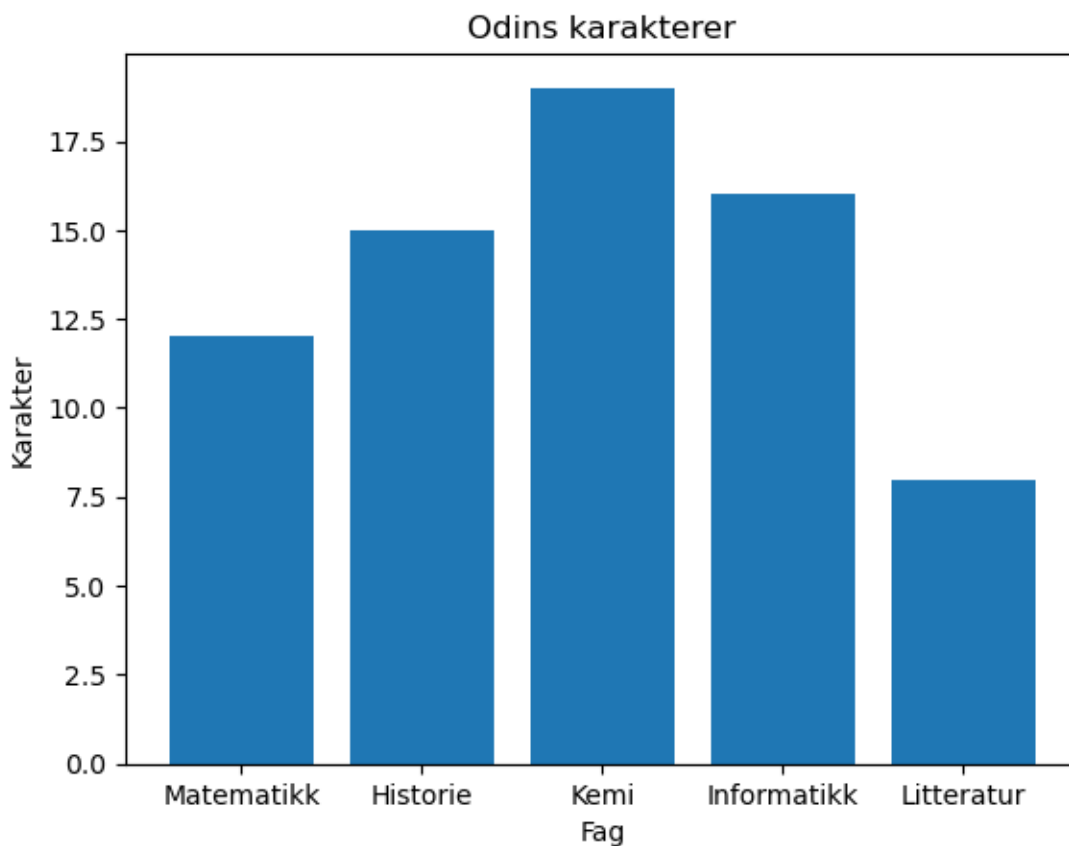
Eksempel: En students karakterer i forskjellige fag.

```
[5]: # ----- Data -----
fag = ["Matematikk", "Historie", "Kemi", "Informatikk", "Litteratur"]
karakterer = [12, 15, 19, 16, 8]

# ----- Bruk dataene våre til å lage en figur -----
fig, ax = plt.subplots()
ax.bar(x=fag, height=karakterer)

# ----- Legg til tekst til figuren -----
ax.set_xlabel("Fag")
ax.set_ylabel("Karakter")
ax.set_title("Odins karakterer")

# ----- Lagr figuren -----
fig.savefig("Odins_karakterer")
```



## 1.4 Linjediagram (Line plot)

Eksempel: Utviklingen av en students karakterer gjennom årene.

