STK-IN1050 - Statistikk for informatikere

9. Kontinuerlige fordelinger

Matematisk institutt

2026-02-18

Innledning

Vi skal se på tre grunnleggende kontinuerlige sannsynlighetsfordelinger for å beskrive fordelingen til kontinuerlige stokastiske variabler:

- uniform
- eksponensiell
- normal (Gaussisk)

Sentrale begreper for å beskrive sannsynlighetsfordelinger:

- tetthetshetsfunksjon: $f(x)$
- fordelingsfunksjon: $P(X \leq x) = F(x)$
- forventning: $E(X)$
- varians: $\text{Var}(X)$

Uniformfordeling

Anta at X er en kontinuerlig stokastisk variabel med verdimengde $\mathcal{X} = [a, b]$ (for $a \leq b$) hvor alle verdiene på intervallet $[a, b]$ er like sannsynlige.

Da sier vi at X er **uniform fordelt** på $[a, b]$ og skriver $X \sim U([a, b])$. Da er

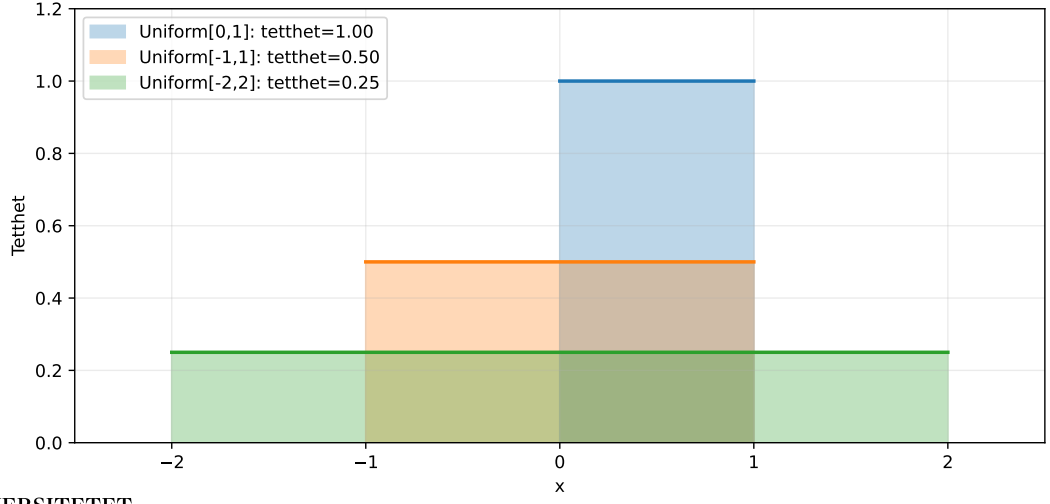
$$f(x) = \begin{cases} \frac{1}{b-a}, & a \leq x \leq b \\ 0, & \text{ellers} \end{cases}$$

og

$$P(c \leq X \leq d) = \int_c^d \frac{1}{b-a} dx = \frac{d-c}{b-a}, \quad \text{for alle } c \text{ og } d \text{ med } a \leq c \leq d \leq b.$$

$X \sim U([a, b])$ har forventning $\mathbb{E}(X) = \frac{a+b}{2}$ og varians $\text{Var}(X) = \frac{(b-a)^2}{12}$.

Uniformfordeling



Uniformfordeling i scipy.stats

```python
import numpy as np
from scipy.stats import uniform

a = 0.0
b = 20.0
loc = a
scale = b - a

dist = uniform(loc=loc, scale=scale)

print(f"E(X) = {dist.mean():.1f}")
print(f"Var(X) = {dist.var():.2f}")

points = np.array([-5.0, 0.0, 5.0, 10.0, 20.0, 25.0])
pdf_vals = dist.pdf(points)
cdf_vals = dist.cdf(points)
print("\nVerdi\tPDF\t\tCDF")
for p, pdf_val, cdf_val in zip(points, pdf_vals, cdf_vals):
    print(f"{p:5.1f}\t{pdf_val:0.3f}\t{cdf_val:0.3f}")
```

