

ISTx1001: Industriell statistikk

13. november 2023

Vi starter klokka 15:15

Timen blir tatt opp :)

Thea Bjørnland, Institutt for matematiske fag, NTNU

Del 2 av prosjektet i ISTx1001

Ressurser:

Læreboka kapittel 11

Temavideoer (laget av Ketil Arnesen)

Forelesninger med slides

Jupyter-notatbok **papirbro.ipynb**

Statistisk kvalitetskontroll

- [Shewharts kontrolldiagram](#)
- [s-diagram og p-diagram](#)
- [Karakteristikk](#)
- [Kapabilitet](#)

Del 2 av prosjektet i ISTx1001

Oppgave 2 - Statistisk prosesskontroll

Målet med oppgaven er å bestemme kontrollgrenser i en prosesskontroll, samt visualisere og forklare hva dette innebærer ved hjelp av stokastiske simuleringer.

Tenk dere at dere skal lage et produkt basert på forsøket dere gjennomførte i Oppgave 1, f.eks muffins, skråplan med ball eller papirhelikopter. Dere må selv velge nivåer for forklaringsvariablene som produktet skal produseres under. En stor produksjon skal settes i gang, og responsvariabelen fra Oppgave 1 skal brukes for å kontrollere (med stikkprøver) at prosessen er under kontroll.

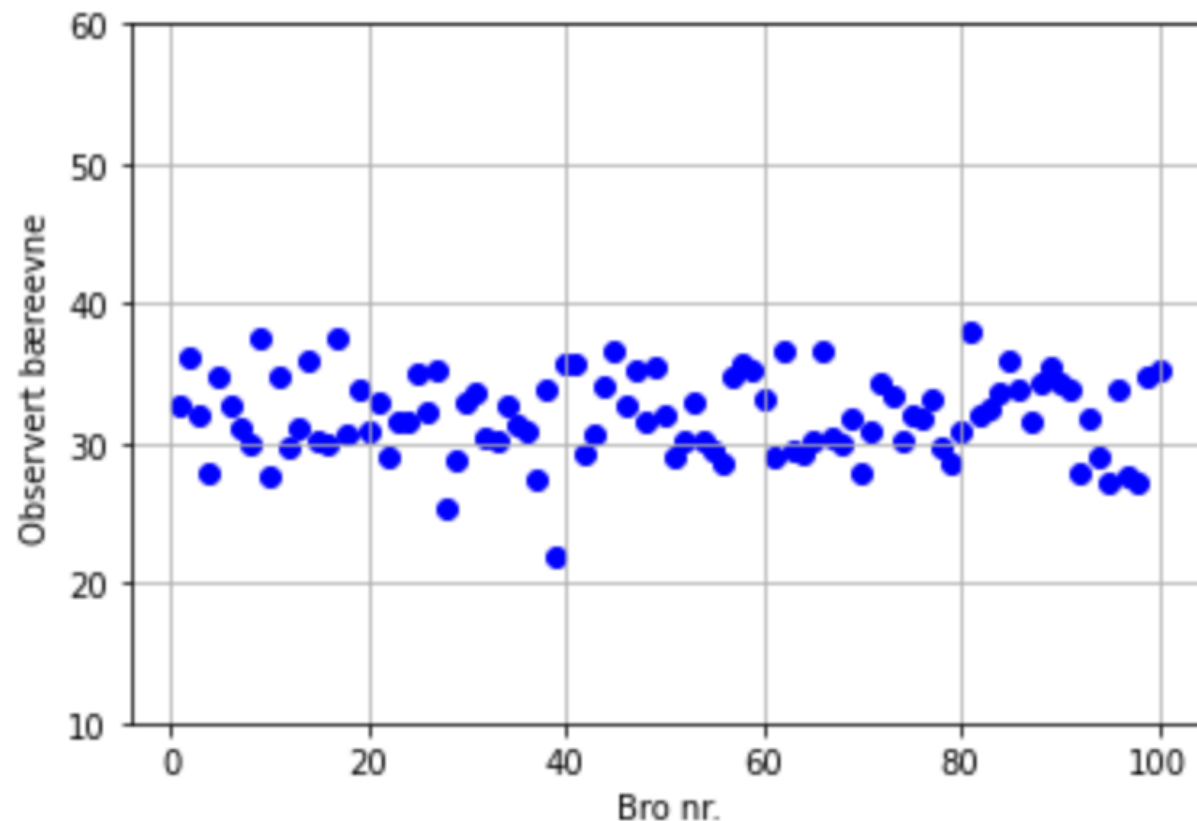
Repetisjon: Stokastisk simulering av normal prosess

```
mean = 31.81
std_dev = 2.92
antall_broer = 100
bæreevne = np.zeros(antall_broer)

for i in range(antall_broer):
    bæreevne[i] = np.random.normal(mean, std_dev)

bro_nr = np.arange(1, antall_broer + 1)

plt.scatter(bro_nr, bæreevne, marker='o', color='b')
plt.ylim(10,60); plt.xlabel("Bro nr."); plt.ylabel("Observert bæreevne")
plt.grid()
plt.show()
```