Se [https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.plot.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html)

```
[6]: # ----- Data -----
karakterer_kemi = [12, 15, 19, 16, 8]
karakterer_info = [9, 12, 18, 20, 19.5]
```

```

# Initierer en liste med [2015, 2016, 2017, 2018, 2019]
år = []
for året in range(2015,2020):
    år.append(året)

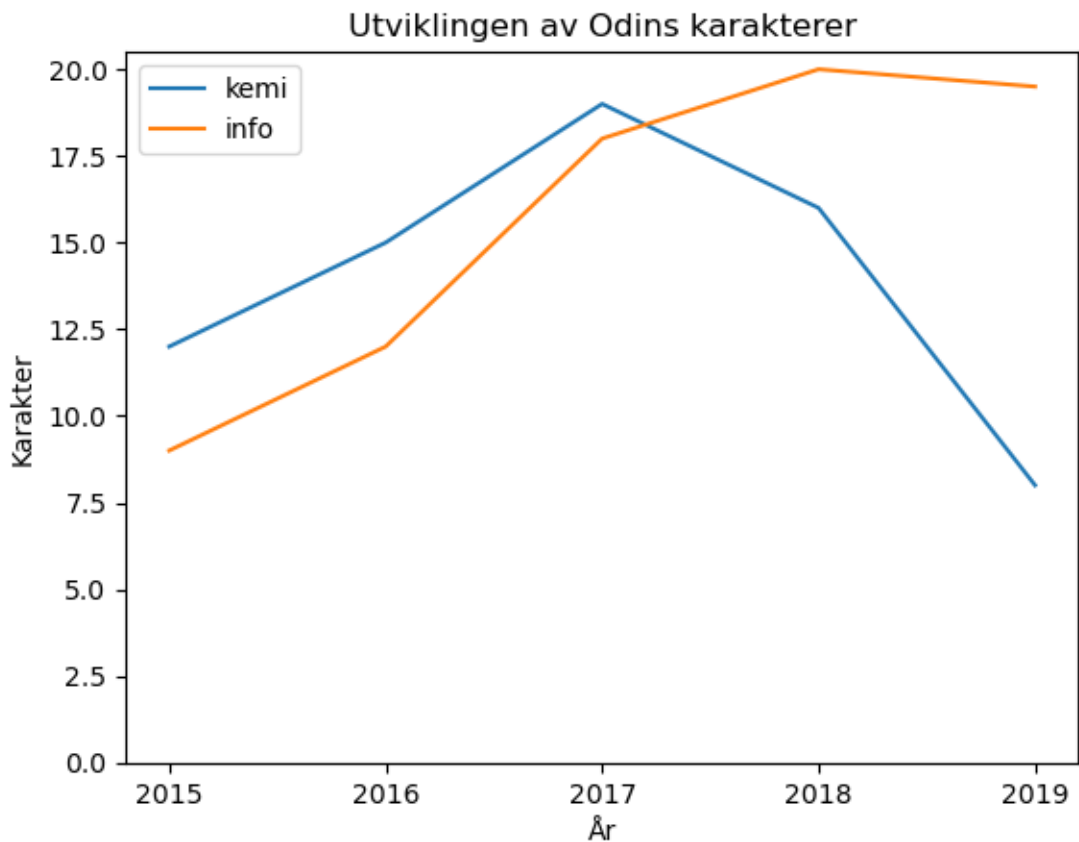
# ----- Bruk dataene våre til å lage en figur -----
# Opprett en tom figur med som inneholder en tom akse
fig, ax = plt.subplots()

# Syntaksen: ax.plot(x-verdiene, y-verdiene)
ax.plot(år, karakterer_kemi, label="kemi")
# Vi kan legge til flere linjer i et gitt diagrammet
# Syntaksen: ax.plot(x-verdiene, y-verdiene, label="marken")
ax.plot(år, karakterer_info, label="info")

# ----- Legg til tekst til figuren -----
ax.set_xlabel("År")
ax.set_ylabel("Karakter")
# For å vise bare 5 ticks som tilsvarer årene
ax.set_xticks(år)
ax.set_ylim(0, 20.5)
# Nå at vi her flere linjer og et merket for hver linje, bør vi
# legge til en figurtekst med markene.
ax.legend()
ax.set_title("Utviklingen av Odins karakterer")

# ----- Lagr figuren -----
fig.savefig("Utviklingen_Odins_karakterer")

```



## 1.5 Spredningsplott (Scatter plot)

Eksempel: Lengde og vekt av babyer ved fødsel.

Se [https://matplotlib.org/stable/api/\\_as\\_gen/matplotlib.pyplot.plot.html](https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html)

```
[7]: # ----- Data -----
# Lengde og vekt av babyer
gutt_lengde = [
    47.3, 49.4, 48.0, 50.7, 51.1, 50.5, 51.1, 49.3, 47.8, 49.0, 49.4,
    50.0, 47.9, 49.0, 49.0, 49.4, 47.6, 49.2, 48.5, 49.0
]
gutt_vekt = [
    3.06, 3.19, 3.1, 3.31, 3.28, 3.26, 3.3, 3.21, 3.08, 3.17, 3.18, 3.25,
    3.08, 3.14, 3.16, 3.19, 3.08, 3.17, 3.12, 3.15
]

jente_lengde = [
    49.2, 47.8, 48.7, 47.1, 47.5, 48.5, 47.0, 47.8, 47.7, 46.9, 47.5, 47.3, 47.9, 48.0, ↵
    ↵47.7, 49.5, 48.2, 46.3, 47.5, 47.5
]
jente_vekt = [
    3.21, 3.12, 3.12, 3.04, 3.1, 3.09, 3.04, 3.07, 3.08, 2.99, 3.05, 3.1, 3.1, 3.1, 3.09, ↵
    ↵3.18, 3.09, 3.02, 3.06, 3.06
]
```