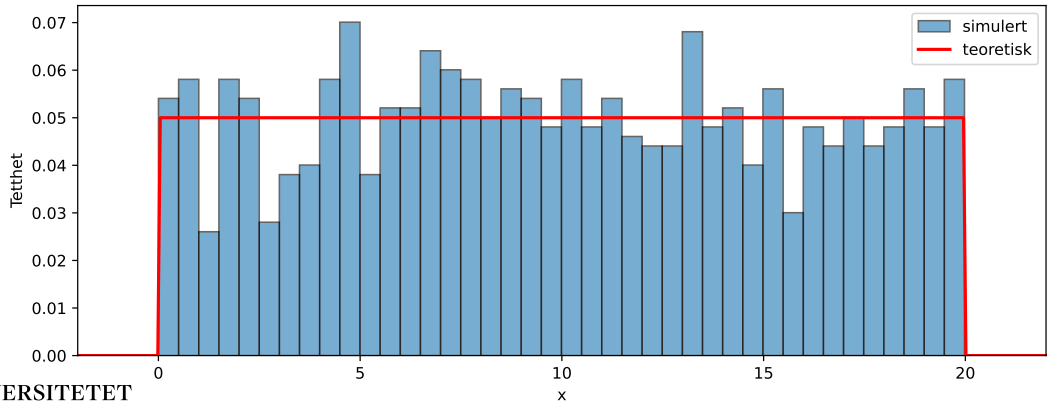
```
E(X) = 10.0
Var(X) = 33.33

Verdi    PDF      CDF
 -5.0    0.000    0.000
  0.0    0.050    0.000
  5.0    0.050    0.250
 10.0    0.050    0.500
 20.0    0.050    1.000
 25.0    0.000    1.000
```

pdf = probability density function
= tetthetsfunksjonen

Uniformfordeling i scipy.stats

```
rng_seed = 123
n_samples = 1000
samples = dist.rvs(size=n_samples, random_state=rng_seed)
```



Eksempel: Ventetid på bussholdeplassen

Du ankommer bussholdeplassen på et helt tilfeldig tidspunkt.

En buss kommer **nøyaktig hvert 10. minutt**.

La X være den stokastiske variabelen «tid du må vente på bussen» (målt i minutter).

Da er $X \sim U([0, 10])$ med

$$\mathbb{E}(X) = \frac{0 + 10}{2} = 5 \text{ minutter}$$

$$Var(X) = \frac{10^2}{12} = 8.3 \text{ minutter}^2$$

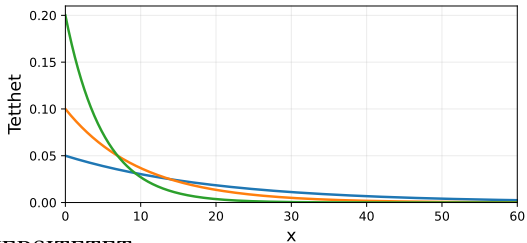
$$sd(X) = \sqrt{8.3} = 2.9 \text{ minutter}$$

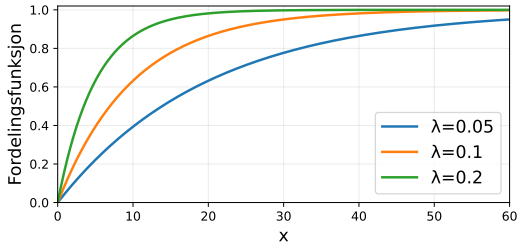
Eksponensiell fordeling

Anta at X er en kontinuerlig stokastisk variabel med $\mathcal{X} = (0, +\infty)$, dvs. X kan ta alle positive verdier.

X har **eksponensiell fordeling**, $X \sim Exp(\lambda)$, hvis

$$f(x) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0, \end{cases} \qquad F(x) = \int_0^x f(t)dt = \begin{cases} 1 - e^{-\lambda x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$





Eksponensiell fordeling i scipy.stats

```python
import numpy as np
from scipy.stats import expon

lam = 0.1
scale = 1/lam

dist = expon(scale=scale)

print(f"E(X) = {dist.mean():.1f}")
print(f"Var(X) = {dist.var():.2f}")

points = np.array([-5.0, 0.0, 5.0, 10.0, 20.0, 25.0])
pdf_vals = dist.pdf(points)
cdf_vals = dist.cdf(points)
print("\nVerdi\tPDF\t\tCDF")
for p, pdf_val, cdf_val in zip(points, pdf_vals, cdf_vals):
    print(f"{p:5.1f}\t{pdf_val:0.3f}\t{cdf_val:0.3f}")
```

```
E(X) = 10.0
Var(X) = 100.00

Verdi    PDF      CDF
 -5.0    0.000    0.000
  0.0    0.100    0.000
  5.0    0.061    0.393
 10.0    0.037    0.632
 20.0    0.014    0.865
 25.0    0.008    0.918
```