Repetisjon: Stokastisk simulering: hva kan gå galt?

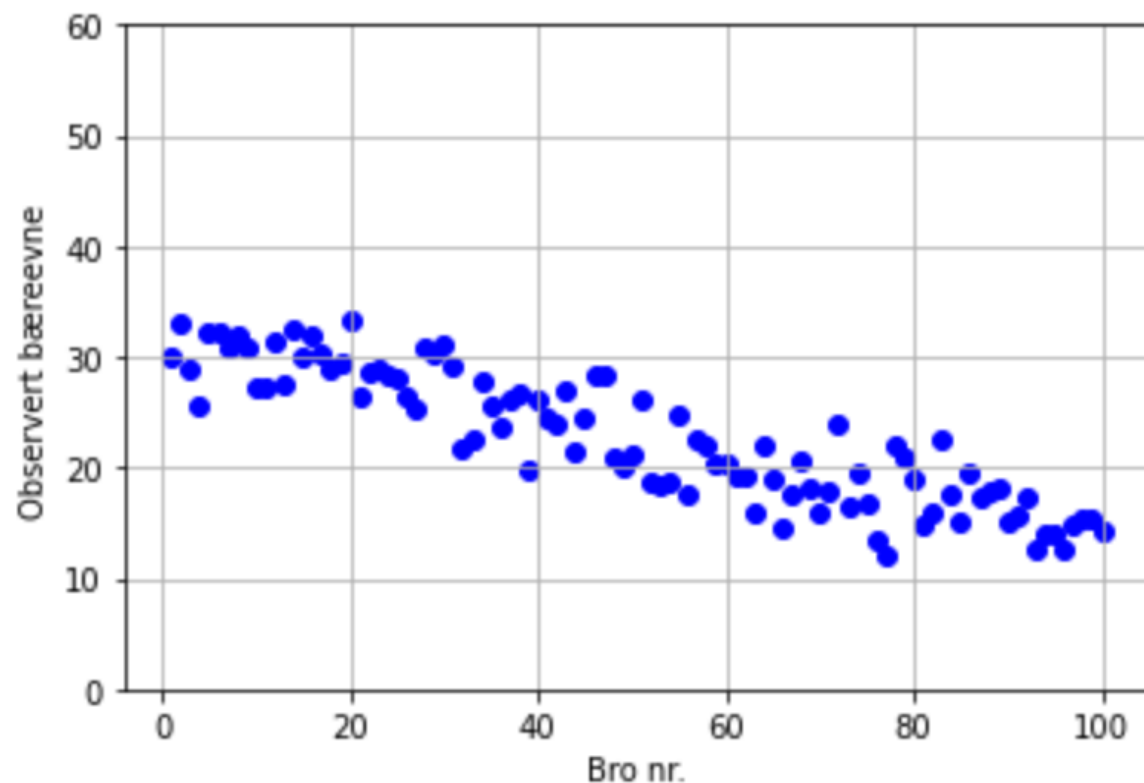
Repetisjon: Stokastisk simulering: hva kan gå galt?

```
mean = 31.81
std_dev = 2.92
antall_broer = 100
bæreevne = np.zeros(antall_broer)

for i in range(antall_broer):
    if (i + 1) % 5 == 0:
        mean -= 0.3 * std_dev
        bæreevne[i] = np.random.normal(mean, std_dev)

bro_nr = np.arange(1, antall_broer + 1)

plt.scatter(bro_nr, bæreevne, marker='o', color='b')
plt.ylim(0,60); plt.xlabel("Bro nr."); plt.ylabel("Observert bæreevne")
plt.grid()
plt.show()
```