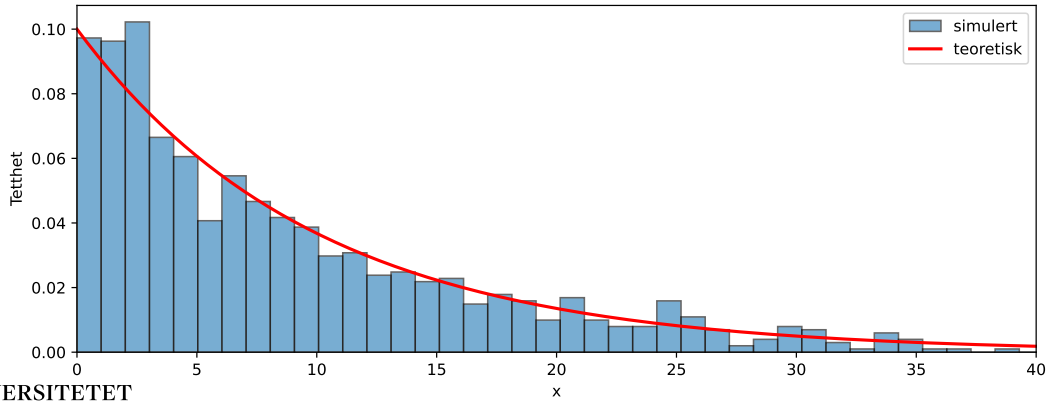
Generelt:

$$\mathbb{E}(X) = 1/\lambda$$

$$Var(X) = 1/\lambda^2$$

Eksponensiell fordeling i scipy.stats

```
rng_seed = 13
n_samples = 1000
samples = dist.rvs(size=n_samples, random_state=rng_seed)
```



Eksempel: Ventetid på bussholdeplassen

Du ankommer bussholdeplassen på et helt tilfeldig tidspunkt.

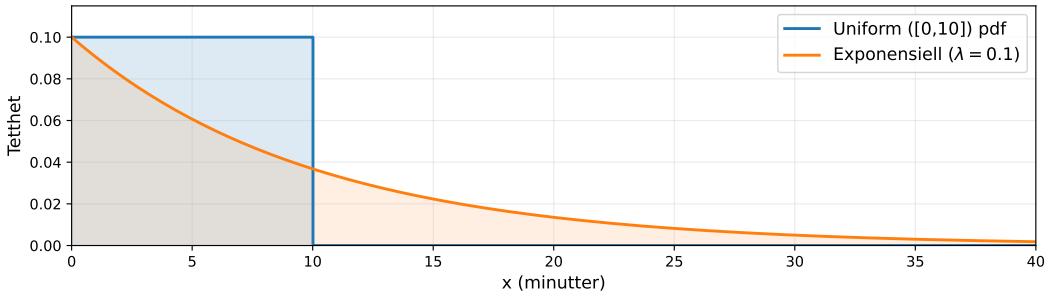
En buss kommer **i gjennomsnitt hvert 10. minutt**.

La X være den stokastiske variabelen «tid du må vente på bussen» (målt i minutter).

Da er $X \sim Exp(0.1)$ med

$$\mathbb{E}(X) = \frac{1}{0.1} = 10 \text{ minutter}$$
$$Var(X) = \frac{1}{0.1^2} = 100 \text{ minutter}^2$$
$$sd(X) = \sqrt{100} = 10 \text{ minutter}$$

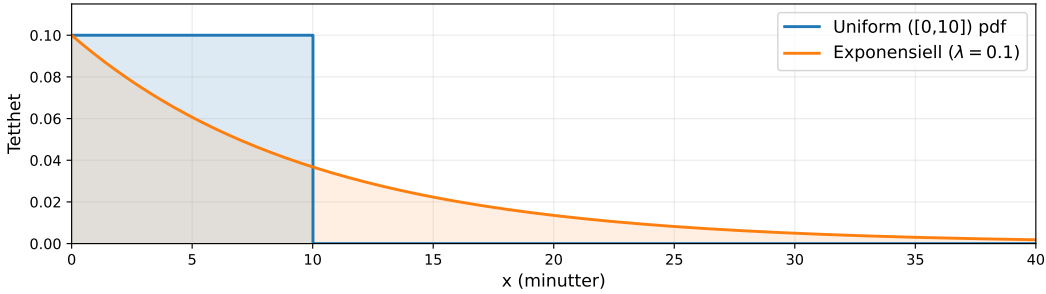
Eksempel: Ventetid på bussholdeplassen



Uniform: Bussen kommer **nøyaktig hvert 10. minutt**

Eksponensiell: Bussen kommer **i gjennomsnitt hvert 10. minutt**

Eksempel: Ventetid på bussholdeplassen



Bussen kommer **nøyaktig hvert 10. minutt**: Hvis du allerede har ventet i 8 minutter, da kommer bussen sikkert innenfor de neste to minuttene.

Bussen kommer **i gjennomsnitt hvert 10. minutt**: Hvis du allerede har ventet i 10 minutter, er sannsynligheten for å vente i ytterligere 10 minutter den samme som da du først ankom. Prosessen har ingen hukommelse.

Eksponensiell fordeling er hukommelsesløs

Anta at X følger en eksponensiell fordeling, $X \sim Exp(\lambda)$. Da gjelder

$$P(X > t) = 1 - P(X \leq t) = 1 - F(t) = 1 - (1 - e^{-\lambda t}) = e^{-\lambda t},$$

for alle $t > 0$.

Regnereglen for betinget sannsynlighet gir dermed for alle $t, s > 0$,

$$P(X > s + t \mid X > s) = \frac{P(X > s + t)}{P(X > s)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda s}} = e^{-\lambda t} = P(X > t).$$

Eksempel: Ventetid på bussholdeplassen

Uniform: Bussen kommer **nøyaktig hvert 10. minutt**

Forventning: 5 minutter.

Median: 5 minutter.

$P(X \leq 10) = 1$.

Eksponensiell: Bussen kommer **i gjennomsnitt hvert 10. minutt**

Forventning: 10 minutter.

Median: $\frac{ln(2)}{\lambda} = \frac{ln(2)}{0.1} = 6.93$ minutter.

$P(X \leq 10) = 0.63$.

(Husk: Forventningen til en skjev kurve er trukket bort fra medianen ut i den lange halen.)

Eksempel: Ventetid på bussholdeplassen

```
import numpy as np
from scipy.stats import uniform, expon

a, b = 0.0, 10.0
lam = 0.1
dist_u = uniform(loc=a, scale=b - a)
dist_e = expon(scale=1.0/lam)
u_mean = dist_u.mean()
u_med = dist_u.median()
u_p = dist_u.cdf(10.0)
e_mean = dist_e.mean()
e_med = dist_e.median()
e_p = dist_e.cdf(10.0)
print(f"Uniform: mean = {u_mean:.2f}, median = {u_med:.2f}, P(X <= 10) = {u_p:.2f}")
print(f"Exponential: mean = {e_mean:.2f}, median = {e_med:.2f}, P(X <= 10) = {e_p:.2f}")
```

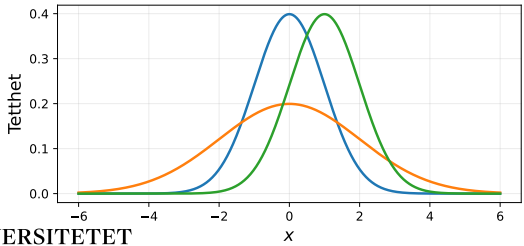
```
Uniform: mean = 5.00, median = 5.00, P(X <= 10) = 1.00
Exponential: mean = 10.00, median = 6.93, P(X <= 10) = 0.63
```

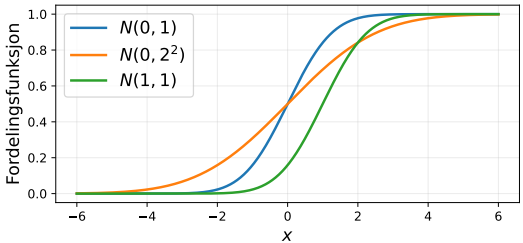
Normalfordeling

Anta at X er en kontinuerlig stokastisk variabel med $\mathcal{X} = (-\infty, +\infty)$, dvs. X kan ta alle reelle verdier.

X har **normalfordeling** med forventing μ og varians σ^2 og vi skriver $X \sim N(\mu, \sigma^2)$, hvis

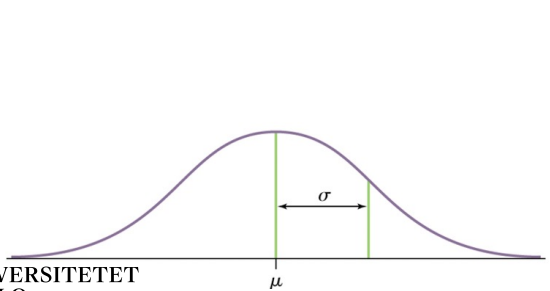
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \qquad x \in \mathbb{R}.$$

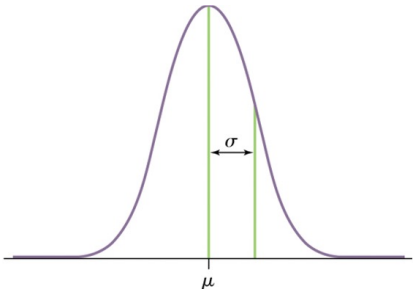




Normalfordelinger er en viktig klassse av sannsynlighets -fordelinger

- Alle normalkurver er symmetriske, en-toppede, og klokke-formede.
- Forventningen μ er midtpunktet.
- Standardavviket σ er avstanden fra μ til der kurvaturen forandrer seg på begge sider.
- En spesifikk normalkurve beskrives fullstendig ved μ og σ.