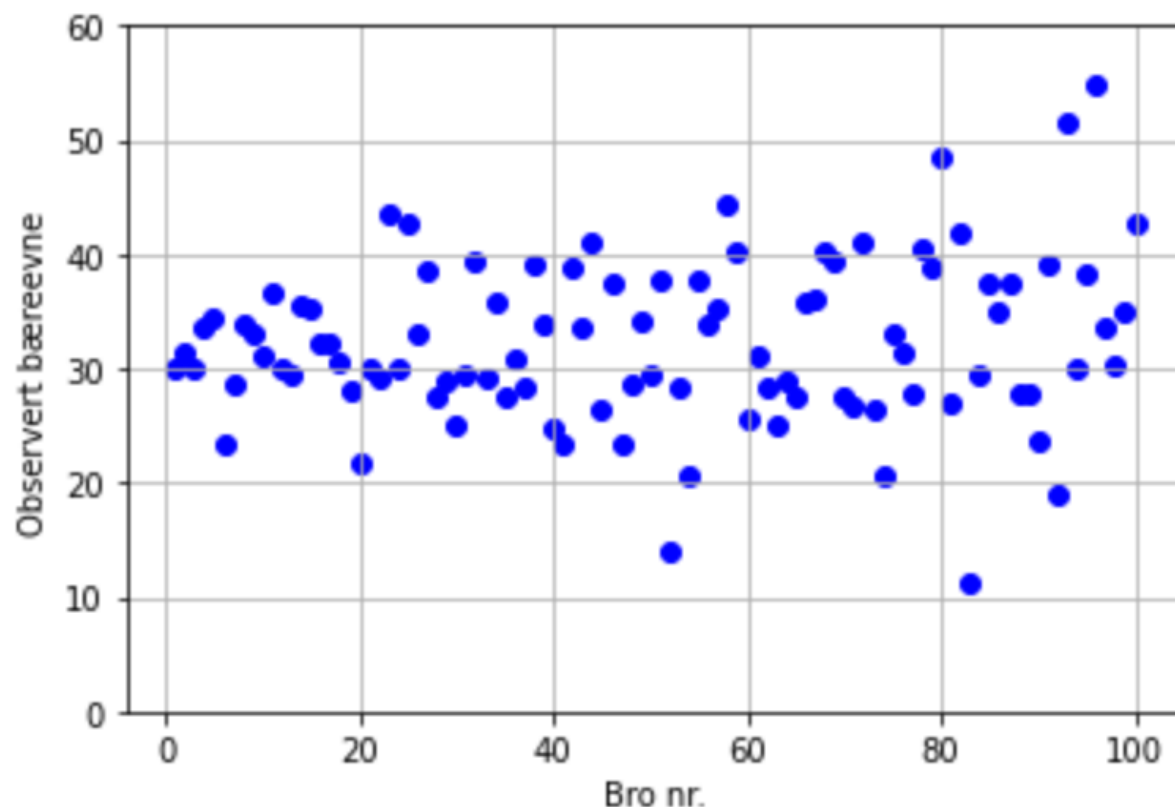
Repetisjon: Stokastisk simulering: hva kan gå galt?

```
mean = 31.81
std_dev = 2.92
antall_broer = 100
bæreevne = np.zeros(antall_broer)

for i in range(antall_broer):
    if (i + 1) % 5 == 0:
        std_dev += 0.5
        bæreevne[i] = np.random.normal(mean, std_dev)

bro_nr = np.arange(1, antall_broer + 1)

plt.scatter(bro_nr, bæreevne, marker='o', color='b')
plt.ylim(0,60); plt.xlabel("Bro nr."); plt.ylabel("Observert bæreevne")
plt.grid()
plt.show()
```



Stokastisk simulering: hva kan gå galt?

Vi må følge med på forventningsverdi og standardavvik!

Del 2 av prosjektet i ISTx1001

2a) Kontrollgrenser

Finn en estimert forventningsverdi og et standardavvik på Y fra Oppgave 1 for den valgte faktorkombinasjonen. Dette setter dere som verdier på μ og σ for en prosess som er under kontroll. Bestem hvor mange (n) produkter dere vil kontrollere i hver stikkprøve.

Bestem kontrollgrenser for gjennomsnittet til en stikkprøve. Bruk Shewarts' anbefaling om 3-sigma-grenser.