Normalfordeling = Gaussisk fordeling



Normalfordeling i scipy.stats

```python
import numpy as np
from scipy.stats import norm

mu, sigma = 0.0, 1.0
dist = norm(loc=mu, scale=sigma)


print(f"E(X) = {dist.mean():.1f}")
print(f"Var(X) = {dist.var():.2f}")

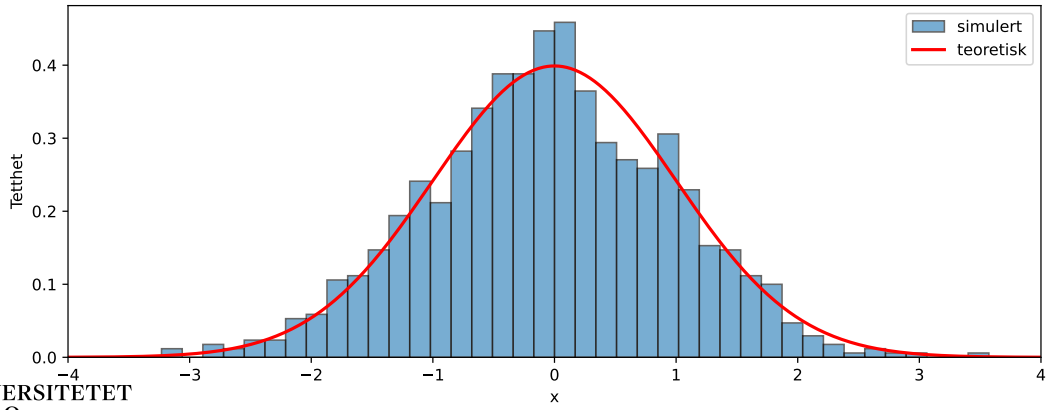
points = np.array([-4.0, -2.0, -1.0, 0.0, 1.0, 2.0, 4.0])
pdf_vals = dist.pdf(points)
cdf_vals = dist.cdf(points)
print("\nVerdi\tPDF\t\tCDF")
for p, pdf_val, cdf_val in zip(points, pdf_vals, cdf_vals):
    print(f"{p:5.1f}\t{pdf_val:0.3f}\t{cdf_val:0.3f}")
```

```
E(X) = 0.0
Var(X) = 1.00

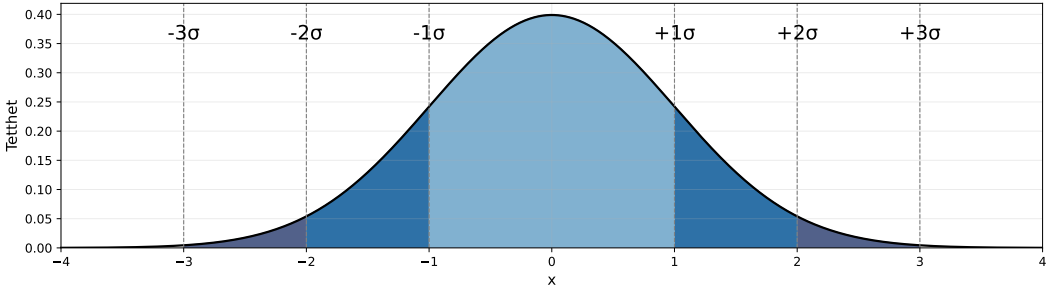
Verdi   PDF     CDF
 -4.0   0.000   0.000
 -2.0   0.054   0.023
 -1.0   0.242   0.159
  0.0   0.399   0.500
  1.0   0.242   0.841
  2.0   0.054   0.977
  4.0   0.000   1.000
```

Normalfordeling i scipy.stats

```
rng_seed = 123
n_samples = 1000
samples = dist.rvs(size=n_samples, random_state=rng_seed)
```



68-95-99.7-regelen



I normalfordelingen med forventning μ og standardavvik σ:

- tilnærmet 68% av observasjonene faller innenfor σ fra μ.
- tilnærmet 95% av observasjonene faller innenfor 2σ fra μ.
- tilnærmet 99.7% av observasjonene faller innenfor 3σ fra μ.

Eksempel: IQ-score

IQ-tester er designed på en måte hvor IQ-scorene følger en normalfordeling med gjennomsnitt på 100 og standardavvik på 15.

- 68 % av befolkningen har en IQ-score mellom 85 og 115.
- 95 % av befolkningen har en IQ-score mellom 70 og 130.
- 99.7 % av befolkningen har en IQ-score mellom 55 og 145.

Eksempel: IQ-score

La X betegne IQ-scoren til en tilfeldig person.

Da er $X \sim N(100, 15^2)$.

```
import numpy as np
from scipy.stats import norm
mu = 100.0
sigma = 15.0
dist = norm(loc=mu, scale=sigma)
```

a) Hva er sannsynligheten for at en tilfeldig person har IQ-score mellom 90 og 110?

```
# Vi skal finne P(90 <= X <= 110)
a, b = 90.0, 110.0
p = dist.cdf(b) - dist.cdf(a)
print(f"P({a} <= X <= {b}) = {p:.4f}")
```

```
P(90.0 <= X <= 110.0) = 0.4950
```

Eksempel: IQ-score

La X betegne IQ-scoren til en tilfeldig person.

Da er $X \sim N(100, 15^2)$.

```
import numpy as np
from scipy.stats import norm
mu = 100.0
sigma = 15.0
dist = norm(loc=mu, scale=sigma)
```

b) Hva er sannsynligheten for at en tilfeldig person har IQ på over 95?

```
# Vi skal finne P(X > 95)
c = 95.0
p_greater = dist.sf(c)
# same as: p_greater = 1 - dist.cdf(c)
print(f"P(X > {c}) = {p_greater:.4f}")
```

```
P(X > 95.0) = 0.6306
```

Eksempel: IQ-score

La X betegne IQ-scoren til en tilfeldig person.

Da er $X \sim N(100, 15^2)$.

```
import numpy as np
from scipy.stats import norm
mu = 100.0
sigma = 15.0
dist = norm(loc=mu, scale=sigma)
```

c) Hvor høy IQ trenger én for å ligge blant de med 5% høyest IQ-score?

```
# Vi skal finne x_q med P(X <= x_q) = 0.95
k = 0.95
x_q = dist.ppf(k)
# ppf = percent point function
# eller kvantilfunksjonen
print(f"P(X <= x_q) = 0.95 if x_q = {x_q:.4f}")
```

```
P(X <= x_q) = 0.95 if x_q = 124.6728
```

Dvs. minst 95% av befolkningen har lavere IQ-score enn én med IQ på 125 eller mer.

Standardisering av normalfordelte variabler

Anta at X er normalfordelt med forventning μ og standardavvik σ: $X \sim N(\mu, \sigma^2)$.

Da gjelder for alle $a, b \in \mathbb{R}$ at $aX + b \sim N(a\mu + b,\ a^2\sigma^2)$.

Følgelig: Vi kan **standardisere** X ved å definere $Z = \dfrac{X - \mu}{\sigma}$. Da gjelder:

- Z er **standard normalfordelt**: $Z \sim N(0, 1)$
- Sannsynligheter kan slås opp i standard normalfordelingstabeller:
  - $P(X \leq x) = P\left(Z \leq \dfrac{x - \mu}{\sigma}\right)$
  - $P(a \leq X \leq b) = P\left(\dfrac{a - \mu}{\sigma} \leq Z \leq \dfrac{b - \mu}{\sigma}\right)$

Eksempel: Bruk av standardisering og Z-tabell

Anta at X er normalfordelt med $\mu = 170$ cm og $\sigma = 6$ cm: $X \sim N(170, 6^2)$.

Problem: Finn $P(X \leq 180)$.

1. **Standardiser**:

$$Z = \frac{X - \mu}{\sigma} \Rightarrow z = \frac{180 - 170}{6} = \frac{10}{6} \approx 1.67$$

   Vi trenger altså $P(Z \leq z) = P(Z \leq 1.67)$.

2. **Bruk en liten Z-tabell**
   (tabellen gir $P(Z \leq z)$ for en standard normalfordelt variabel Z)

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
1.6	0.9452	0.9459	0.9466	0.9474	0.9481	0.9488	0.9495	0.9502	0.9509	0.9515
1.7	0.9554	0.9561	0.9568	0.9573	0.9582	0.9589	0.9596	0.9603	0.9610	0.9616

For $z = 1.67$ ser vi på rad 1.6 og kolonne 0.07: $P(Z \leq 1.67) \approx 0.9502$.

3. **Konklusjon:**

$$P(X \leq 180) = P(Z \leq 1.67) \approx 0.95$$

Det vil si at omtrent 95% av observasjonene er ≤ 180 cm.

Normalfordelinger i statistisk modellering

- Normalfordelinger er gode beskrivelser for mange fordelinger av reelle data, for eksempel egenskaper målt på mange individer.
- Normalfordelinger er gode tilnærminger til mange former for tilfeldige utfall.
- Mange statistiske metoder for å trekke slutninger fra data basert på normalfordelinger egner seg ofte også for andre tilnærmet symmetriske fordelinger.
- Merk: Noen datasett følger ikke en normalfordeling! Eksempel: Ventetider.

Sentralgrenseteoremet

La $X_1, X_2, \ldots, X_n$ være uavhengige og identisk fordelte (i.i.d.) variabler med forventning $\mathbb{E}(X_i) = \mu$ og varians $\mathrm{Var}(X_i) = \sigma^2$ hvor vi antar at $|\mu| < \infty$ og $0 < \sigma^2 < \infty$.