Bestem kontrollgrenser for det empiriske standardavviket til en stikkprøve basert på kritiske verdier fra kjikvadratfordelingen med $\alpha/2 = 0.002$.

Del 2 av prosjektet i ISTx1001

```
mean = 31.81
std_dev = 2.92

n_stikkprøve = 5
cu_mean = mean + 3*std_dev/np.sqrt(n_stikkprøve)
cl_mean = mean - 3*std_dev/np.sqrt(n_stikkprøve)

print(cu_mean)
print(cl_mean)

cu_std = std_dev*np.sqrt(stats.chi2.ppf(1-0.002,(n_stikkprøve-1))/(n_stikkprøve-1))
cl_std = std_dev*np.sqrt(stats.chi2.ppf(0.002,(n_stikkprøve-1))/(n_stikkprøve-1))
print(cu_std)
print(cl_std)
```

```
35.72759109657963
27.892408903420367
6.006220356445241
0.5248648521370621
```

Del 2 av prosjektet i ISTx1001

2b) Stokastisk simulering

Gjennomfør to simuleringer av 50 stikkprøver med Python. For hver av simuleringene skal dere plotte et kombinert $\bar{x}$ -s-diagram og forklare hva vi ser i diagrammet.

I den første simuleringen skal dere trekke n responsverdier y for hver stikkprøve. Disse trekkes fra en normalfordeling med forventning μ og standardavvik σ som spesifisert over. Regn ut $\bar{x}$ og s for hver av de 50 stikkprøvene og visualiser i diagrammet sammen med kontrollgrensene.

I den andre simuleringen skal μ gradvis økes ved å legge til 0.5σ etter hver tiende stikkprøve. Presenter simuleringen i et $\bar{x}$ -s-diagram og kommenter hva dere ser. Når går alarmen?

Presenter arbeidet deres (inkl. figurer) på maksimalt én A4-side.