Definer utvalgs-gjennomsnittet $\overline{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$.

Sentralgrenseteoremet

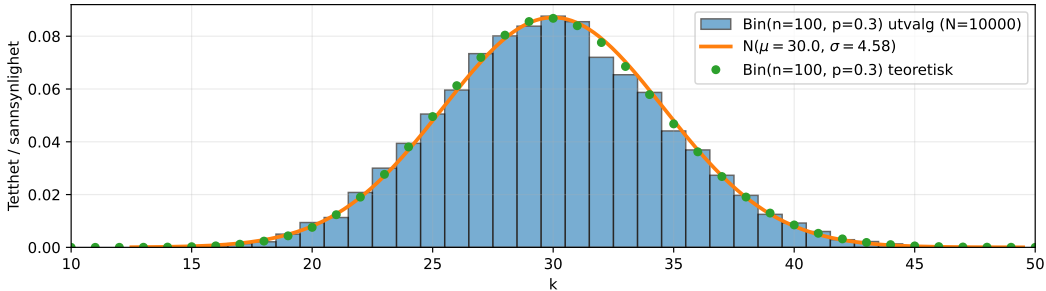
Når n er stor, er fordelingen til $\overline{X}_n$ tilnærmet normal: $\overline{X}_n \approx N\left(\mu, \frac{\sigma^2}{n}\right)$.

Standardisering av $\overline{X}_n$ gir: $Z = \frac{\overline{X}_n - \mu}{\sigma/\sqrt{n}} \approx N(0,1)$ for store n.

Normalfordeling som tilnærming til binomisk fordeling

La $X \sim Bin(n = 100, p = 0.3)$.

Da er $\mathbb{E}(X) = np = 30$ og $\text{sd}(X) = \sqrt{np(1-p)} = 4.58$.



Kvantil-kvantil plott

Anta at vi har n observasjoner $x_1, x_2, \ldots, x_n$ av uavhengige og identisk fordelte stokastiske variabler $X_1, X_2, \ldots, X_n$.

Histogrammer kan indikere formen på den empiriske fordelingen til dataene og vi kan, for eksempel, vurdere om den er tilnærmet normal.

En bedre måte å sjekke dette på er å plotte dataene ved bruk av et **kvantil-kvantil plott**.

Fremgangsmåte kvantil-kvantil plott:

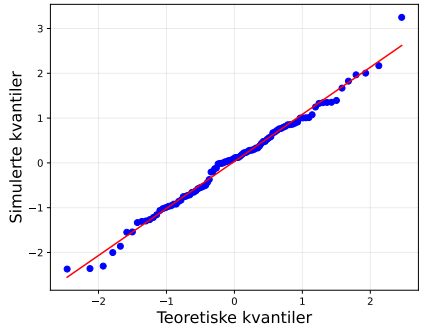
- Ranger observasjonene $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$.
- Finn hvilken persentil hver observasjon svarer til (hvor mange % av alle observasjonene er lik eller mindre enn denne observasjonen).
- Finn de tilsvarende persentilene i en teoretisk sannsynlighetsfordeling du ønsker å sammenligne dataene mot.
- Plott verdien av hver observasjon mot den tilhørende teoretiske verdien.

Eksempel: Er dataene normalfordelte?

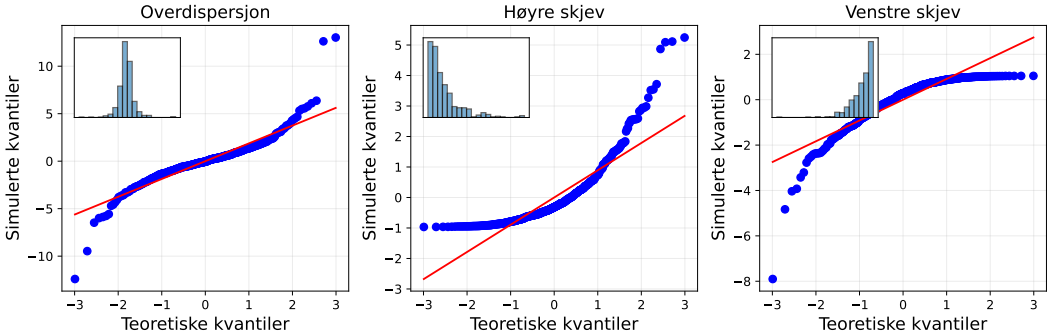
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats

mu, sigma = 0.0, 1.0
dist = stats.norm(loc=mu, scale=sigma)
sample_normal = dist.rvs(size=100, random_state=12345)

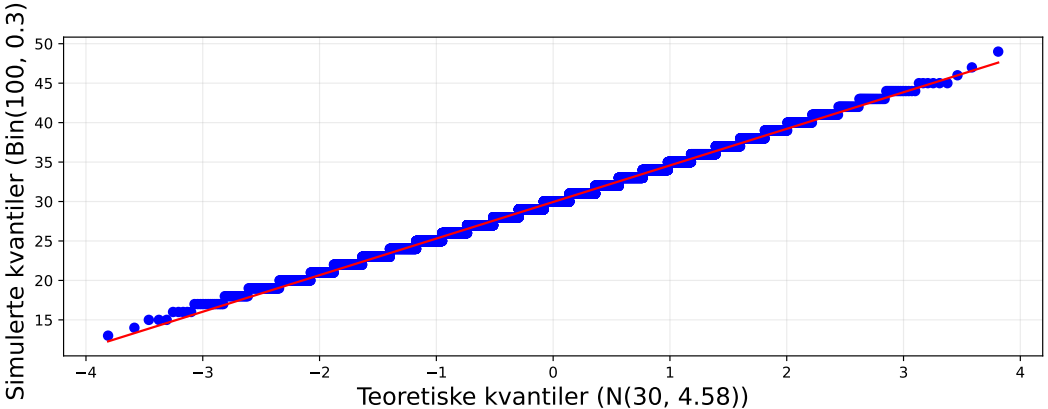
fig1, ax1 = plt.subplots(figsize=(6, 5))
res = stats.probplot(sample_normal, dist="norm", plot=ax1)
ax1.grid(alpha=0.25)
ax1.grid(alpha=0.25)
ax1.set_title(" ")
ax1.set_xlabel("Teoretiske kvantiler", fontsize=16)
ax1.set_ylabel("Simulerte kvantiler", fontsize=16)
plt.tight_layout()
plt.show()
```



Eksempel: Er dataene normalfordelte?



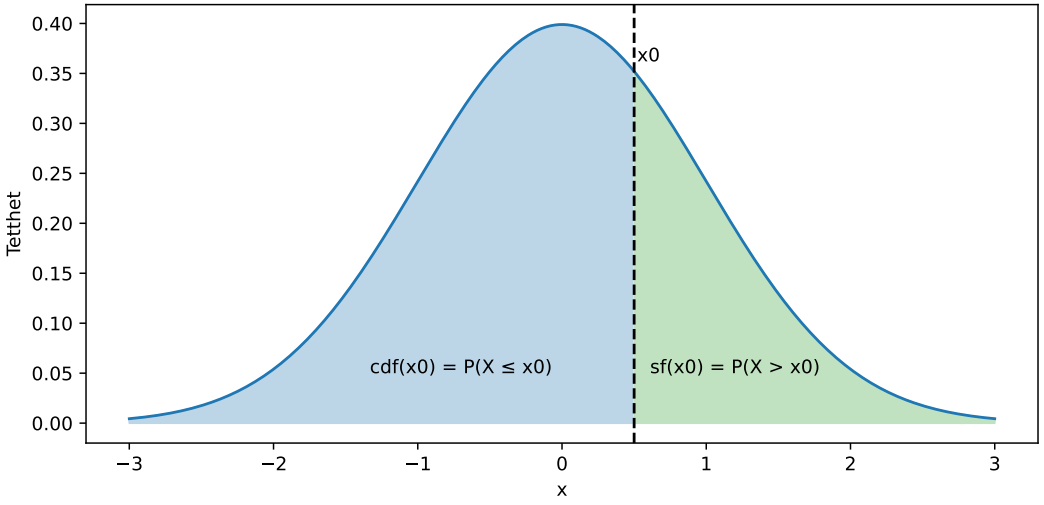
Normalfordeling som tilnærming til binomisk fordeling



Oppsummering av `scipy.stats`: `cdf`, `pdf`, `sf`, `ppf`, `rvs`

Vi tenker oss en kontinuerlig fordeling, f.eks. standard normalfordelingen.

- `pdf(x)`: **tetthet** i punktet x
  (høyden på kurven i x).
- `cdf(x)`: **kumulativ fordeling**
  $F(x) = P(X \leq x)$ (arealet under kurven til venstre for x).
- `sf(x)`: **overlevelsesfunksjon**
  $S(x) = P(X > x) = 1 - F(x)$ (arealet under kurven til høyre for x).
- `ppf(q)`: **kvantilfunksjon**
  $x_q = F^{-1}(q)$ slik at $P(X \leq x_q) = q$
  (f.eks. $x_{0,95}$ er 95%-kvantilen).
- `rvs(size=n)`: trekker **tilfeldige observasjoner**
  $X_1, \ldots, X_n$ fra fordelingen.


0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
Tetthet
x0
cdf(x0) = P(X ≤ x0)
sf(x0) = P(X > x0)
−3
−2
−1
0
1
2
3
x