Stikkprøver: Stokastisk simulering av normal prosess

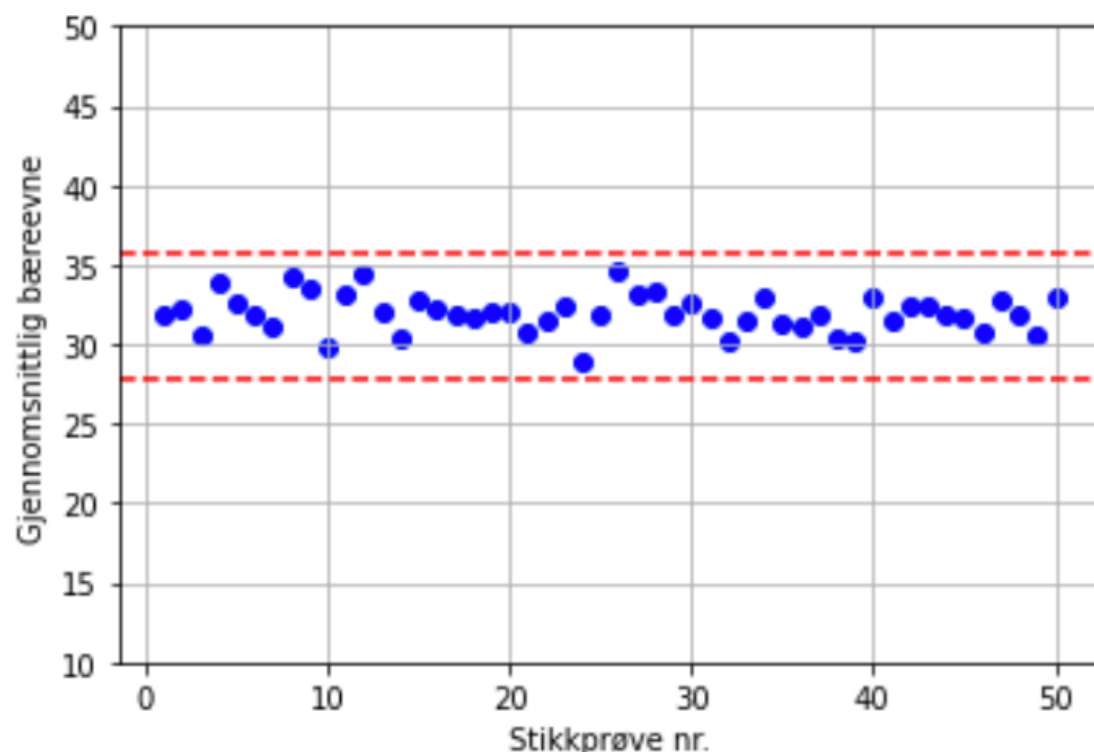
```
stikkprøver = 50
mean = 31.81
std_dev = 2.92

snitt_bæreevne = np.zeros(stikkprøver)
std_bæreevne = np.zeros(stikkprøver)

for i in range(stikkprøver):
    utvalg = np.random.normal(mean, std_dev, n_stikkprøve)
    snitt_bæreevne[i] = np.mean(utvalg)
    std_bæreevne[i] = np.std(utvalg, ddof=1)

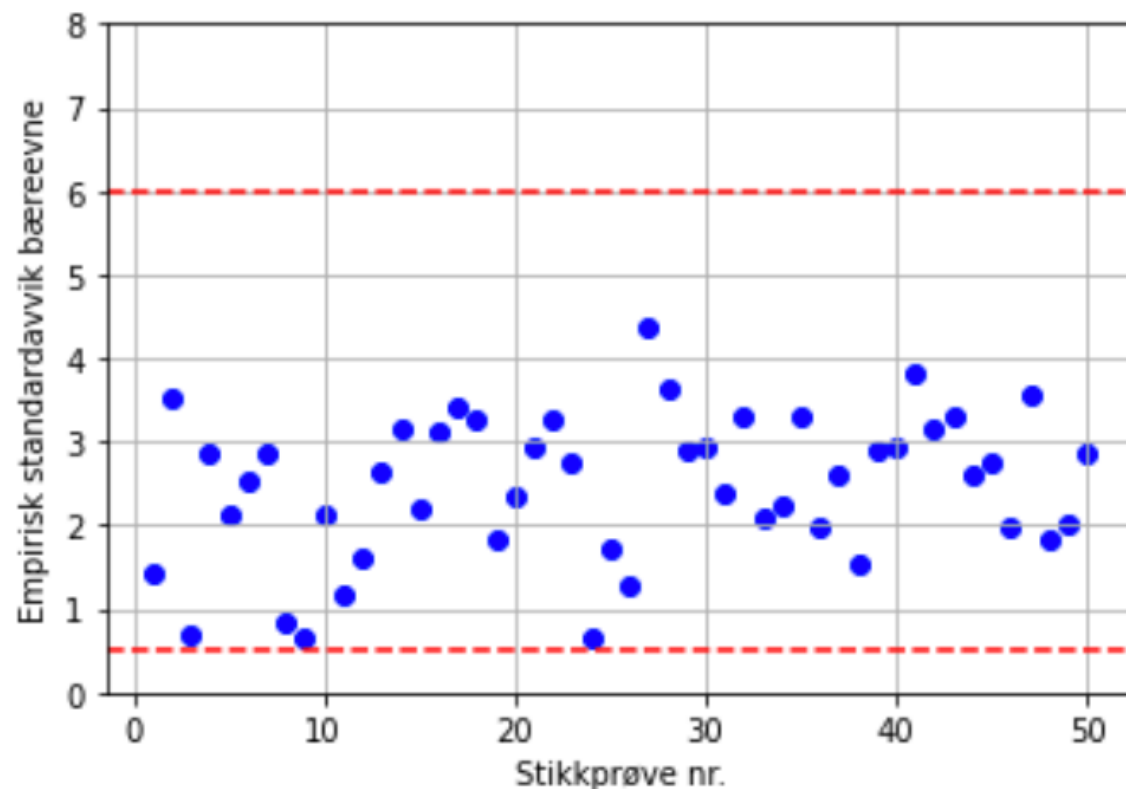
stikkprøve_nr = np.arange(1, stikkprøver + 1)

plt.scatter(stikkprøve_nr, snitt_bæreevne, marker='o', color='b')
plt.ylim(10,50); plt.xlabel("Stikkprøve nr."); plt.ylabel("Gjennomsnittlig bæreevne")
plt.axhline(y=cu_mean, color='r', linestyle='--'); plt.axhline(y=cl_mean, color='r', linestyle='--')
plt.grid()
plt.show()
```



Stikkprøver: Stokastisk simulering av normal prosess

```
plt.scatter(stikkprøve_nr, std_bæreevne, marker='o', color='b')
plt.ylim(0,8); plt.xlabel("Stikkprøve nr."); plt.ylabel("Empirisk standardavvik bæreevne")
plt.axhline(y=cu_std, color='r', linestyle='--')
plt.axhline(y=cl_std, color='r', linestyle='--')
plt.grid()
plt.show()
```



Stikkprøver: Stokastisk simulering når “noe” går galt

```
stikkprøver = 50

mean = 31.81
std_dev = 2.92

snitt_bæreevne = np.zeros(stikkprøver)
std_bæreevne = np.zeros(stikkprøver)

for i in range(stikkprøver):
    if (i + 1) % 5 == 0:
        mean += 0.5
        std_dev += 0.1
    utvalg = np.random.normal(mean, std_dev, n_stikkprøve)
    snitt_bæreevne[i] = np.mean(utvalg)
    std_bæreevne[i] = np.std(utvalg, ddof=1)
```

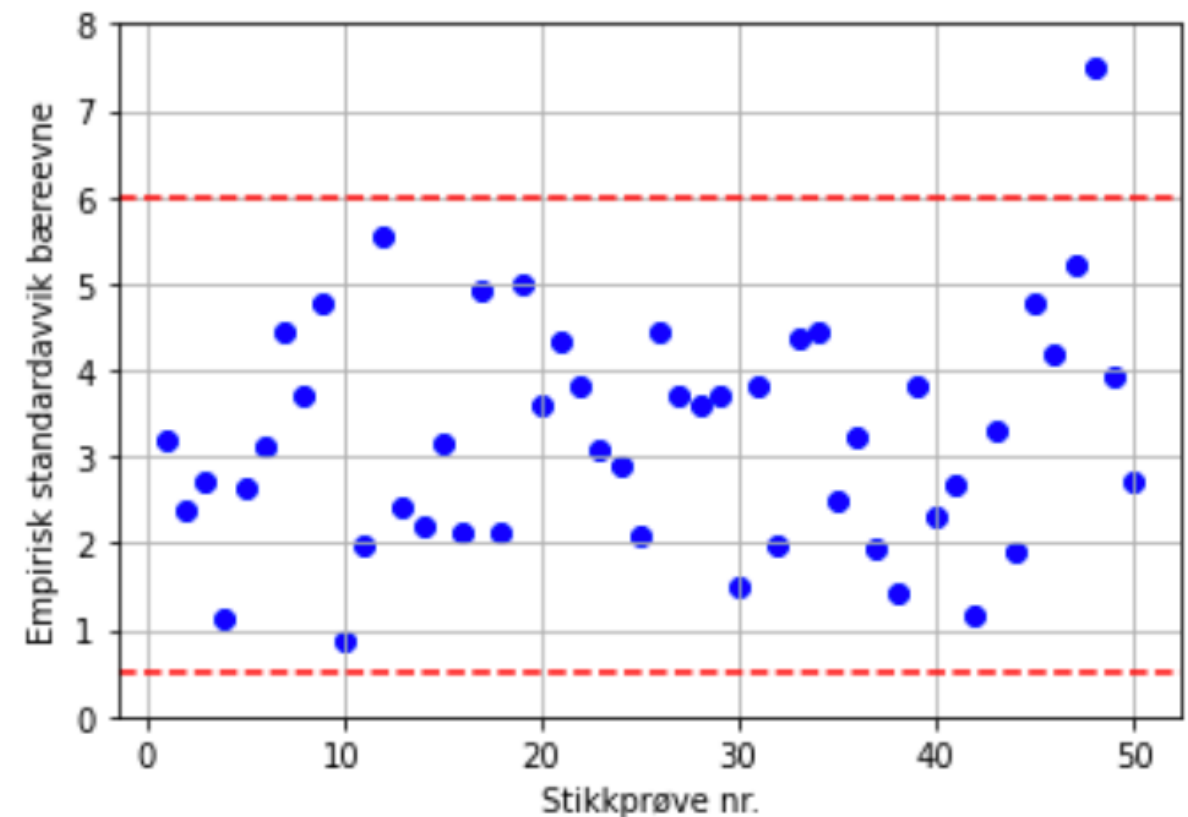
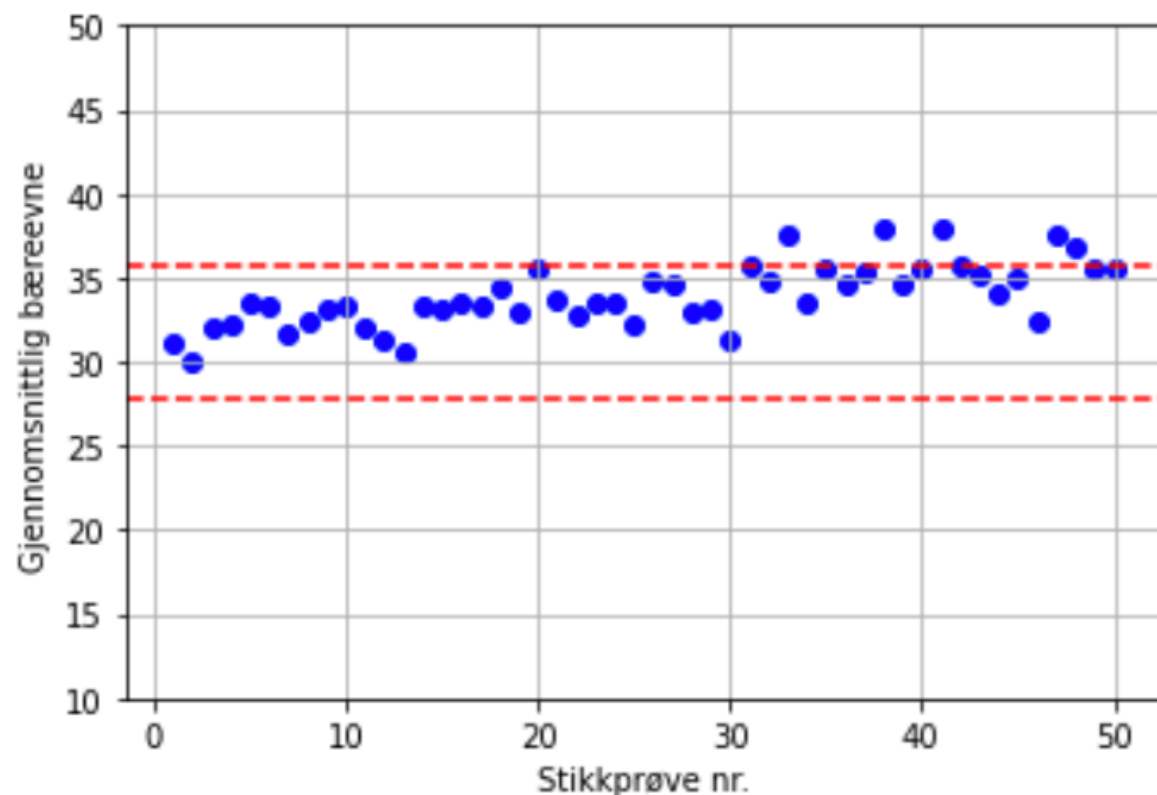
Stikkprøver: Stokastisk simulering når “noe” går galt

```
stikkprøver = 50

mean = 31.81
std_dev = 2.92

snitt_bæreevne = np.zeros(stikkprøver)
std_bæreevne = np.zeros(stikkprøver)

for i in range(stikkprøver):
    if (i + 1) % 5 == 0:
        mean += 0.5
        std_dev += 0.1
    utvalg = np.random.normal(mean, std_dev, n_stikkprøve)
    snitt_bæreevne[i] = np.mean(utvalg)
    std_bæreevne[i] = np.std(utvalg, ddof=1)
```



Del 2 av prosjektet i ISTx1001

2b) Stokastisk simulering

Gjennomfør to simuleringer av 50 stikkprøver med Python. For hver av simuleringene skal dere plotte et kombinert $\bar{x}$ -s-diagram og forklare hva vi ser i diagrammet.

I den første simuleringen skal dere trekke n responsverdier y for hver stikkprøve. Disse trekkes fra en normalfordeling med forventning μ og standardavvik σ som spesifisert over. Regn ut $\bar{x}$ og s for hver av de 50 stikkprøvene og visualiser i diagrammet sammen med kontrollgrensene.

I den andre simuleringen skal μ gradvis økes ved å legge til 0.5σ etter hver tiende stikkprøve. Presenter simuleringen i et $\bar{x}$ -s-diagram og kommenter hva dere ser. Når går alarmen?

Presenter arbeidet deres (inkl. figurer) på maksimalt én A4-side.